HTL - Perg
Höhere Abteilung für Informatik

Diplomarbeit

Projektteam: Tristan Eder
Christoph Gattermayr
Julian Mühlberger

Projektbetreuer: Mag. Gabriela Danner

In Zusammenarbeit mit HerBody GbR

Betreuer Herr Michael Twardy & Herr Florian Steinkellner

Bearbeitungszeitraum: 01.10.2016 – 31.03.2017

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 _____

Tristan Eder

Perg, _____ Unterschrift _____

Christoph Gattermayr

Perg, _____ Unterschrift _____

Julian Mühlberger

Kurzfassung

Die folgende wissenschaftliche Arbeit befasst sich mit dem Vertrieb von digitalen Fitnesspaketen und wurde in Zusammenarbeit mit der HerBody GbR durchgeführt.
Bei der HerBody GbR handelt es sich um ein Start-Up Unternehmen, welches digitale Fitnesspakete anbieten will, welche speziell für junge Mütter entwickelt worden sind.
Ziel ist es ein System zu erschaffen, welches aus einer Webseite, einer App und einem gemeinsamen Backend besteht.
Auf der Webseite werden dem Kunden verschiedene Fitnesspakete, bestehend aus Trainingsübungen und Ernährungstipps in Form von Rezepten, angeboten. Der Kunde, kann diese buchen und hat dadurch eine gewisse Zeit lang Zugriff darauf.
Mit der dazugehörigen, plattformübergreifenden, mobilen App kann der Benutzer dann ein Live-Training absolvieren. Der Benutzer wird außerdem von einer Stoppuhr durch das Training geleitet.
Die Webseite und die App verwenden beide dieselbe Datenbank, die die Userdaten und die Fitnesspakete in Form von Bildern und Rezepte beinhaltet.

Abstract

The following thesis deals with the distribution of digital fitness packages and was made in cooperation with HerBody GbR.

HerBody GbR is a start-up company, that wants to offer digital fitness packages, that have been specially developed for young mothers.

The aim is to create a system, which consists of a website, an app and a common backend.

The website offers the customer various fitness packages, consisting of training exercises and nutrition tips in form of recipes. The customer can book these packages and has access to them for a certain period of time.

The user can use our cross-platform, mobile app to run through a live-training. The special thing about our live-training is, that the user is guided by a stopwatch through the training.

The website and the app are using the same database, which contains the user data and the fitness packages in form of pictures and recipes.

Danksagung

Wir bedanken uns ganz herzlich bei allen, die uns während der Durchführung dieser Diplomarbeit zur Seite gestanden sind und uns unterstützt haben.

Besonders bedanken möchten wir uns bei unserer Betreuungslehrerin Frau Mag. Gabriela Danner, welche uns während der Durchführung der Diplomarbeit zur Seite gestanden ist und uns immer wieder mit ihrem weitreichenden projektorganisatorischen Wissen unterstützen konnte.

Darüber hinaus bedanken wir uns bei den anderen Professoren der HTL Perg, die uns bei Fragen und Problemen immer hilfsbereit unterstützt haben.

Einen besonderen Dank wollen wir noch an Herrn Dominik Dörr von der Firma Softaware für die Unterstützung bei der Systemarchitektur aussprechen.

Weiteres wollen wir uns noch bei Herrn Timim Al Timimi bedanken, welcher uns kostenlos mit Icons und Grafiken versorgt hat.

Herzlichen Dank!

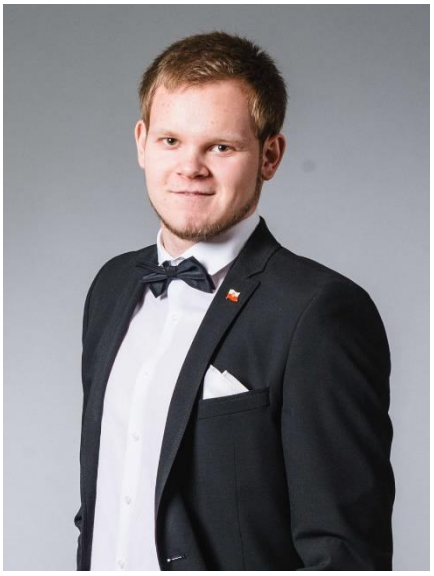
Inhaltsverzeichnis

1. Projektumgebung

1.1. Diplomanden

1.1.1. Tristan Eder

Geburtsdatum	11.08.1998
Adresse	Teichweg 1a 4470 Enns AT
Kontaktdaten	tristan.eder@gmx.at +43 677 620 882 20
Schulausbildung	2004-2008 VS1 Enns 2008-2012 BRG Enns 2012-2017 HTL - Perg
Fähigkeiten	Web- & Appentwicklungserfahrung durch Schulprojekte



1.1.2. Christoph Gattermayr

Geburtsdatum	13.07.1998
Adresse	Stelzhamer-Straße 15 4470 Enns AT
Kontaktdaten	c.gattermayr@outlook.com +43 680 206 90 87
Schulausbildung	2004-2008 VS1 Enns 2008-2012 BRG Enns 2012-2017 HTL-Perg
Fähigkeiten	Mobile Computing Erfahrung durch Schulprojekte (Acheeve, SmartShift), den Freigegenstand und diverse Privatprojekte

1.1.3. Julian Mühlberger

Geburtsdatum	31.10.1997
Adresse	Klein Erla 109 4303 St. Pantaleon-Erla AT
Kontaktdaten	muehli2@gmx.at +43 660 45 54 365
Schulausbildung	2004-2008 VS St. Pantaleon 2008-2012 NMS Schubertviertel 2012-2017 HTL-Perg
Fähigkeiten	Datenbankerfahrung durch Schulprojekte, Fitnesserfahrung

1.2. Projektumfeld

1.2.1. Schule

Die Diplomarbeit wurde in Zusammenarbeit mit der HTL-Perg im Zuge der Reife- und Diplomprüfung durchgeführt.

Machlandstraße 48
4320 Perg
Tel. +43 72 62 / 539 26
Fax. +43 72 62 / 539 26 - 6

Abbildung 1 - HTL-Perg Logo

1.2.2. Auftraggeber

Abbildung 2 - HerBody Logo

Unternehmen	HerBody GbR
Kontaktdaten	team@her-body.at
Unternehmenssitz	Severinweg 1/4 4320 Perg AT
Ansprechpartner	Michael Twardy +43 660 762 53 32
	Florian Steinkellner +43 699 172 848 87

Tabelle 1 - Auftraggeber

Die HerBody GbR wurde von Michael Twardy und Florian Steinkellner im Jahr 2016 als junges Startup-Unternehmen gegründet. Sie haben es sich zum Ziel gemacht, Müttern beim Erreichen ihrer Fitnessziele mit digitalen Fitnesspaketen zu unterstützen.

Der Kontakt entstand dadurch, dass der Mitgründer Florian Steinkellner an den Direktor Christian Reisinger herangetreten ist und ihn mit der Suche nach einem motivierten Projektteam beauftragte.

1.2.3. Betreuungslehrer

Für projektorganisatorische Fragen stand uns während der Durchführung der Diplomarbeit immer Frau Mag. Gabriela Danner mit ihrer weitrechenden Projekterfahrung zur Seite. Vor allem bei Problemen und Fragen hat sie uns mit ausführlichen Lösungsvorschlägen aushelfen können.

Mag. Gabriela Danner

Email:
g.danner@htl-perg.ac.at

Abbildung 3 - Betreuungslehrer

Werdegang:

- Studium der Betriebs- und Verwaltungsinformatik an der Universität Linz
- 1988 Sponsion
- 1988-1992 Lehrtätigkeit an der Universität Linz und am BFI Linz
- 1992-1994 Softwareentwicklerin und -betreuerin, Fa. Ponte Wels
- 1994-1998 2 Kinder
- 1999 Einstieg in den Lehrberuf an der HTL-Leonding

2. Hintergrund

2.1. Ausgangslage

Einer der beiden Auftraggeber dieser Diplomarbeit betreibt ein Fitnessstudio in Perg. Ihm ist aufgefallen, das immer weniger Frauen Zeit für den Fitnessstudio-Besuch haben. Das liegt daran, dass sich viele seiner Kundinnen um ihre Kinder und den Haushalt kümmern müssen und daher keine Zeit finden, um im Fitnessstudio zu trainieren.

Um dieses Problem zu lösen, haben die Auftraggeber ein Fitnessprogramm entwickelt, mit dem die Kunden ortsunabhängig trainieren können und zusätzlich Rezepte erhalten, die schnell zubereitet sind, um eine fitnessgerechte Ernährung im stressigen Alltag zu ermöglichen.

Da es sich bei den Auftraggebern um Start-Up-Unternehmer handelt, erstellen wir die Lösung komplett neu und bauen nicht auf einem vorhandenen Projekt auf.

2.2. Zielsetzung

2.2.1. Projektziel

Projektziel ist es ein funktionierendes, plattformübergreifendes System, bestehend aus einer Webseite und einer mobilen App, zu erschaffen, welches Online-Fitnesspläne anbietet. Dieses System verwendet ein gemeinsames Backend.

2.2.2. Geschäftsziel

Geschäftsziel ist es Fitnesspläne speziell für Mütter anzubieten, welche sich dann automatisiert verkaufen lassen und somit den Gewinn des Betreibers maximieren. Wie hoch der Gewinn schlussendlich wirklich ausfällt, kann noch nicht gesagt werden, da er von Marketing und Verwendung abhängig ist und der Gewinn bei diesem Start-Up generell nicht einschätzbar ist.

2.2.3. Ziel der Benutzer

Das Ziel der Benutzer ist, Gewicht abzunehmen und mögliche Spuren einer Schwangerschaft zu minimieren.

2.3. Thematik

2.3.1. Training

Neben reichhaltiger, bewusster und gesunder Ernährung ist es essentiell, seinen Körper zu bewegen, um nachhaltig gesund durchs Leben zu gehen. Bewegung trainiert die Muskeln unseres Körpers, stärkt Knochen und Bänder, steigert unsere Immunkräfte, fördert die Produktion von Hormonen und steigert somit jedermanns Wohlbefinden. Nebenbei hat Bewegung noch einen positiven Effekt auf unser Aussehen. Unsere Muskeln wachsen, geben unserem Körper Kurven und Kanten und lassen ihn athletisch und ästhetisch wirken. Jedoch ist Training nicht gleich Training. Man kann es grob einteilen, je nach Zielen welche man verfolgt. Zum Beispiel möchte man einfach nur Ausdauer aufbauen, spezielle Muskelpartien stärken, eventuell auf eine spezielle Sportart hintrainieren, um einfach nur abzunehmen oder die Muskelmasse steigern. Das ist lange noch nicht alles. Je nach gesetztem Ziel gibt es verschiedene Wege dieses zu erreichen.

Egal was man trainiert, es ist sehr wichtig vorher die dafür notwendige Nahrung zu sich zu nehmen. Der Körper braucht eine Menge Energie. Führt man diese nicht in genug großer Menge zu, wird man in seinen Erfolgen stagnieren. Dazu jedoch mehr im Thema Ernährung (siehe Kapitel 2.3.2. Ernährung).
Ein gutes Training beginnt immer mit ausgiebigen Aufwärmen des Körpers, oder bei speziellen Trainings, nur der Muskelpartien, welche beansprucht werden. Hier sollte besonders darauf geachtet werden, dass jeweilige Rotationsgelenke (Hüfte und Schulter) keinesfalls zu kurz kommen. Diese sollten mit speziellen Rotationsübungen aufgewärmt werden, um auch genau die Bänder, Knorpel und Gelenksschalen auf die bevorstehende Ertüchtigung vorzubereiten.

Nach dem Aufwärmen kommt das Dehnen. Hier ist zu beachten es nicht zu übertreiben, oft sind gezerrte oder sogar gerissene Bänder die Folge. Grundsätzlich sollte nach längeren Ruhepausen vom Training langsam mit dem Dehnen begonnen werden. Dehnen, wie der Name schon sagt, dehnt die Muskel und Bänder, um diese vor Verkürzungen zu schützen.

Auch Dehnen kann man grob in verschiedenen Arten einteilen:

Dynamisches Dehnen

Dehnposition mit schneller Bewegung einnehmen, nach kurzer Zeit (einer Sekunde) wieder verlassen, danach wieder mit kurzer Ausholbewegung die Dehnposition einnehmen. Oft werden diese Bewegungen als rhythmisch, schwingend oder ballistisch bezeichnet.

Statisches Dehnen

Dehnposition wird langsam eingenommen und mehrere Sekunden bis Minuten gehalten.

AC (Antagonist-Contract)

Unterschied zum statischen Dehnen ist nur, dass während der Ruhepause der einen Muskelgruppe die kontrahierende Gruppe gedehnt wird. (Bizeps – Trizeps, Brust – Rücken, Oberschenkelbeuger – Oberschenkelstrecker, Bauch – unterer Rücken)

CR (Contract-Relax)

Bevor der Muskel gedehnt wird, wird dieser möglichst isoliert, maximal angespannt/beansprucht und sofort danach gedehnt.

CR – AC

Verknüpfung der Typen.

Das sind die meistverbreiteten Typen, um Vor- und Nachteile zu erfahren, beziehungsweise welche Typen man anwenden soll, sollte man Fachkräfte befragen. Generell kann man aber sagen, Dinge die man beispielweise im Turnunterricht oder bei Sportvereinen gelernt hat kann man wieder anwenden.

Nach diesen zwei Teilen folgt das eigentliche Training, welches darauf zugschnitten ist, den ganzen Körper, in möglichst kurzer Zeit, möglichst effizient zu beanspruchen. Dabei liegt das Augenmerk auf Mütter, die gerade eine Schwangerschaft hinter sich hatten. Um ihnen so wenig Zeit wie nur möglich zu nehmen, ist das Training in gut 15 Minuten vollendet. Es werden keine zusätzlichen Gewichte benötigt, beziehungsweise ist das auch genau so gewollt. Grund dafür ist, die bereits von der Geburt oder vom Alltag mitgenommene Mutter nicht überzustrapazieren, jedoch noch genug zu fordern, um ihre Traumfigur wieder zu erhalten.
Das Kickoff-Training ist im Prinzip wie ein Zirkeltraining aufgebaut.

Es dient zur Vorbereitung, also generellen Kräftigung des Körpers (Muskel, Bänder, Knochen) und kann beliebig oft wiederholt werden. Dabei sollten die Ruhephasen des Körpers nicht zu kurz kommen, übertrainieren schadet dem Körper. Dieses vier wöchige Programm ist der Beginn eines der zwei (Fit 4 You, Mum's Package) sein. In diesen zwei Paketen werden die Übungen auf zwei Tage aufgeteilt. Sie beinhalten beide sowohl statische, als auch dynamische Übungen, welche den ganzen Körper betreffen.

Den zwei oben genannten Paketen steht das dritte Paket (Wedding Package) gegenüber. Dies besitzt keine Eingewöhnungsphase. Es ist ein zehn Wochen andauernder Zirkel der sich wochenweise steigert. Dieser Plan kann entweder als Single oder als Paar durchgeführt werden, je nach Durchführung stehen verschiedene Übungen zur Verfügung.

Alle drei Pläne erreichen ein hohes Kontraktionsniveau der Muskeln, welche die nachhaltige Fettverbrennung anregt. Das wiederrum begründet, warum 15 Minuten Training einen genug großen Effekt bewirken. [1]

2.3.2. Ernährung

In den meisten Fällen wird von einer Gewichtung von 30 Prozent Training und 70 Prozent Ernährung gesprochen, um erfolgreich seinen Zielen nachzugehen. Logischerweise sind hier körperliche Ziele angesprochen, wie Leistungssteigerung und keine Ziele wie besser Fußball zu spielen.
Ein Körper braucht je nach Forderung eine gewisse Menge an Kohlenhydraten, Eiweiß und Fetten, kurz – alle Dinge, die dem Körper Energie geben. Diese Nährstoffe beschreibt man mit den Wort Makronährstoffen und sie werden in Joule beziehungsweise in Kalorien gerechnet. Neben den Makronährstoffen gibt es noch die Mikronährstoffe, welche Spurenelemente, Vitamine und Mineralstoffe umfassen.

Im Thema Ernährung ist es wichtig, eine gute Balance in allen Punkten zu halten. Da man jedoch hier nicht für die breite Masse sprechen kann, sondern eher auf jedes Individuum einzeln eingehen müsste, da alle Kriterien hier zusammenspielen (Alter, Bewegungsgrad, Geschlecht, Stoffwechseltyp, Klimatische Bedingungen und noch vieles mehr), ist es sehr schwierig eine generelle Lösung zu finden. Deshalb ist es umso wichtiger zu wissen, was man zu sich nimmt, man muss es praktisch für sich selbst herausfinden was der Körper braucht. Dies kann man entweder durch einfaches Versuchen oder jedoch auch durch spezielle Test, welche genaue Auskunft über unseren Stoffwechsel liefern, erreichen.

Das HerBody Kochbuch umfasst nur jene Rezepte, die gesunde Ernährung unterstützen und den Stoffwechsel anregen. Man kann diese beliebig mischen und braucht sich trotzdem keinerlei Sorgen um den Bedarf an Kohlenhydraten, Proteinen oder Fetten zu machen. Beispielsweise hat man an einem Tag schon zwei kalorienreichere Mahlzeiten zu sich genommen, wird man nächsten Tag kalorienarm essen, das bringt die Kalorienbilanz über beispielsweise eine Woche gesehen in ein gutes Mittel.

Folgendes beschreibt die zugrundeliegenden Überlegungen und Verwendung unserer Rezepte:

„Zwei Beispiele: Eine 50-jährige Frau, die 65 Kilo wiegt, im Büro arbeitet und gelegentlich Sport betreibt, hat einen Tagesbedarf von 1890 kcal. Würde sie sich den ganzen Tag lang gar nicht bewegen, würde ihr Grundumsatz, also der Energiewert, der notwendig ist, um die Körperfunktionen aufrechtzuerhalten, bei 1430 kcal liegen. Ein gleichaltriger Mann mit einem Gewicht von 80 Kilo, der ebenfalls einen sitzenden Beruf hat und hin und wieder Sport betreibt, braucht 2665 kcal (bzw. 2020 kcal in Ruhestellung).“ [2]

50 – 30 – 20 Verteilung [3, 4, 5]

50 Prozent Kohlenhydrate

Kohlenhydrate stellen für unseren Körper die Energiezufuhr dar. Sie werden durch Spaltung zu Glucose welches wiederum der Energielieferant für Gehirn, Organe und Muskeln ist.

Langkettige Kohlenhydrate: liefern langsam Energie

- Vollkornprodukte
- Gemüse (Kartoffeln, Reis)
- Obst

Kurzkettige Kohlenhydrate: liefern schnell Energie:

- Zucker
- Produkte aus weißen Mehl

Energiegehalt pro Gramm: 4,1 kcal

Grund für die jeweilige schnelle/langsame Entfaltung liegt in der Arbeit, welche benötigt wird um die Kohlenhydrate für den Körper verwertbar zu machen (Spaltung durch Enzyme).

Dabei muss man darauf achten, dass bei beispielsweiser großer Zufuhr von kurzkettigen Kohlenhydraten der Blutzuckerspiegel rasant ansteigt, somit steigt auch der Insulinspiegel stark. Das viele Insulin senkt den Blutzuckerspiegel sofort, meist sogar unter den Normalwert, was Müdigkeit und Heißhungerattacken zur Folge hat. Deshalb sollte man kurzkettige Kohlenhydrate eher vermeiden und durch langkettige Kohlenhydrate tauschen.

30 Prozent Proteine

Auch bekannt als Eiweiß. Eiweiße bestehen aus Ketten von Aminosäuren. Acht der bekannten 20 Aminosäuren müssen dem Körper zugeführt werden, weil er diese nicht selbst produzieren kann. Eiweiße haben eine breit gefächerte Funktion – sie sind Hauptbestandteil unserer Zellen, aber auch unserer Organe und Muskulatur. Darüber hinaus steuern Eiweiße in Form von Enzymen unzählige Körperfunktionen.

Die Abbauprodukte des Eiweißes werden von unserer Niere gefiltert und ausgeschieden. Eine normale gesunde Niere kommt mit vermehrter Eiweißaufnahme gut klar, betreibt man diese nicht ins Extreme (Eiweißvergiftungsgefahr). Diabetiker und Menschen mit Nierenunterfunktion sollten auf ihre Blutwerte achten.

Enthalten in:

- Fleisch, Fisch, Eiern, Milch und Milchprodukte
- Hülsenfrüchte, Sojaprodukte, Nüsse

Energiegehalt pro Gramm: 4,1 kcal

20 Prozent Fett

Fett setzt sich aus Glycerin und verschiedenen Fettsäuren zusammen. Vorwiegend ist Fett ein Energielieferant, jedoch nicht nur. Es ist ebenso Träger von fettlöslichen Vitaminen und versorgt den Körper mit wichtigen ungesättigten Fettsäuren. Zu diesen gehören Omega 3, und 6 Fettsäuren. Der Körper verwendet diese Fettsäuren als Polster und Wärmeschutz, außerdem schützt es Organe vor Verletzungen. Darüber hinaus ist es wichtiger Bestandteil der Zellmembran und ist beteiligt an der Produktion von verschiedenen Hormonen und anderen körpereigenen Stoffen.

Genauso wie bei übermäßiger Zuvor von Kohlenhydrate legt sich Fett als Energiespeicher als Fettpolster ab.

Generell sollte man versuchen, dass nicht mehr als 20 Prozent der täglichen Kalorienzufuhr aus Fett bestehen – diese 20 Prozent sollten vorwiegend aus mehrfach ungesättigten (meist pflanzlich) Fettsäuren bestehen.

Enthalten in:

- Fleisch, Fisch
- Öl(samen), Butter, Milchprodukte

Energiegehalt pro Gramm: 9,3 kcal

Alle diese Vorgaben sind auf den Durchschnitt berechnet und beschrieben. Diese können jedoch variieren, wenn man beispielweise besonders viel Sport betreibt.

Ebenfalls muss man bestimmte Mengen an Mikronährstoffen den Körper zuführen.

3. Produkt

3.1. Funktionalität

3.1.1. Überblick

Das Produkt besteht aus einer App und einer Webseite, welche Daten über eine gemeinsame Schnittstelle von einer Datenbank konsumieren. Das Produkt kann mithilfe eines Admininterfaces gewartet werden.

3.1.2. App

Nach Start der App wird der User mit einem Willkommensbildschirm begrüßt, der einem bereits registrierten Anwender die Möglichkeit bietet, sich mit seiner Emailadresse und seinem Passwort einzuloggen oder einem neuen User die Möglichkeit bietet, sich zu registrieren.

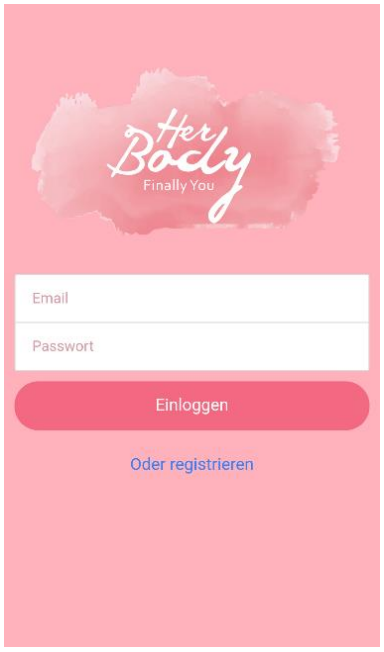


Abbildung 4 - Loginpage

Will sich der User registrieren, wird er auf das Registrierungsformular weitergeleitet, wo er gewisse Grunddaten wie zum Beispiel seinen Vor- und Zuname, seine Emailadresse und sein Passwort von sich Preis geben muss.

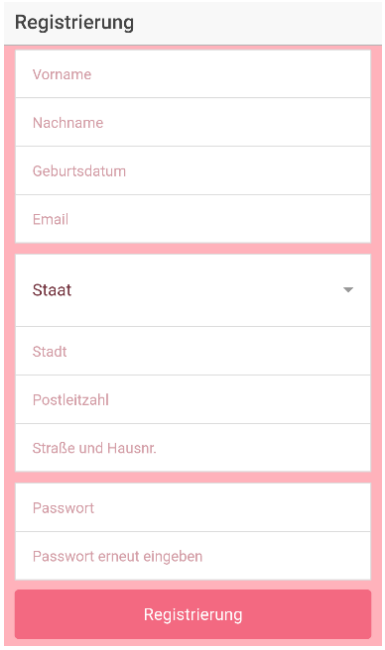


Abbildung 5 - Registerpage

Bei erfolgreicher Registrierung wird der neue User wieder auf den Willkommensbildschirm geleitet, wo sich dieser dann direkt einloggen kann. Sollte die Registrierung jedoch nicht erfolgreich gewesen sein, wird der User beim Registrierungsformular dementsprechend aufmerksam gemacht.

Ist der User nun eingeloggt wird er auf die Startseite der App geleitet, wo er gewisse Informationen und News zum HerBody-Prinzip entnehmen kann. Weiteres kann er zwischen den Tabs „Benutzer“ und „Shop“ wechseln. Darüber hinaus kann der Anwender über das Side Menu zwischen den Ansichten „Benutzer“, „Start“, „HerBody Live“, „Shop“, „Einstellungen“ und „Impressum“ navigieren.

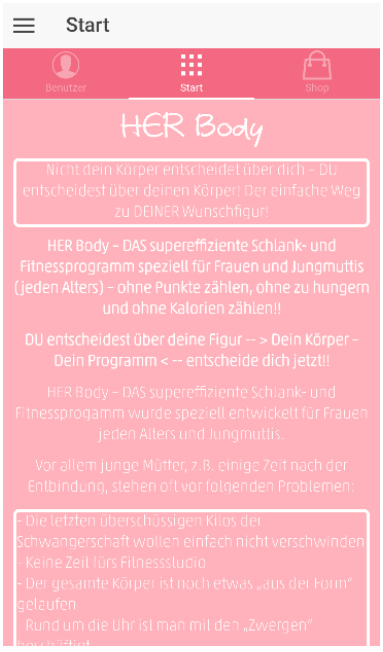


Abbildung 6 - Startpage

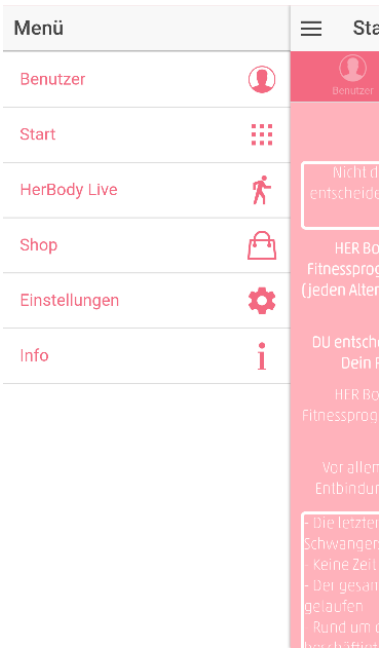


Abbildung 7 - Sidemenu

Navigiert der Benutzer zur Benutzeransicht, bekommt der angemeldete User einen Überblick über seine Stammdaten. Weiteres wird ihm ermöglicht, seine persönlichen Daten und sein Passwort zu ändern. Sollte er sein Passwort ändern, wird er vorerst aufgefordert sein altes Passwort einzugeben. Durch eine korrektere Eingabe des alten Passwortes hat der Benutzer dann die Erlaubnis sein neues Passwort zu definieren.

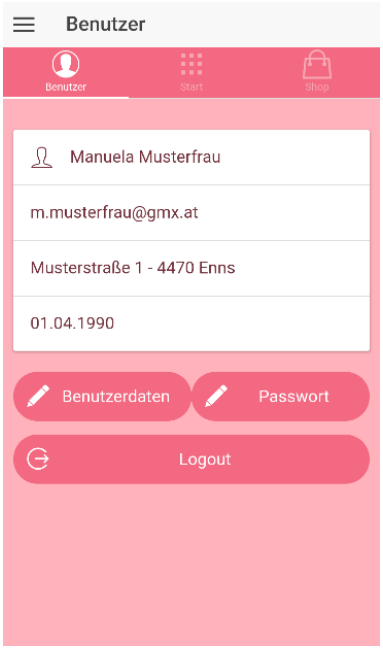


Abbildung 10 - Userpage

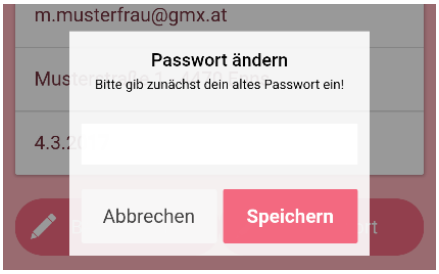


Abbildung 8 - PasswordPromt01

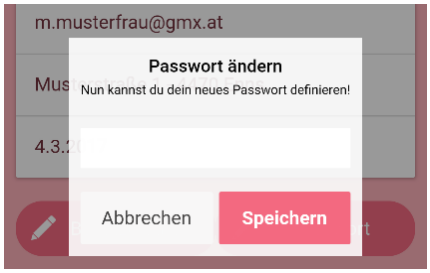


Abbildung 9 - PasswordPrompt02

In der Registerkarte Shop bekommt der Anwender eine Vorschau an Fitness-Paketen, die erworben werden können. Diese Pakete werden in der Vorschau nur durch ein Bild, den Namen, den Preis und die Dauer beschrieben. Klickt der User lange auf ein Paket, öffnet sich von unten nach oben ein kleines Kontext-Menü mit den Auswahlmöglichkeiten „Details“ und „In den Warenkorb“. Wählt der Benutzer die Detail-Option aus, so bekommt er einen detaillierten Überblick mit den dazugehörigen Beschreibungen und Informationen des Paketes.

Abbildung 11 - Shoppage

Abbildung 12 - PackageDetailPage

Hat sich der User schon für ein Paket, dass er erwerben will entschieden, so kann er es mittels dem zweiten Menü-Punkt „In den Warenkorb“ in seinen Warenkorb hinzufügen. Automatisch wird oben im Shop die Statusleiste aktualisiert und der User bekommt Einsicht, wie viele Pakete er in seinem Warenkorb hat.

Will der Anwender nun dieses Paket erwerben, kann er auf die Statusleiste mit seinem Warenkorb klicken und er erhält noch einmal eine Einsicht, welche Pakete er im Warenkorb hat und wie hoch die Summe des zu zahlenden Betrages ist. Auch hier wird die Möglichkeit eines Kontext-Menüs angeboten, dass sich wieder mittels langem Drücken des Paketes öffnet. In diesem Menü kann das ausgewählte Paket wieder aus dem Warenkorb entfernt werden. Hat sich der Kunde jedoch endgültig für den Kauf entschieden kann er mittels dem „Zur Kassa“ Buttons auschecken. Es folgt ein Hinweis, dass die Bezahlung nur über die Webseite möglich ist und eine direkte Weiterleitung auf die Webseite.

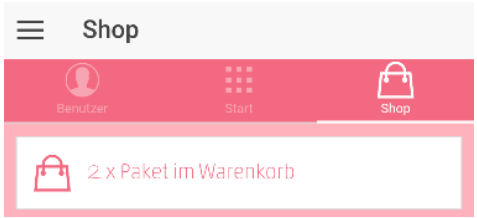


Abbildung 13 - Basketinfo

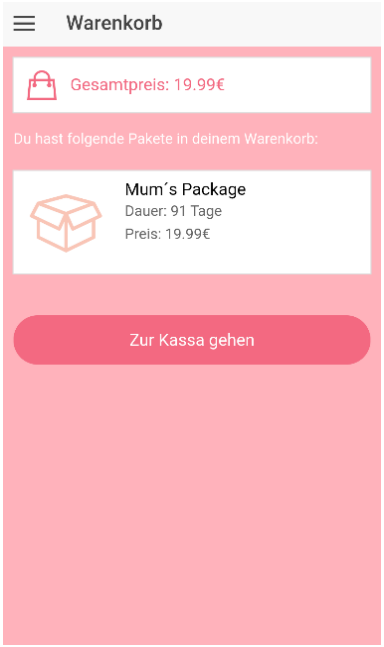


Abbildung 14 - Basketpage

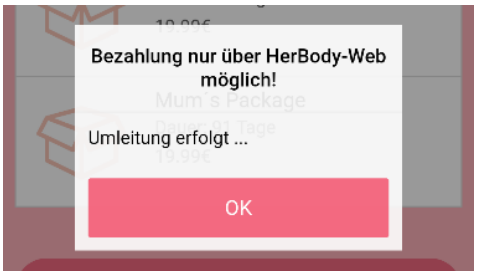


Abbildung 15 - Redirectinfo

Hat der Kunde ein Paket erworben und navigiert via Side-Menu auf die Registerkarte „HerBody-Live“ bekommt er auf dem Tab „Training“ einen Überblick über jene Pakete, die er erworben hat und mit denen er somit trainieren kann.

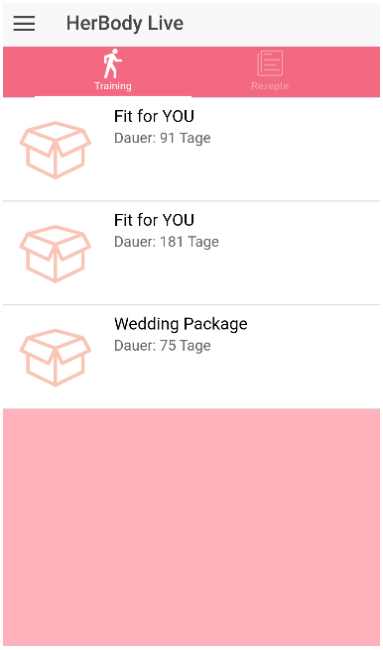


Abbildung 16 - LivePackagepage

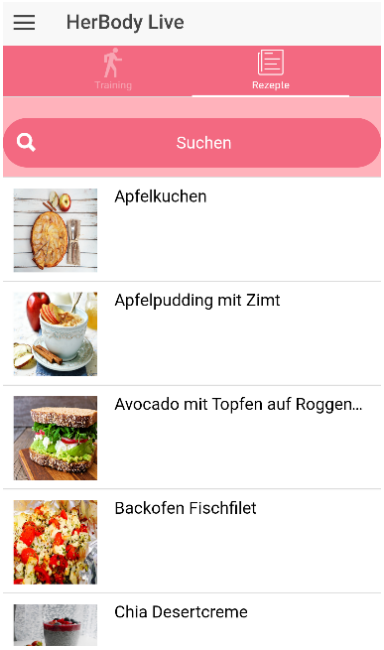


Abbildung 17 - LiveRecipepage

Im zweiten Tab „Rezepte“ werden alle Rezepte, die in den Paketen enthalten sind mit Bild in Miniaturansicht und Namen angezeigt. Um die Suche zu erleichtern kann mittels des „Suchen-Buttons“ nach bestimmten Schlagworten wie zum Beispiel „Kuchen“, „Apfel“ oder „Dressing“ gesucht werden. Anschließend werden alle Rezepte, die mit diesem Schlagwort beschlagwortet sind, angezeigt.

In die genaue Zubereitungsformel kann mittels einem Click auf das jeweilige Rezept eingesehen werden. Will der Anwender jedoch nicht kochen, sondern trainieren, so kann er wieder auf den vorherigen Tab „Training“ wechseln und ein Paket auswählen, mit dem er trainieren möchte.

Anschließend wird der User je nach Paket-Art gefragt ob er sich entweder im „Kickoff-Stadium“ oder im „Powerworkout-Stadium“ befindet oder ob er mit seinem Partner oder alleine trainieren möchte.

Die Frage ist wie folgt abhängig:

- Paket-Art = WeddingPackage
    - Frage = Möchtest du alleine oder mit deinem Partner trainieren?
        - Antwort 1 = Alleine
        - Antwort 2 = Mit dem Partner
- Paket-Art = NormalPackage
    - Frage = In welchem Stadium befindest du dich?
        - Antwort 1 = Kickoff-Plan
        - Antwort 2 = Powerworkout-Plan

Diese Frage ist deshalb wichtig, weil je nach Training andere Übungen zu absolvieren sind.

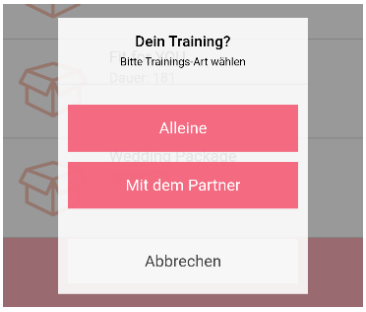


Abbildung 18 - TrainingPrompt01

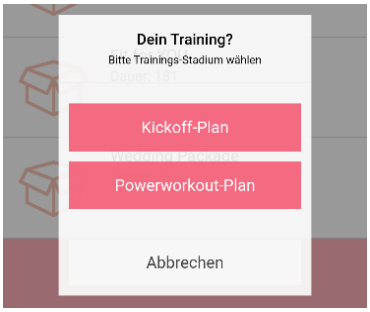


Abbildung 19 - TrainingPrompt02

Hat sich der User für ein Paket und eine „Trainingsform“ entschieden, wird er zum sogenannten „Direct-Training“ geleitet, wo er alle möglichen Übungen aufgelistet sieht.

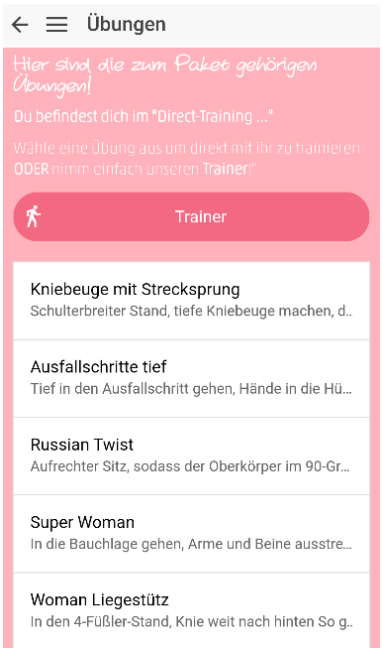


Abbildung 20 - DirectTrainingpage

Will der Kunde gezielt mit einer Übung trainieren hat er hier die Möglichkeit eine Übung auszuwählen. Anschließen werden ihm die genauen Details wie zum Beispiel Wiederholungen, Durchführungszeiten und Pausen gezeigt. Weiteres sieht der User bis zu drei Bilder auf der Übungsdetail-Seite, nämlich die Ausgangsposition, die Durchführungsposition und die Endposition. Von diesen drei Positionen ist nur die Ausgangsposition zwingend notwendig. Am Ende der Seite kann die Übung schlussendlich gestartet werden.

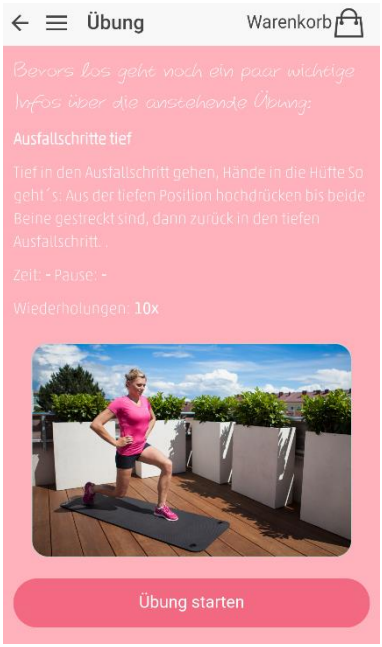


Abbildung 21 - ExerciseRevisions

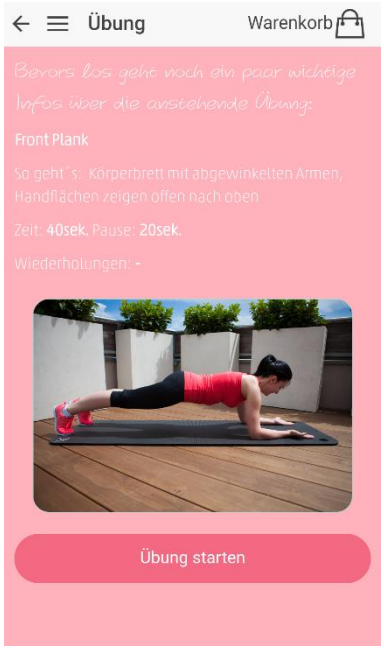


Abbildung 22 - ExerciseTime

Nach Start der Übung wird der User, sofern die Übung wiederholt werden soll noch einmal mit einem Hinweis benachrichtigt, wie oft die Übung zu absolvieren ist.

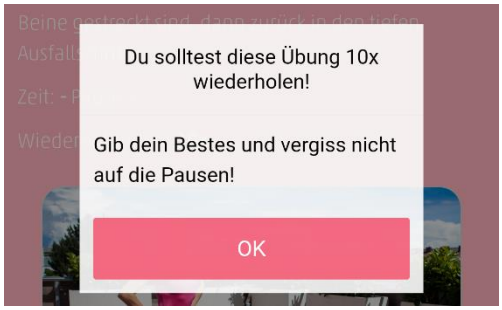


Abbildung 23 - RevisionPrompt

Sollte die Übung jedoch eine bestimmte Zeit durchgeführt werden öffnet sich von unten nach oben eine Seite mit einer Stoppuhr, die bereits auf jene Zeit gestellt ist, wie lange die Übung wiederholt werden soll. Ist die Zeit abgelaufen wird dem Anwender zur erfolgreichen Durchführung gratuliert und daran erinnert wie lange er sich eine Pause verdient hat.

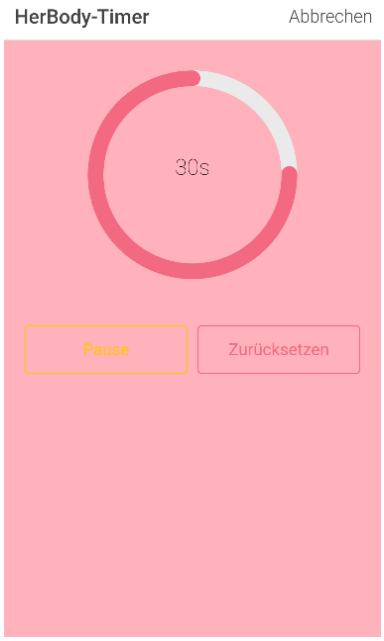


Abbildung 24 - Timerpage01

Abbildung 25 - Timerpage02

Neben dem “DirectTraining” biete sich jedoch noch eine weitere Möglichkeit für den User mit dem HerBody LiveTraining zu trainieren. Nämlich durch den HerBody-Trainer, der auf der Seite des „DirectTrainings“ optional startbar ist. Wird dieser gestartet so wird dem Anwender eine genaue Reihenfolge an Übungen vorgelegt wie die Übungen zu absolvieren sind. Das Starten einer Übung erfolgt nach dem Prinzip des „DirectTrainings“.

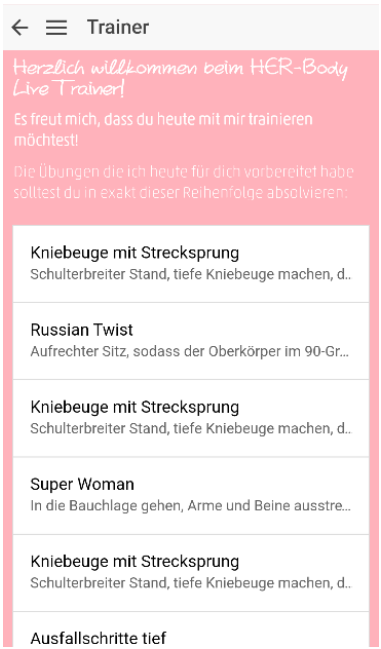


Abbildung 26 - Trainerpage

Eine weitere Registerkarte im Side Menu ist unter dem Namen „Einstellungen“ verfügbar. Wird diese geöffnet hat der Benutzer drei Möglichkeiten sich zu entscheiden:

Benutzereinstellungen:

Nach dieser Auswahl wird der User auf die Benutzerseite geleitet, wo er diverse Änderungen vornehmen kann.

Kontakt:

Unter diesem Menüpunkt wird der Benutzer auf die Kontaktseite weitergeleitet, auf welcher er Kontakt mit dem HerBody-Team aufnehmen kann.

Sollte er Fragen haben, kann er auf die Emailadresse team@her-body.at klicken, wo sich anschließend die Mail App des Gerätes öffnet in der bereits ein neuer Mailentwurf angelegt wurde und die Adresse des Empfängers und der Betreff eingefügt wurde.

Will der User jedoch Fortschrittsfotos von seinen Erfolgen an das HerBody-Team mailen, wird ihm dies durch einen Klick auf die progress@her-body.at Mailadresse ermöglicht, bei der sich ebenfalls die Mail App des Gerätes mit einer neu angelegten Mail öffnet.

Impressum:

Mit der letzten Auswahlmöglichkeit Impressum erhält der User Informationen über das Unternehmen und die verschiedenen Zahlungsmöglichkeiten.

Abbildung 27 - Contactpage

Abbildung 28 - Imprintpage

Der letzte verfügbare Menüpunkt im Side Menu ist der Punkt „Info“ wo der Anwender Informationen über das Drei-Säulen-Prinzip erfährt, auf dem HerBody aufbaut.

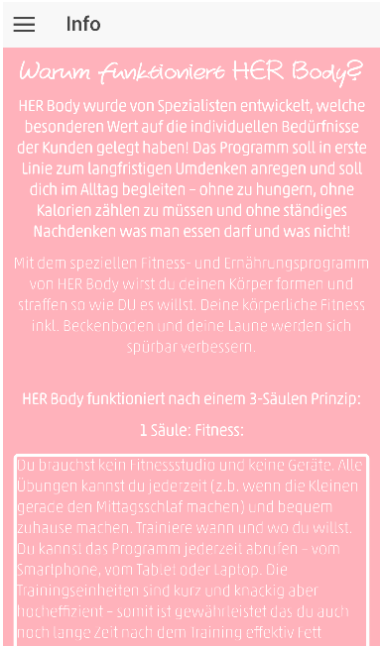


Abbildung 29 - Informationpage

3.1.3. Webseite

3.1.3.1. Aufbau

Die Webseite ist wie folgt aufgebaut. Die Indexseite besteht aus einem Webseitenheader auf dem sich eine Navigationsleiste befindet, mit der durch die Seite navigiert werden kann. Unterhalb des Headerbereichs befindet sich der Content-Bereich, der dynamisch geladen wird. Ganz unten befindet sich noch eine Footer-Leiste mit Links zum Impressum, zu den AGBs, zum Datenschutz, zu den Liefer- und Zahlungsbedingungen und zum Kontaktformular.

3.1.3.2. Navigation

Navigation: Desktop-Ansicht

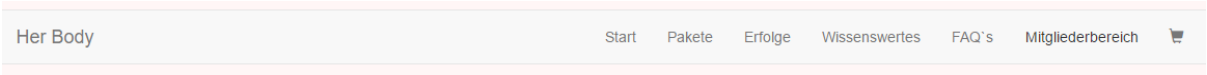


Abbildung 30 – Navigation: Desktop-Ansicht

Navigation: Smartphone-Ansicht

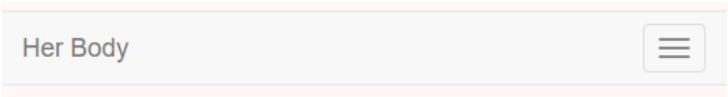


Abbildung 31 - Navigation: Smartphone - zugeklappt

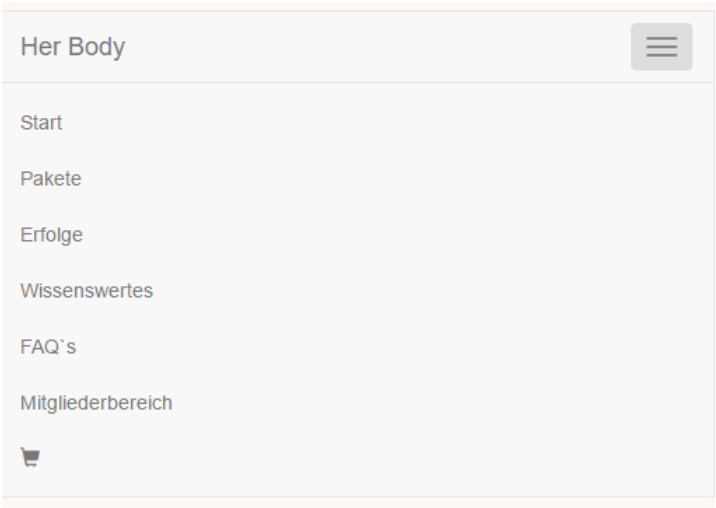


Abbildung 32 – Navigation: Smartphone - aufgeklappt

3.1.3.3. Startseite

Die Startseite dient dazu, dass potentielle Kunden einen positiven Eindruck bekommen und sich die Webseite genauer ansehen wollen.

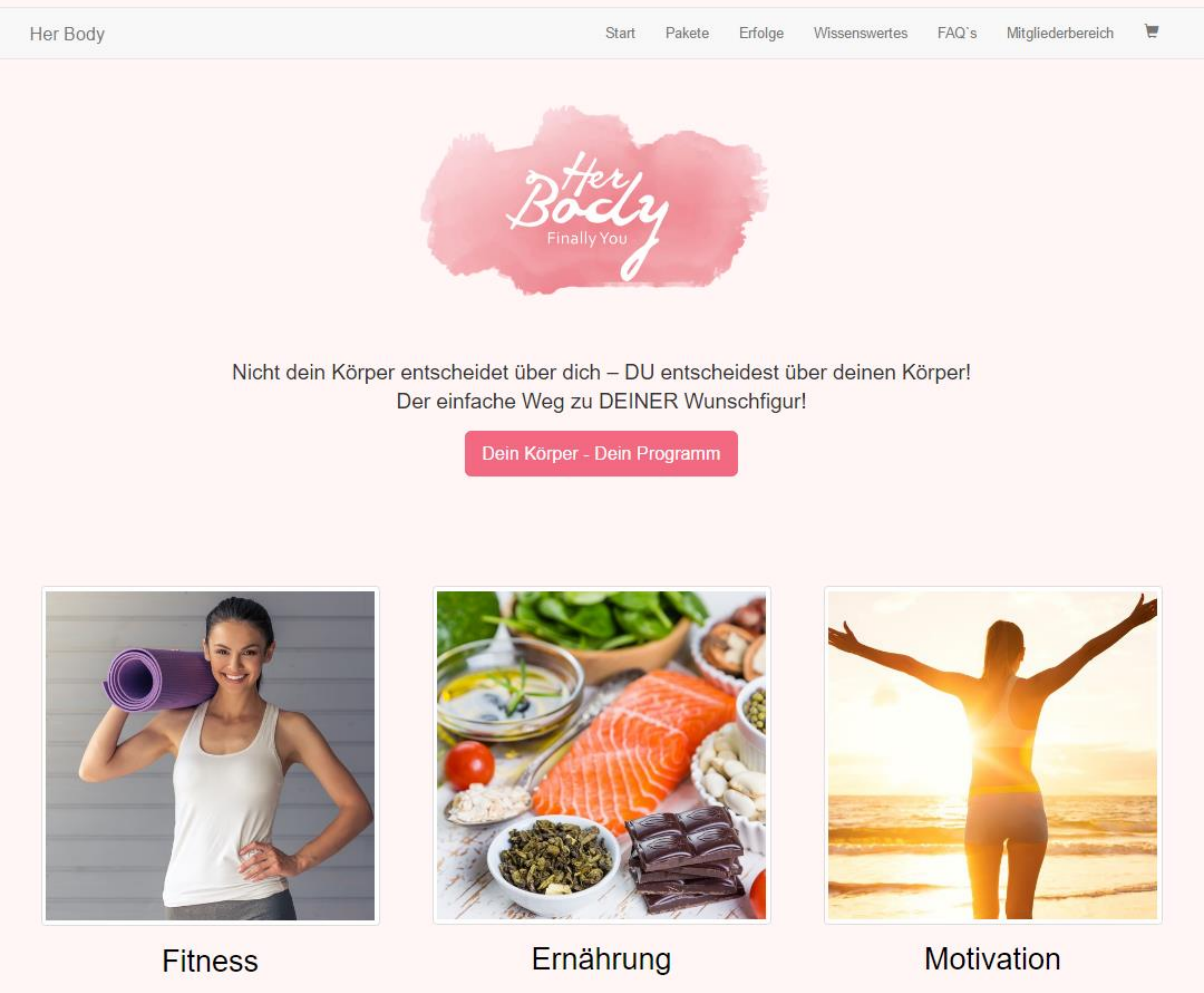


Abbildung 33 - Startseite

Die Startseite besteht aus einem Banner mit einem Motivationsspruch. Gleich darunter befindet sich der erste Conversion-Button.

Abbildung 34 - Banner

3.1.3.4. Conversion-Button

Der Conversion-Button wiederholt sich auf der Webseite nach jedem Abschnitt und dient dazu, die potentiellen Kunden auf die Pakete-Seite zu leiten und ihnen die Fitnesspakete anzubieten. Die auffällige Platzierung des Buttons soll die Verkaufsrate erhöhen und dabei helfen, die Webseitenbesucher in Kunden zu verwandeln.

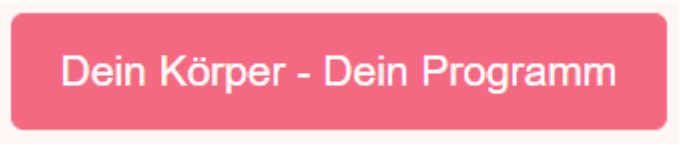


Abbildung 35 - Conversion-Button

Unter dem Banner kann sich der potentielle Kunde über die Bereiche Fitness, Ernährung und Motivation informieren.

Abbildung 36 - Themenbereiche

Wenn der Webseitenbesucher auf die einzelnen Bereiche klickt, kommt er auf die jeweilige Themen-Seite und kann sich dort weiter informieren.

3.1.3.5. Fitness

Auf der Seite „Fitness“ findet der Webseitenbesucher einige Informationen zum „HerBody - Fitnesskonzept“. Außerdem sind wieder Conversion-Buttons platziert, um den Benutzer auf die Paketseite zu leiten.

3.1.3.6. Ernährung

Auf der Seite „Ernährung“ findet der Webseitenbesucher einige Informationen zum „HerBody - Ernährungskonzept“. Außerdem kann sich der Webseitenbesucher auf dieser Seite einige „HerBody Ernährungstipps“ durchlesen. Auf der Ernährungsseite sind wieder Conversion-Buttons platziert, um den Benutzer auf die Paketseite zu leiten.

3.1.3.7. Motivation

Auf der Seite „Motivation“ findet der Webseitenbesucher einige Informationen zum „HerBody - Motivationskonzept“. Außerdem ist wieder ein Conversion-Button platziert, um den Benutzer auf die Paketseite zu leiten.

3.1.3.8. Pakete

Auf der Seite „Pakete“ kann sich der Webseitenbesucher alle verfügbaren Pakete ansehen. Zu jedem Produkt kann der Benutzer ein Paketbild, einen Paketnamen, einen Paketkurztext, eine Laufzeit und den Preis sehen.

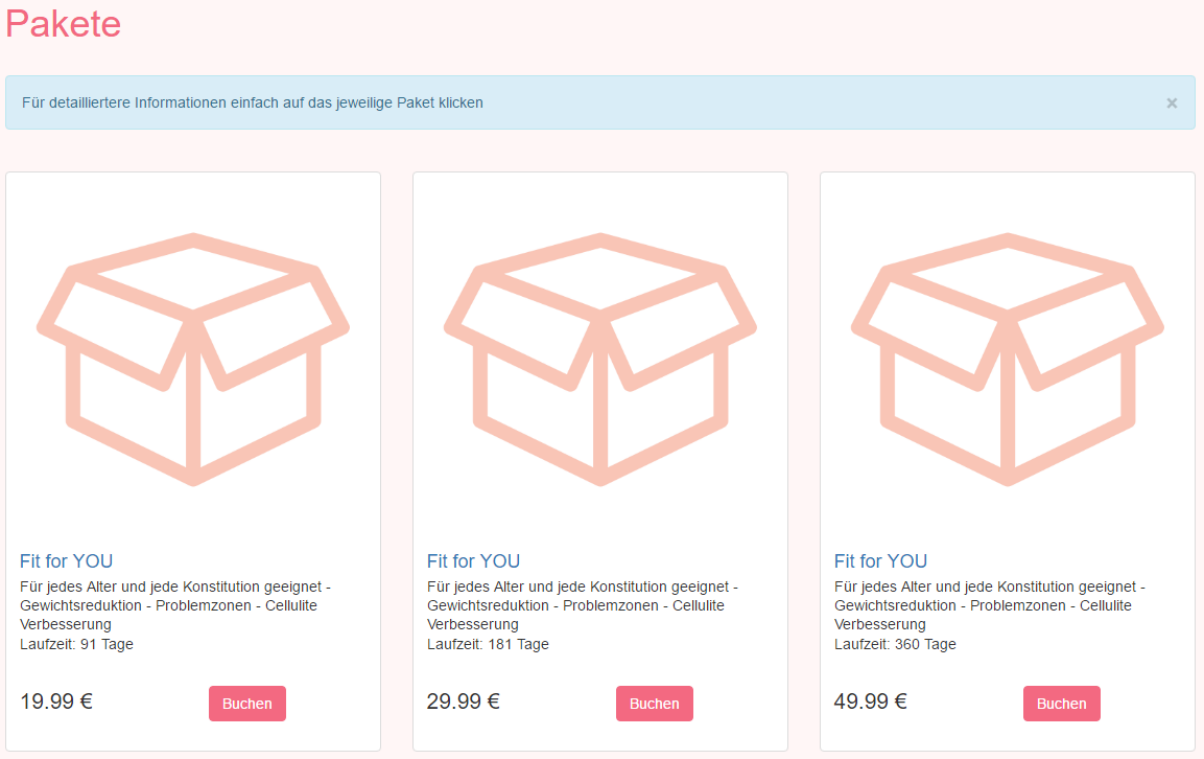


Abbildung 37 - Verkaufsseite: Pakete

Wenn der Webseitenbesucher detailliertere Informationen zu einem Paket erhalten möchte, hat er die Möglichkeit auf das Paketbild oder auf den Paketnamen zu klicken. Dabei öffnet sich ein Modal-Popup mit einer detaillierteren Beschreibung.

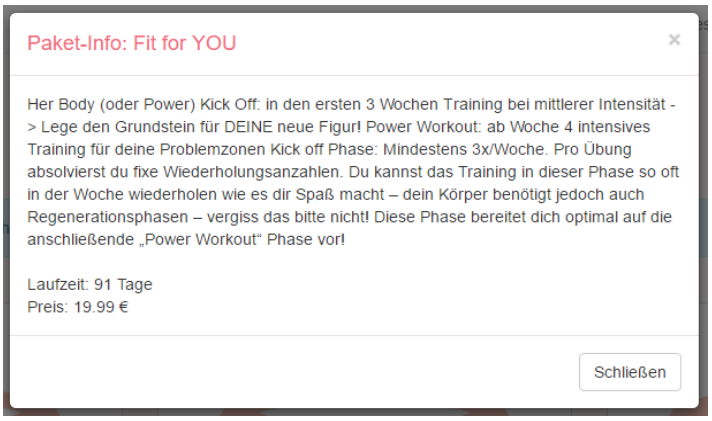


Abbildung 38 - Paket-Info

Wenn der Webseitenbesucher ein Paket buchen will, klickt er auf den „Buchen“-Button. Wenn der Webseitenbesucher noch nicht angemeldet ist, wird er aufgefordert sich anzumelden.

Wenn der Webseitenbesucher angemeldet ist, wird das gewählte Paket im Warenkorb abgelegt. Die Pakete, die im Warenkorb abgelegt werden, werden mit der Datenbank synchronisiert, damit der Benutzer später noch alle Pakete im Warenkorb vorfindet, wenn er seine Pakete erst später kaufen will.

3.1.3.9. Erfolge

Auf der Seite „Erfolge“ kann sich der Webseitenbesucher die Erfolgsgeschichte der „HerBody-Figur“ Julia durchlesen. Unterhalb kann sich der Webseitenbesucher noch die Erfolgsbilder von Julia ansehen und bekommt einen Eindruck davon, wie gut das „HerBody-Konzept“ funktioniert.
Auch auf dieser Seite sind Conversion-Buttons platziert, um den Webseitenbesucher wieder dazu zu verleiten, sich die Pakete auf der Paketseite anzuschauen und vielleicht auch eines zu buchen.

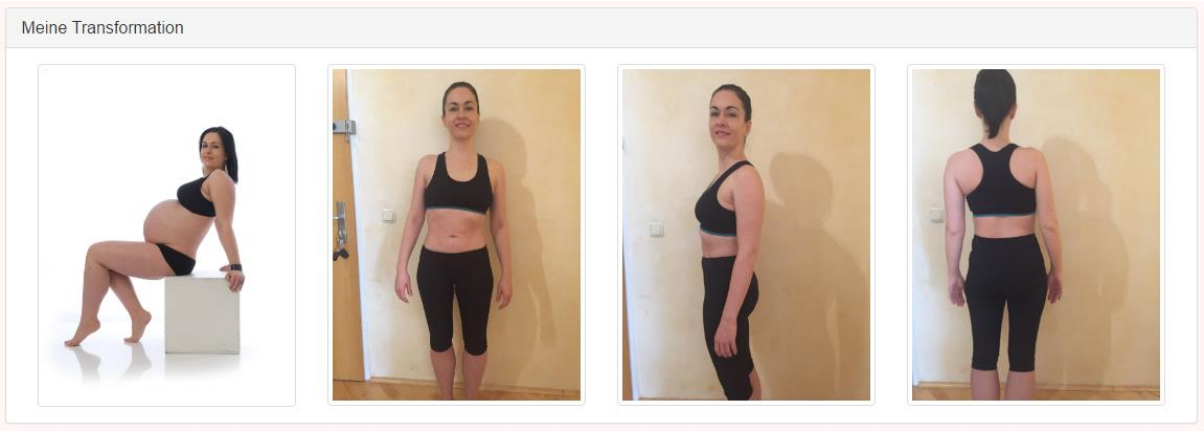


Abbildung 39 - Erfolge: Transformation

3.1.3.10. Wissenswertes

Auf der Seite „Wissenswertes“ kann sich der Webseitenbesucher einige Tipps und wissenswerte Dinge zum Thema Fitness & Ernährung durchlesen. Wenn der Webseitenbesucher weitere Fitness- & Ernährungstipps erhalten möchte, hat er die Möglichkeit, sich auf dieser Seite in den „HerBody-Newsletter“ einzutragen.

Diese Seite ist außerdem relevant für die On-Page Suchmaschinenoptimierung (siehe 4.9.1. Suchmaschinenoptimierung).

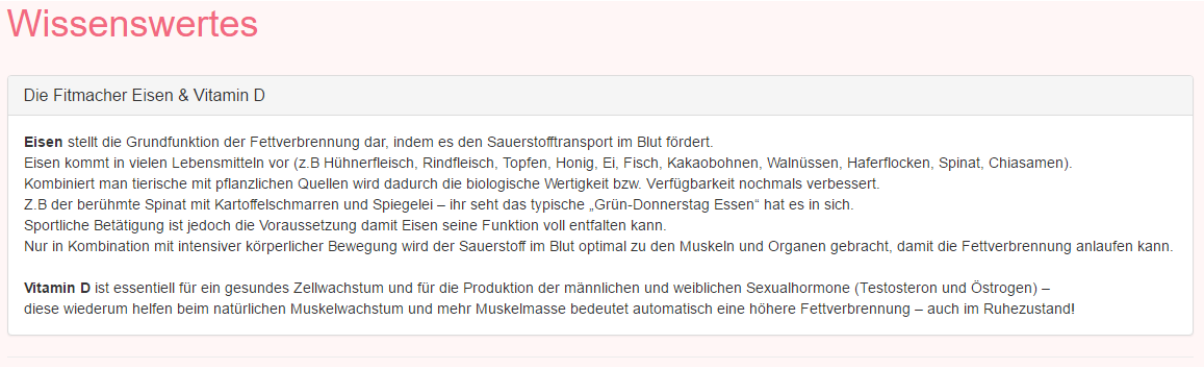


Abbildung 40 - Wissenswertes

3.1.3.11. FAQ`s

Auf der Seite „FAQ`s“ findet der Webseitenbesucher oft gestellte Fragen, die unterhalb auch beantwortet sind. Der Nutzen dieser Seite ist, dass der Webseitenbesucher mögliche Fragen beantworten kann, ohne dass er extra den Support kontaktieren muss und der Support nicht mit einer Vielzahl von doppelten Fragen konfrontiert wird. Wenn der Webseitenbesucher jedoch eine Frage hat, die noch nicht beantwortet worden ist, kann der Benutzer von dieser Seite auch direkt zum Kontaktformular wechseln und den Support kontaktieren.

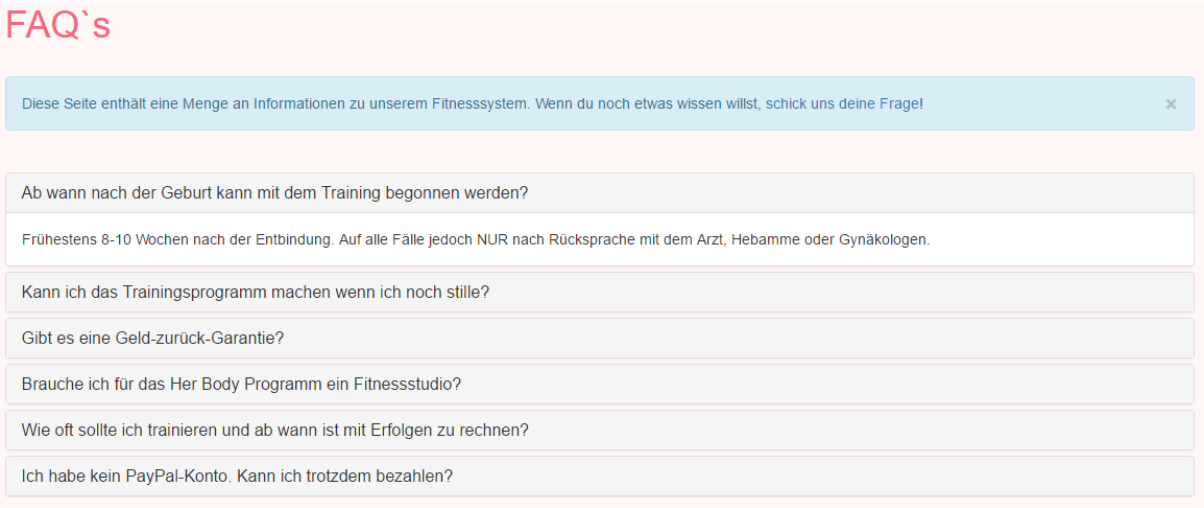


Abbildung 41 - FAQ`s

3.1.3.12. Mitgliederbereich

Der Webseitenbesucher hat nur Zugriff auf den Mitgliederbereich, wenn er sich anmeldet. Wenn der Kunde angemeldet ist, sieht er die Übersichtsseite des Mitgliederbereichs. Dort wird er mit seinem Vornamen begrüßt und hat auch die Möglichkeit sich wieder abzumelden.

Unterhalb kann der Kunde sein Profil ansehen und bearbeiten. Weiteres kann er seine gebuchten Pakete ansehen.

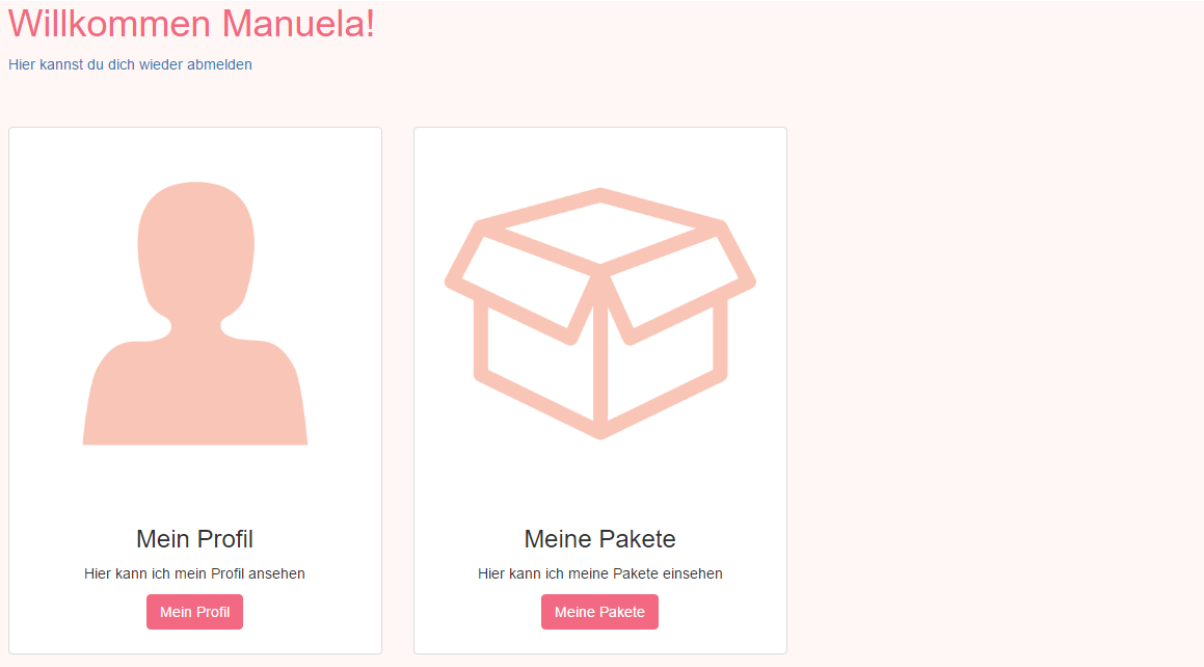


Abbildung 42 - Mitgliederbereich

3.1.3.13. Mein Profil

Auf der Seite „Mein Profil“ bekommt der Kunde einen Überblick über seine Profildaten. Weiteres wird es dem Kunden ermöglicht seine Profildaten und sein Passwort zu ändern.
Wenn der Kunde sein Passwort ändern möchte, muss er zuerst sein altes Passwort eingeben. Durch das Eingeben des richtigen Passworts kann der Kunde nun sein neues Passwort definieren.

Wenn die Profildaten geändert wurden, erhält der Kunde per E-Mail eine Bestätigung, dass seine Benutzerdaten geändert wurden.

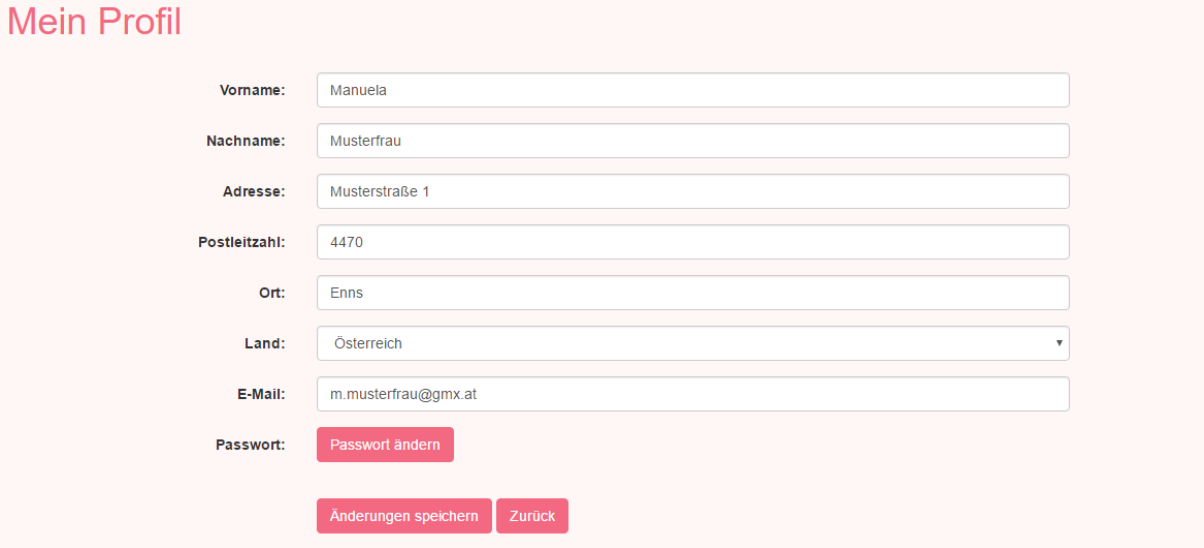


Abbildung 43 - Profilseite

3.1.3.14. Meine Pakete

Auf der Seite „Meine Pakete“ kann der Kunde seine gebuchten Pakete ansehen. Wenn der Kunde auf eines seiner gebuchten Pakete klickt, kommt er auf die Paketübersichtsseite.

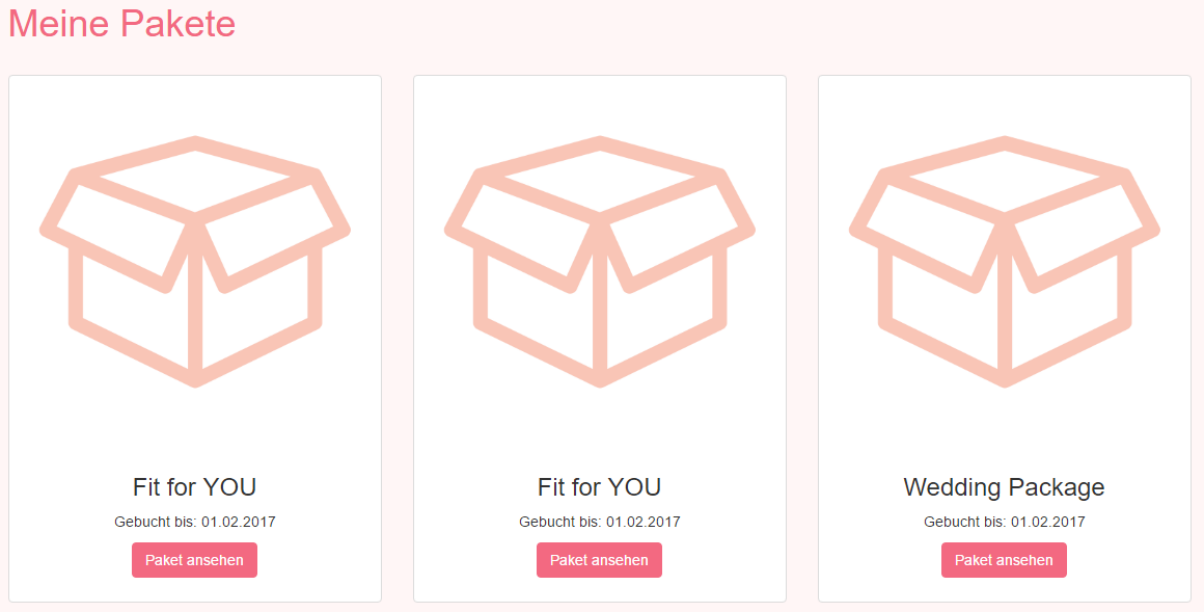


Abbildung 44 - Gebuchte Pakete

3.1.3.15. Paketübersicht

Auf der Seite „Paketübersicht“ kann der Kunde entweder seine Übungen oder seine Rezepte ansehen. Wenn der Kunde auf „Mein Training“ klickt, kann er seine Übungen ansehen. Wenn der Kunde auf „Meine Ernährung“ klickt, kann er seine Rezepte ansehen.

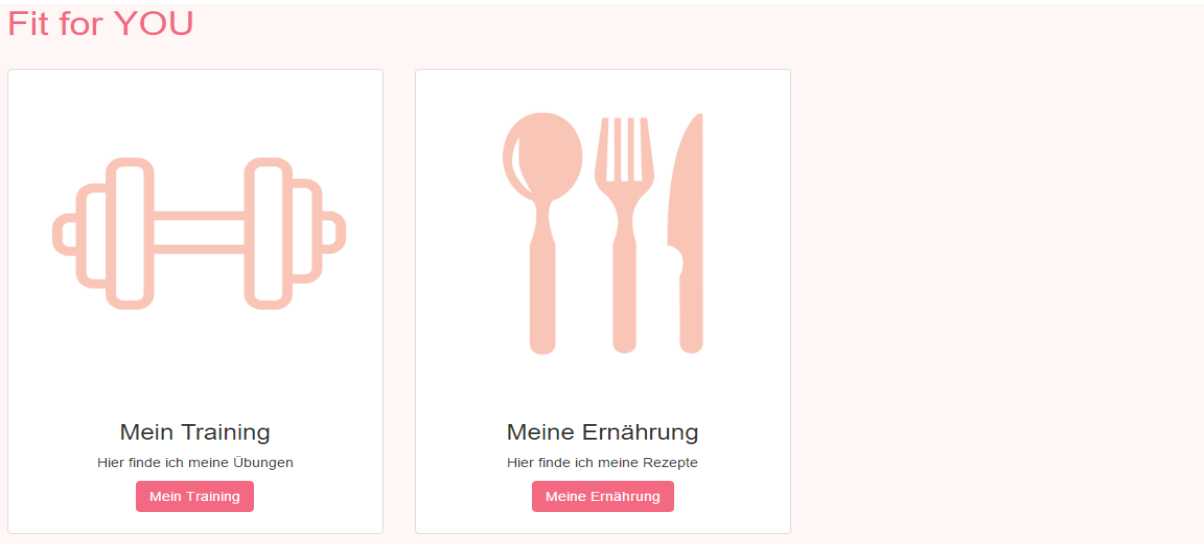


Abbildung 45 - Paketübersicht

3.1.3.16. Meine Übungen

Auf der Seite „Meine Übungen“ findet der Kunde alle Übungen, die zum ausgewählten Paket gehören. Wenn der Benutzer auf eine Übung klickt, öffnet sich ein Modal-Popup. In diesem Popup findet der Kunde eine kurze Übungserklärung und ein bis drei Bilder zur Ausführung.

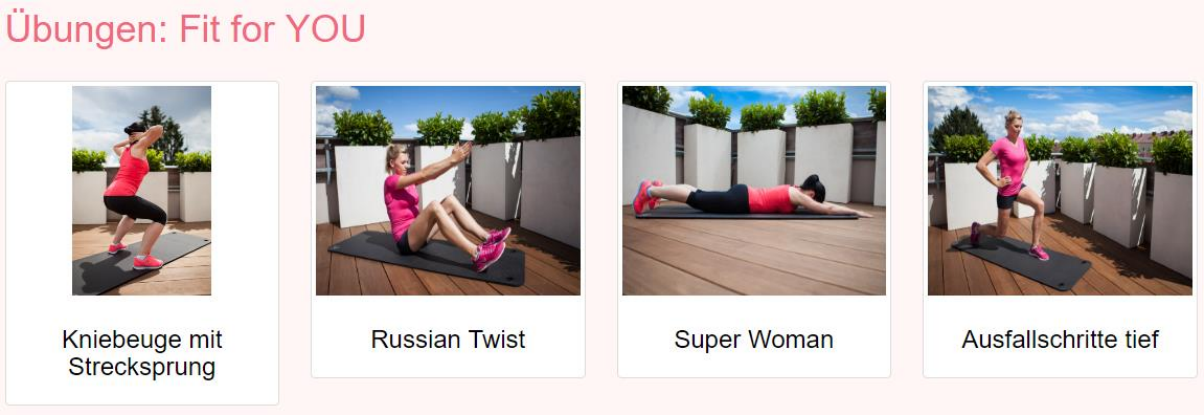


Abbildung 46 - Übungen

3.1.3.17. Meine Rezepte

Auf der Seite „Meine Rezepte“ findet der Kunde alle Rezepte, die zum ausgewählten Paket gehören. Wenn der Benutzer auf ein Rezept klickt, öffnet sich im Browser das Rezept in Form einer PDF-Datei. Der Kunde kann nun sein Rezept online ansehen und nachkochen oder auch herunterladen und ausdrucken.

Abbildung 47 - Rezepte

3.1.3.18. Warenkorb

Der Webseitenbenutzer hat nur Zugriff auf seinen Warenkorb, wenn er eingeloggt ist. Wenn der Webseitenbesucher angemeldet ist, sieht er alle Pakete, die er im Warenkorb abgelegt hat. Wenn er kein Paket im Warenkorb abgelegt hat, wird ihm angeboten zur Paketseite zu wechseln und sich ein Paket auszusuchen. Wenn Pakete im Warenkorb sind, kann der Webseitenbesucher diese auch entfernen.

Wenn der Webseitenbesucher mit seinem Warenkorb zufrieden ist, kann er seine „HerBody-Bestellung“ bezahlen. Dazu wird er auf die PayPal-Seite weitergeleitet. Dort kann er sich entweder mit seinem PayPal-Konto anmelden und die Summe begleichen oder er klickt auf „PayPal-Konto eröffnen“. Dort kann der Kunde je nach Land seine Kreditkartendaten oder seine Bankdaten eintragen und damit bezahlen. Nach der Bezahlung erhält der Kunde eine Bestellbestätigung per E-Mail.

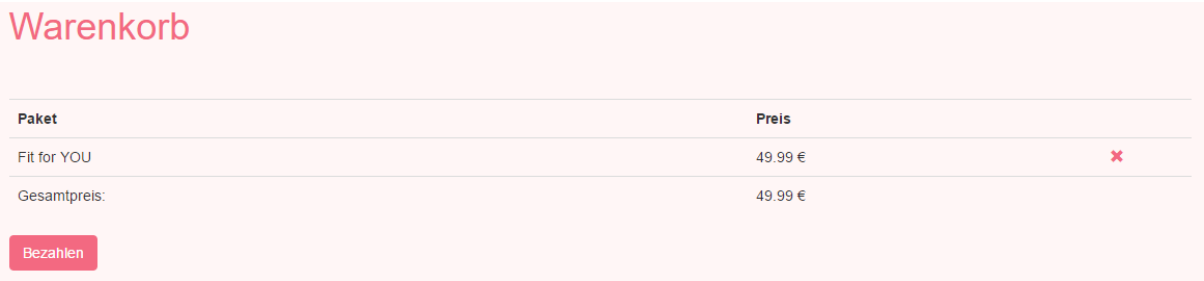


Abbildung 48 - Warenkorb

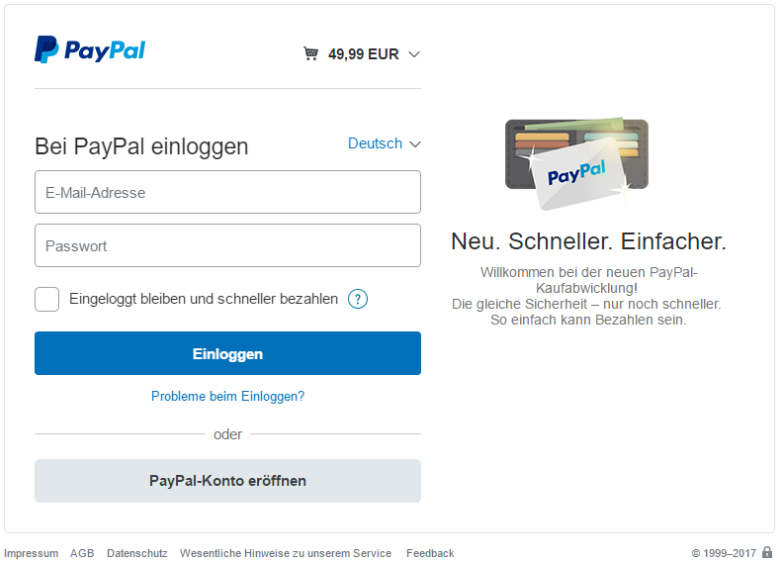


Abbildung 49 - PayPal Zahlung

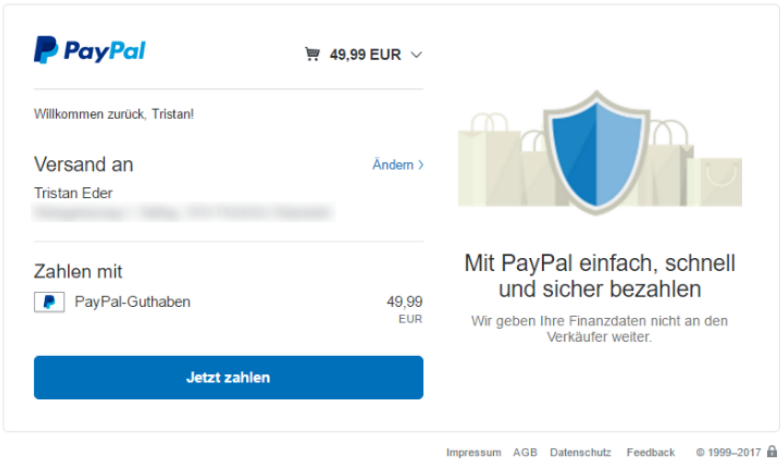


Abbildung 50 - PayPal Zahlung bestätigen

3.1.3.19. Login

Auf der Seite „Login“ hat der Webseitenbesucher die Möglichkeit seine E-Mail-Adresse und sein Passwort einzutragen. Wenn er auf den Login-Button drückt und der Login erfolgreich ist, wird der Kunde auf die Mitgliederbereich-Seite weitergeleitet. Wenn der Login nicht erfolgreich ist, wird der Webseitenbesucher mit entsprechendem Alert darauf aufmerksam gemacht.

Weiteres hat der Webseitenbesucher, der noch keine Login-Daten besitzt, die Möglichkeit auf die Registrierungsseite zu wechseln, und sich zu registrieren.

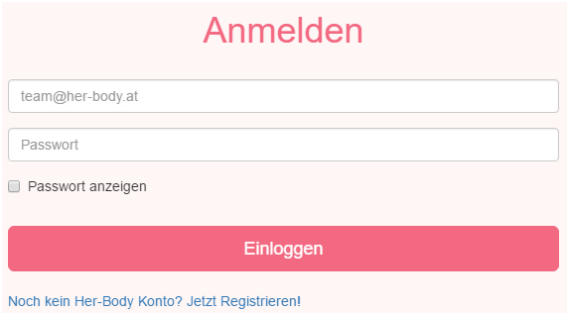


Abbildung 51 - Login

3.1.3.20. Registrierung

Auf der Seite „Registrierung“ hat der Webseitenbesucher die Möglichkeit sich bei HerBody zu registrieren und sich somit einen Account zu erstellen. Mit den erhaltenen Login-Daten kann sich der Kunde dann auf der Login-Seite anmelden. Um sich zu registrieren muss der Webseitenbesucher seinen Vornamen, seinen Nachnamen, sein Geburtsdatum seine Adresse, bestehend aus Straße, Postleitzahl, Ort und Land, seine E-Mail-Adresse und ein Passwort angeben. Bei erfolgreicher Registration wird der Kunde auf die Login-Seite weitergeleitet. Dort kann er sich dann mit seinen Login-Daten anmelden.

Weiteres hat der Webseitenbesucher, der seine Login-Daten schon besitzt, die Möglichkeit direkt auf die Login-Seite zu wechseln, und sich dort einzuloggen.

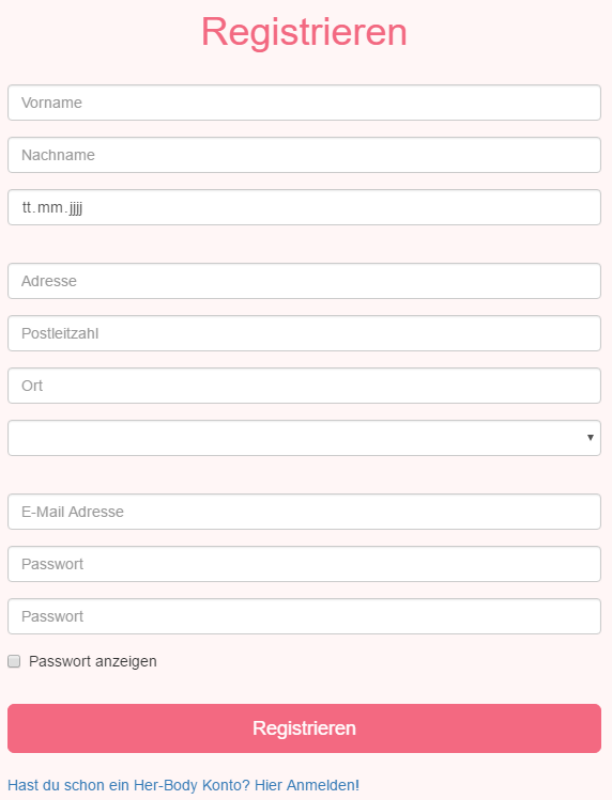


Abbildung 52 - Registrierung

3.1.3.21. Impressum

Auf der Seite „Impressum“ kann der Webseitenbesucher die Impressumsdaten ansehen. Wenn der Webseitenbenutzer die Betreiber der Webseite kontaktieren möchte, kann er dies auch mit dem Kontaktformular unternehmen, welches im Impressum verlinkt ist.

3.1.3.22. AGB

Auf der Seite „AGB“ kann der Webseitenbesucher die Allgemeinen Geschäftsbedingungen einsehen.

3.1.3.23. Datenschutz

Auf der Seite „Datenschutz“ findet der Webseitenbesucher alle Informationen zum Datenschutz.

3.1.3.24. Liefer- und Zahlungsbedingungen

Auf der Seite „Liefer- und Zahlungsbedingungen“ findet der Webseitenbesucher alle Informationen zu den Liefer- und Zahlungsbedingungen von HerBody.

3.1.3.25. Kontakt

Auf der Seite Kontakt findet der Webseitenbesucher ein Kontaktformular vor, mit dem er das „HerBody-Team“ einfach und direkt kontaktieren kann. Um eine Nachricht mit dem Kontaktformular abzusenden muss ein Name, eine E-Mail-Adresse, ein Betreff und eine Nachricht eingegeben werden.

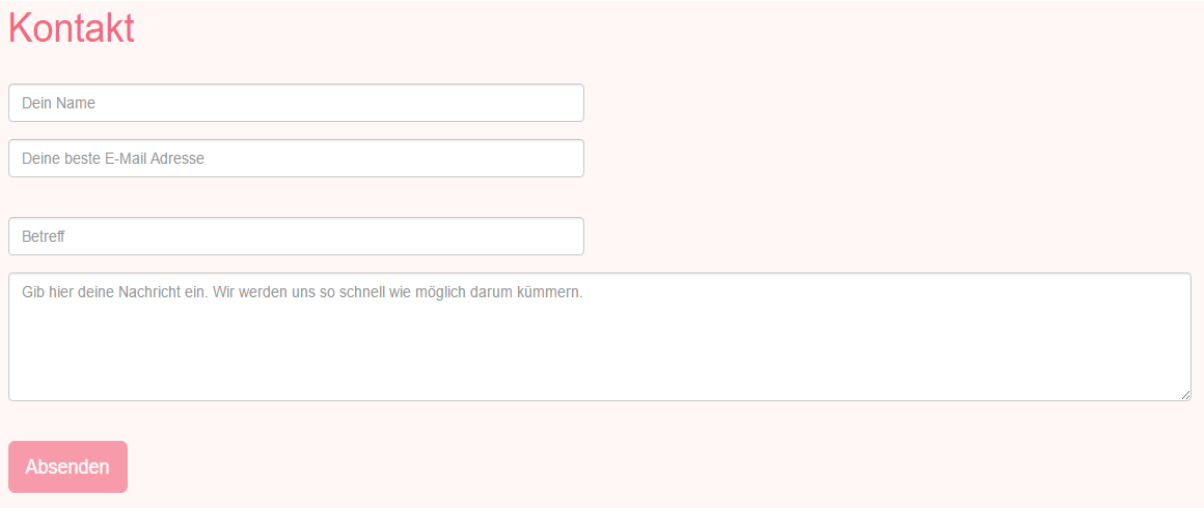


Abbildung 53 – Kontaktformular

3.1.4. WebAPI

Die WebAPI stellt jene Services, die von den beiden graphischen Oberflächen genutzt werden, um Daten von der Datenbank zu empfangen oder welche zu versenden, zur Verfügung. Ohne solche Services wäre es nicht möglich Inhalte auf den Oberflächen der App beziehungsweise der Webseite dynamisch anzuzeigen. Die WebAPI arbeitet für den User unsichtbar, also er bekommt von den Vorgängen zum Beispiel dem Ladevorgang von Daten praktisch nichts mit.

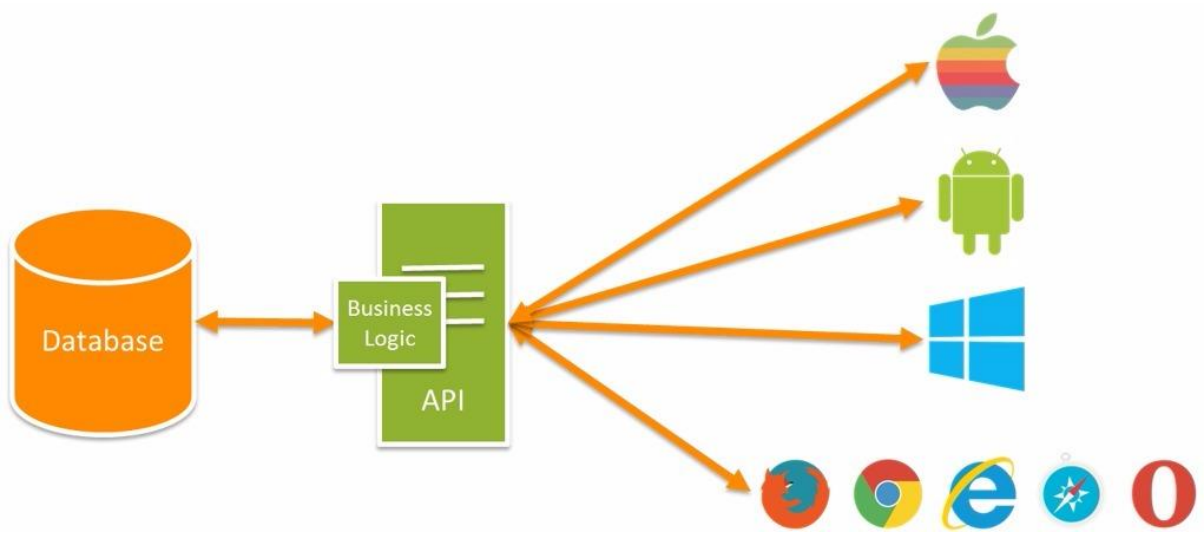


Abbildung 54 - WebAPI Kommunikation

Die obige Grafik veranschaulicht den Kommunikationsprozess einer WebAPI mit den Endgeräten und der Datenbank.

3.1.5. Admininterface

Das Admininterface ist ein Tool für die beiden Auftraggeber, dass es Ihnen ermöglicht Datenbankinhalte zu warten und neue zu erstellen.

Die folgenden Grafiken zeigen Ausschnitte der grafischen Benutzeroberfläche des Admininterfaces:

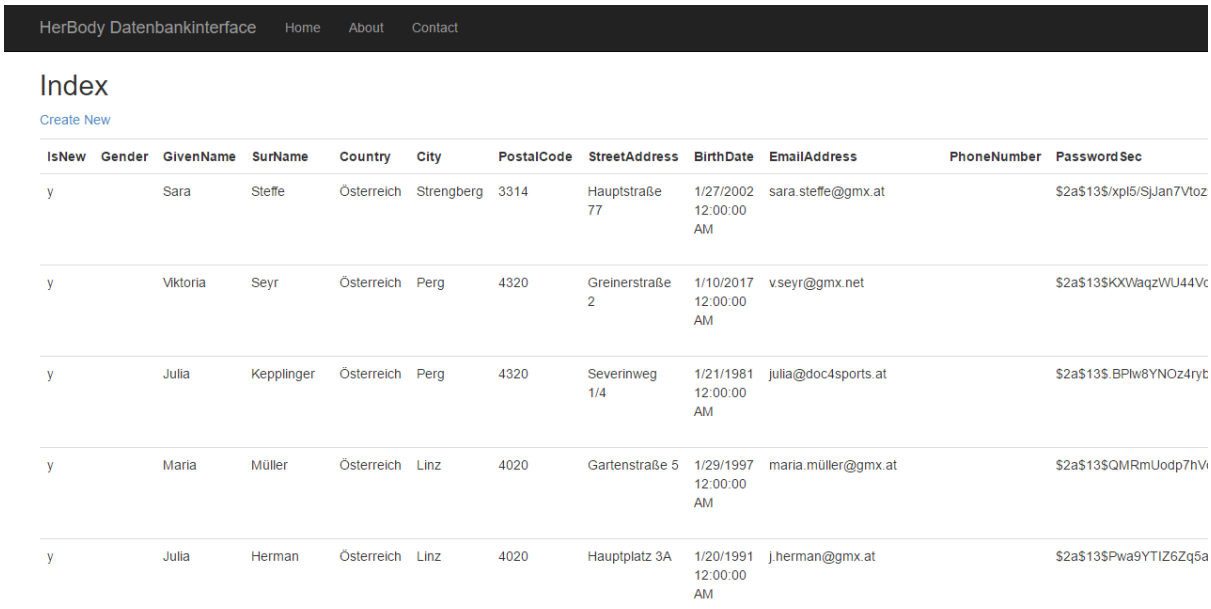


Abbildung 55 – Admininterface: Customer

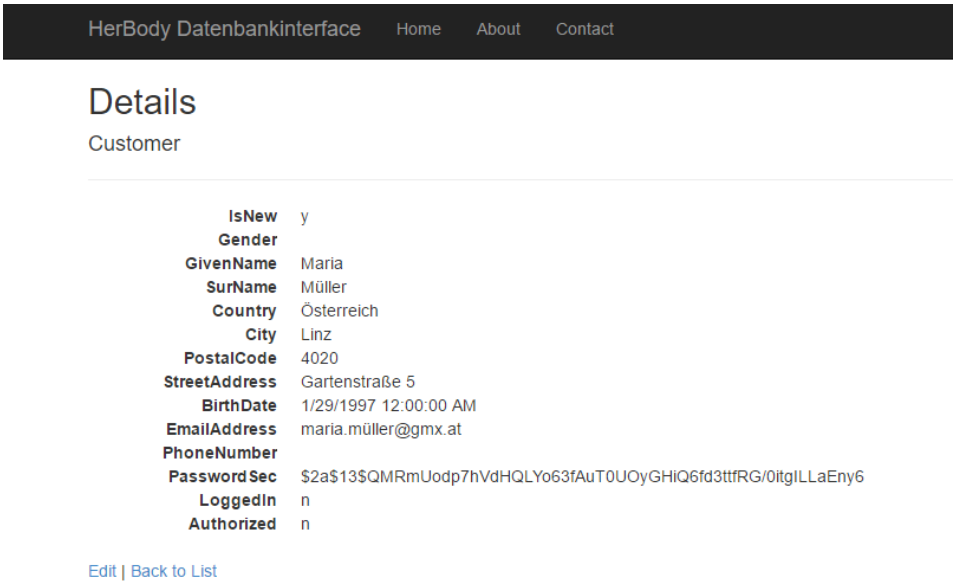


Abbildung 56 - Admininterface: Details

4. Realisierung und Umsetzung

4.1. Organisatorische Umsetzung

4.1.1. Projektorganisation

Die Projektorganisation wird durch folgende Grafik genau beschrieben:

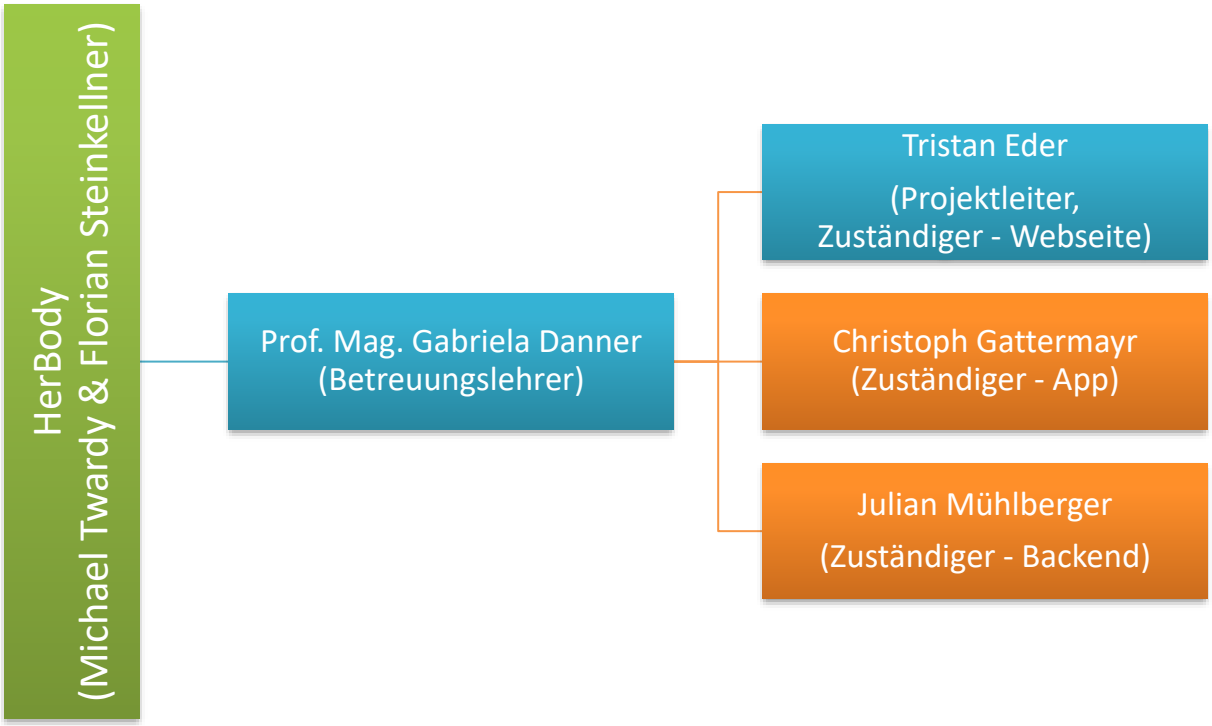


Abbildung 57 - Projektorganisation

4.1.2. Zuständigkeit

Die folgende Zuständigkeitsmatrix beschreibt die genaue Zuständigkeit bezüglich einzelnen Arbeitsaufgaben. Informiert wurde während der gesamten Durchführung der Arbeit über digitale Kommunikationswege wie Email und WhatsApp oder über persönliche Meetings in der Schulaula.

Arbeitsaufgabe	Eder	Gattermayr	Mühlberger	Twardy/Steinkellner	Danner
Kickoff – Meeting	MV	M	M	IM	I
Projektdefinition	MV	M	IM	IM	I
Pflichtenheft	MV	M	M	I	I
Planung Systemarchitektur	I	MV	M	I	I
Cloud-Konfiguration	I	MV	M	I	I
Entwicklung Backend	I	M	MV	I	I
Entwicklung Webseite	MV	I	I	I	I
Entwicklung App	I	MV	I	I	I
Testen	M	IM	MV	IM	I
Verschriftlichung	MV	IM	M	I	I
Projektübergabe	MV	M	M	IM	I

Tabelle 2 - IMV-Matrix

I … Informieren M … Mitarbeiten V … Verantwortlich

4.1.3. Ressource Management

4.1.3.1. Personalressourcen

Diplomand	Rolle
Tristan Eder	Projektleiter & Programmierer • Kommunikation mit dem Auftraggeber • Programmierung der Webseite • Debugging der WebAPI
Christoph Gattermayr	Programmierer & Tester • Programmierung der mobilen App • Debugging der WebAPI
Julian Mühlberger	Programmierer & Architekt • Programmierung der WebAPI und des Admininterfaces • Designen des ERDs

Tabelle 3 – Personalressourcen

4.1.3.2. Sachressourcen

Planung

- Projektentwicklungs-Buch

  Durch die im Unterricht verwendeten Lehrbücher war eine sehr detaillierte Planung der wissenschaftlichen Arbeit möglich.

- Laptops

  Die Verschriftlichung des Pflichtenhefts und anderer planungsrelevanten Dokumente wurde auf den privaten Notebooks der Diplomanden vorgenommen.

Implementierung

- Laptops

  Ebenso wie die Verschriftlichung der Projektplanung wurde die Implementierung auf den privaten Notebooks der Diplomanden durchgeführt.

Testung

- Laptops

  Alle drei Hauptkomponenten (WebAPI, WebApp und App) konnten auf den Diplomanden-Notebooks über den Browser getestet werden.

- Android-Phone

  Da jeder Diplomand ein Android-Phone besaß war es dem Team möglich, die App auch auf einem realen Smartphone zu testen. Außerdem wurde mit dem privaten Android-Phone getestet, wie sich die Webseite verhält, wenn sie mit dem Handy aufgerufen wird.

- Windows Phone (WP8)

  Das von der Schule geliehene Windows Phone diente ebenso zu Testung der mobilen App.

Tabelle 4 - Sachressourcen

4.1.3.3. Aufwandsschätzung

Die Aufwandsschätzung des Entwicklerteams wurde zu Beginn des Projekts abgegeben und basiert auf Erfahrungswerten der letzten Jahre. Sie setzt sich aus diesen Bereichen zusammen:

- 5h – Pflichtenheft
- 15h – Planung
- 130h – Implementierung
- 30h – Verschriftlichung

Pro Person wurden insgesamt 180 Stunden geschätzt woraus folgende Gesamtzeiten resultieren:

Aufgabenpaket	Stunden
Pflichtenheft	15
Planung	45
Implementierung	390
Verschriftlichung	90
Summe:	**540**

Tabelle 5 - Aufwandsschätzung

4.1.4. Zeitmanagement

Der Grundstein für das Zeitmanagement während der Abhandlung der wissenschaftlichen Arbeit war eine neue Meilensteinliste (kann im Anhang eingesehen werden), die erst nach Erstellung des Pflichtenheftes verschriftlicht wurde. Jene Meilensteinliste die dem Pflichtenheft entnommen werden kann, wurde auf Grund einer Änderung der Systemarchitektur verworfen. Es wurde versucht die terminlichen Fristen einzuhalten und die jeweiligen Module bis zu deren Fristen abzuhandeln, welches uns auch sehr gut gelang. Konnte eine Frist nicht termingerecht eingehalten werden, wurde das Arbeitspaket umgehend unter uns Diplomanden aufgeteilt und abgehandelt.

Für das persönliche Zeitmanagement wurde von jedem Diplomanden ein Excel-Sheet geführt in dem er Einträge mit Datum, Tätigkeit und Dauer vermerkte. Schlussendlich wurde in den persönlichen Excel-Sheets eine Summe der Stunden errechnet und alle spezifischen Einträge in einem gemeinsamen Dokument gesammelt und statistisch ausgewertet.

4.1.5. Kommunikationsmanagement

Für den förmlichen Schriftverkehr mit den Auftraggebern wurde zu Projektbeginn eine Emailadresse für das Entwicklerteam mit der erworbenen Domain angelegt, mit der wir mit den Auftraggebern kommunizierten und die nächsten persönlichen Meetings ansetzten. Im Gegensatz dazu, diente eine WhatsApp Gruppe für kurze Fragen. Zusätzlich zur Emailadresse des Entwicklerteams, war das Diplomandenteam unter der gemeinsamen Google-Mailadresse erreichbar.

Diese Grafik beschreibt die Kommunikationswege:

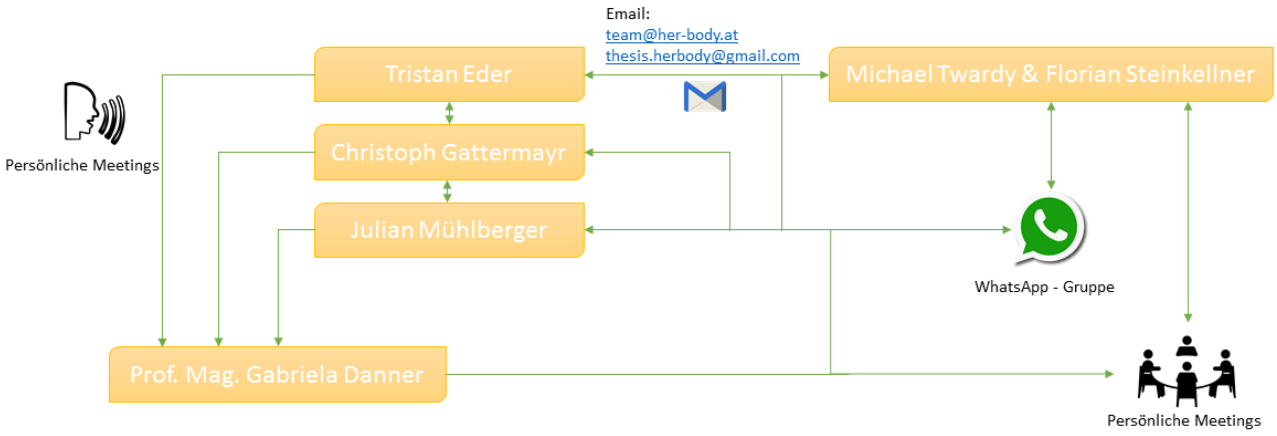


Abbildung 58 - Kommunikationswege

4.1.6. Pflichtenheft

Zum Projektbeginn wurde ein Projekthandbuch angefertigt, und die wichtigsten Punkte werden in diesem Kapitel angeführt.

Wichtige Ausschnitte des Pflichtenhefts können dem Anhang entnommen werden.

4.1.6.1. Meilensteine

Meilensteine	
23.09.2016	Arbeitsumgebung einrichten
30.09.2016	Fertigstellung des Pflichtenhefts
23.12.2016	Testfertiger Prototyp
05.01.2016	Testen des Gesamtsystems
31.01.2017	Entwurf der schriftlichen Diplomarbeit
05.04.2017	Projekt-Abgabe

Tabelle 6 – Meilensteine

4.1.6.2. Projektstrukturplan (vereinfacht)

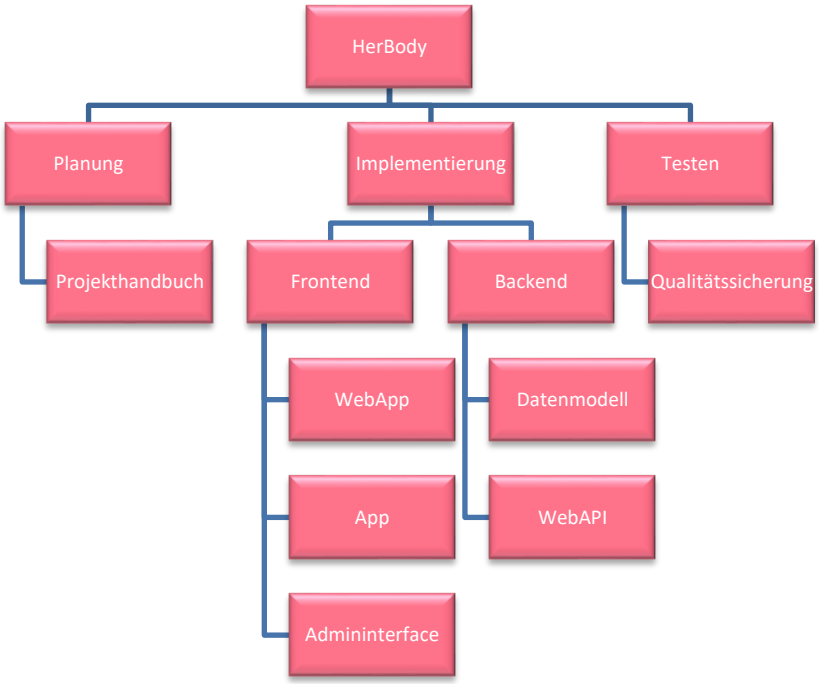


Abbildung 59 - Projektstrukturplan

4.1.6.3. Projektablaufplan

Juni 2016

- Erstbesprechung des Projekts
- Annahme des Projekts
- Einreichung des Projekts

Juli-August 2016

- Projektvorbereitungen
- Reservierung der Domains
- Buchen des Webhosters

September 2016

- Kick-Off-Meeting
- Pflichtenheft
- Projekthandbuch

Oktober 2016

- Einteilung der Aufgaben
- Erlernung der Technologien
- Einrichtung der Arbeitsumgebung

November 2016

- Vorbereitung der HTML Views
- Erstellung des Datenmodels
- Befüllung der Datenbank

Dezember 2016

- Implementierung der WebApp
- Implementierung der App
- Implementierung der WebAPI

Jänner 2016

- Prototyp der WebApp
- Prototyp der App
- Prototyp der WebAPI
- Konsumieren der Daten

Februar 2016

- Beginn der Verschriftlichung
- Letzte Korrekturen
- Debugging der WebAPI
- Implementierung des Admininterfaces
- Abschluss der Implementierung

März 2016

- Projektpräsentation
- Fertigstellung der Diplomschrift
- Projekt-Abgabe

4.1.6.4. **Projektvorgehensweise** [6]

Die Abwicklungsvorgehensweise des Projekts wurde an das agile Vorgehensmodell Scrum angelehnt.

Die Hauptmerkmale von Scrum sind:

- Es gibt nur drei Rollen
    - Product Owner (Vertritt die Interessen des Auftraggebers)
    - Scrum Master (Moderiert den Scrum Prozess)
    - Team (Gruppe von Entwicklern)
- Die Anforderungen werden im Product Backlog gesammelt
    - Product Backlog: Sammlung aller Anforderungen
- Die Produktentwicklung erfolgt in zeitlich klar definierten Zyklen
- Das Projektteam besteht aus gleichberechtigten Teammitgliedern

Abbildung 60 - Rollen bei Scrum

Bei Scrum wird die gesamte Abwicklung des Projekts in kleine Teile heruntergebrochen, die Sprints genannt werden. Am Beginn eines jeden Sprints werden Sprint Planning Meetings abgehalten. Dabei werden einzelne Aufgaben im Sprint Backlog festgelegt, die bis zum Ende des Sprints erledigt werden. Diese ausgewählten Aufgaben werden dem Product Backlog entnommen, welcher die Gesamtheit der zu erledigenden Aufgaben beinhaltet.

Ein Sprint dauert in der Regel ein bis vier Wochen und schließt mit einem Sprint Review Meeting ab, bei dem die erledigten Dinge mit dem Auftraggeber besprochen werden. Jeder Sprint muss eine lauffähige Software hervorbringen.

Während des Sprints werden jeden Tag kurze Meetings abgehalten, bei denen besprochen wird, welche Dinge seit dem letzten Daily Scrum abgeschlossen worden sind, an welchen Dingen bis zum nächsten Daily Scrum gearbeitet wird und welche Probleme in der Zwischenzeit aufgetreten sind.
Zwischen den Sprints können die positiven und negativen Aspekte des Arbeitsprozesses beim Sprint Retrospektive herausgearbeitet und analysiert werden.

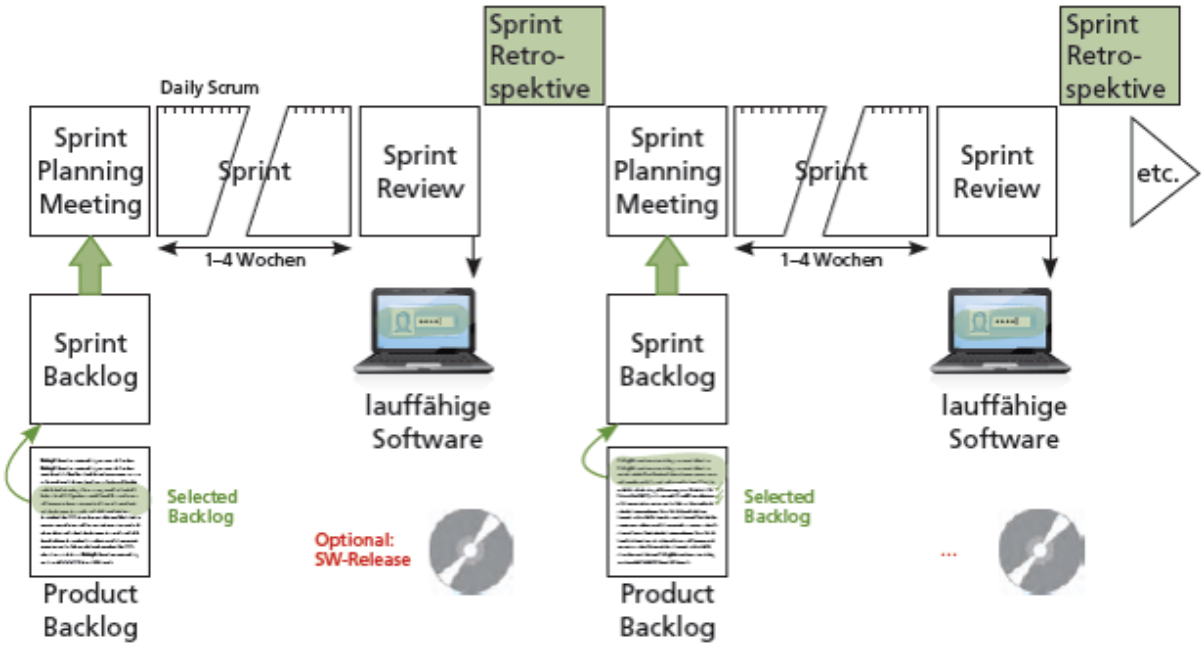


Abbildung 61 - Scrum Ablauf

Durch die Anlehnung der Projektabwicklung an Scrum, konnten wir viel effektiver und schneller arbeiten und durch die täglichen Meetings an Projekttagen konnten wir aufgetretene Probleme schnell und effektiv lösen.

4.2. Technologien

4.2.1. ASP.NET

Eine von Microsoft entwickelte Technologie, welche es ermöglicht webbasierte Applikation und Services zu implementieren. So ist es beispielsweise möglich einen REST-Service in Verbindung mit dem Entity-Framework zu erstellen, wodurch Clients Daten einer spezifischen Datenquelle Abfragen können. Außerdem können Datenbankinterfaces einfach und schnell erstellt werden. Darüber hinaus können auch einfach Webseiten die mit HTML5, CSS und JavaScript implementiert wurden um sogenannte CSHTML Seiten erweitert werden. In diesen Seiten verbindet man C# - Code mit HTML – Code.
ASP.NET wurde für die Implementierung der WebAPI und des Admininterfaces verwendet. Weiteres wurde die WebApp als ASP.NET Web Application angelegt.

IDEs zur Entwicklung von ASP.NET:
Die Entwicklung einer ASP.NET Anwendung kann mit Visual Studio vorangetrieben werden.

Ein großer Vorteil von ASP.NET Anwendungen ist die einfach Veröffentlichung auf die Azure Cloud. Wie genau sich eine solche Applikation veröffentlichen lässt, kann unter dem Punkt Release Verwaltung – Microsoft Azure nachgeschlagen werden (siehe 6. Release Verwaltung). [7]

4.2.2. Apache Cordova

Einleitung – Was ist ApacheCordova?
Mittels des Frameworks Apache Cordova wir die Entwicklung von plattformübergreifenden mobilen Anwendungen, kurz Apps ermöglicht. Die geschriebene Anwendung läuft schlussendlich unter den drei führenden Betriebssystemen auf mobilen Geräten - Android, IOS und Windows. Die durch das Framework programmierten Anwendungen sind weder webbasiert, weil sich die Apps in den plattformspezifischen Stores verteilen lassen, noch nativ, da das Layout mittels Webtechnologien und nicht nativ geschrieben wird. Für das Frontend wird HTML5 und CSS verwendet, wohingegen JavaScript für die Funktionalität im Hintergrund verwendet wird.

Weiteres wird durch JavaScript eine Verbindung zu nativen Klassen und vorgefertigten nativen Modulen (Plugins) gewährt. Durch diese Kombination der drei Sprachen wird es dem Entwickler ermöglicht, die native App-Programmierung mit Sprachen wie Java, Swift oder C# zu umgehen.
Das Apache Cordova Framework wurde für die Implementierung der App verwendet, da es sich als Framework für plattformübergreifende Entwicklung am besten anbietet. [8]

IDEs zur Apache Cordova Entwicklung:
Die Programmierung einer Apache Cordova App ist in kostenlosen Entwicklungsumgebungen wie Eclipse oder Netbeans möglich. Darüber hinaus kann die Entwicklung auch mit der Microsoft IDE Visual Studio durchgeführt werden, welche bei unserer wissenschaftlichen Arbeit auch verwendet wurde.

In unserem Fall kam Visual Studio 2015 mit der Schullizenz zum Einsatz.
Die genaue Beschreibung der verwendeten Entwicklungsumgebung findet sich unter dem Punkt Entwicklungsumgebungen (siehe 4.4. Entwicklungsumgebungen).

Zielplattformen:

- Android
- WP8
- WP10
- IOS

Unterstützt werden jedoch noch viele weitere Plattformen wie zum Beispiel BlackBerry.

Testen:

Wird für Visual Studio die Apache Cordova Erweiterung installiert, wird dem Entwickler im Browser ein Emulator für Android-, IOS- und Windowsgeräte zum Testen zur Verfügung gestellt. Unter der Voraussetzung man verwendet wie wir die Entwicklungsumgebung Visual Studio, bieten sich folgende Varianten des Debuggings für das Programm:

1. Debugging via Browser

   In JavaScript kann mittels des Schlüsselworts „debugger“ eine Break Point im Code gesetzt werden, an dem im Falle des Funktionsaufrufs währen der Laufzeit pausiert wird. So kann über die Browser Konsole (Google Chrome F12) getestet und analysiert werden.

2. Debugging via Visual Studio

   Über die IDE kann auch ein Break Point im Programm gesetzt werden an dem ebenfalls zur Laufzeit pausiert wird und einem die IDE durch das Debugging unterstützt.

Optimal-Anforderungen für die Entwicklung: [9, 10]

- Betriebssystem: Windows 8, 10
- IDE: Visual Studio 2015, Eclipse, Netbeans
- RAM: 4GB
- Disk Space: 30GB
- IDE-Erweiterung: Apache Cordova (inkl. Java Runtime Environment 1.8)

4.2.3. Microsoft Azure

Mit Microsoft Azure stellt Microsoft eine der derzeit größten Cloud-Plattformen zur Verfügung, die eine Sammlung an Cloud-Services beinhaltet und es Entwicklern ermöglicht ihre Anwendung im globalen Netzwerk zu veröffentlichen und zu verwalten. Weiteres werden dem Programmierer keine Einschränkungen hinsichtlich Tools, Anwendungen und Frameworks vorgeschrieben. Darüber hinaus können ganze Server, virtuelle Maschinen oder Datenbanken angemietet und verwendet werden.

Ein großer Vorteil von Microsoft Azure ist die hohe Skalierbarkeit mit der jederzeit die Ressourcen erweitert oder wieder verringert werden können. Auch sehr nützlich ist die statistische Auswertung, mit der zum Beispiel auf Anzahl der Zugriffe oder bestimmte Zugriffszeiten gewisser Ressourcen eingesehen werden kann.

Auf Microsoft Azure wurde unser gesamtes Backend gehostet, da Microsoft Azure zurzeit der beste Anbieter für Cloud-Plattformen ist und die Erweiterung von Ressourcen bei Azure ganz einfach möglich ist. Somit lässt sich das System ganz einfach hochskalieren, wenn die Kundenanzahl steigt. [11]

4.3. Frameworks

4.3.1. AngularJS

Das von Google entwickelte Open Source Framework AngularJS ist ein JavaScript-Framework, dass auf die Programmierung von Webapplikationen, vor allem Single-Page-Applications ausgelegt ist und zum größten Teil nach dem MVVM (Model View ViewModel) Pattern aufgebaut ist. Eine Single-Page-Application ist eine Applikation bei der der größte Teil der benötigten Daten beim Starten der Anwendung geladen wird und bei angeforderter URL-Änderung nicht die komplette Seite aktualisiert wird, sondern nur die neuen benötigten Daten via Ajax nachgeladen werden.

Zu den großen Vorteilen von AngularJS zählt einerseits, dass das Framework eine Erweiterung von HTML ist, andererseits, dass es sehr gut testbar ist.
Darüber hinaus kann die Verwendung eines JavaScript-Objekt (POJO) als Datenstruktur zu den Vorteilen zählen. Somit wird einem die Programmierung von Model-Klassen, die zusätzliche Bibliotheken wrappen erspart.
AngularJS wurde bei der Implementierung der WebApp und der App verwendet, da dies derzeit die geeignetste Möglichkeit ist, Single Page Applications zu entwickeln. [12]

4.3.2. Newtonsoft

Newtonsoft JSON ist eine Bibliothek, welche das Handling mit JSON erleichtert und erweitert. Mit ihr ist es möglich, Objekte aber auch ganze Collections einfach zu einem JSON-Paket zu konvertieren beziehungsweise auch umgekehrt.

```
string json = @"{
                    'EmailAddress':'james@example.com',
                    'Active':'true',
                    'BirthDate':'01.01.1990',
                    'Roles': [
                        'User',
                        'Admin'
                    ]
                }";
Customer c = JsonConvert.DeserializeObject<Customer>(json);
Console.WriteLine(c.EmailAddress);
```

Abbildung 62 - Newtonsoft

Die Bibliothek wurde bei der Implementierung der WebAPI verwendet und kann leicht mittels NuGet in Visual Studio installiert werden. [13]

4.3.3. CORS

Cross-Origin-Resource-Sharing ermöglicht die Verwendung von, durch die Same-Origin-Policy untersagten, Cross-Origin-Requests. Im Grund genommen ermöglicht dieser Mechanismus also einem Webserver auch auf andere Domains zuzugreifen. Dies wird intern mittels eines Header Fields – dem sogenannten Access-Control-Allow-Origin. In diesem Feld muss der Link vom Server stehen, an welchen man Cross-Origin-Requests erlauben möchte. Diese Requests können durch Erweiterung eines Access-Control-* Felds weiter eingeschränkt werden. Beispielsweise durch Access-Control-Allow-Methods: GET. In der Schnittstelle zwischen Datenbank und den Clients müssen wir diese COR ebenfalls erlauben, dies ist im folgenden Codestück ersichtlich.

```
namespace HerBody.Api.Controllers
{
    [EnableCors(origins: "*", headers: "*", methods: "*")]
    public class BasketController : ApiController
    {

        ....
```

Abbildung 63 - CORS Annotation

Somit erlauben wir von allen Domains, mit allen möglichen Headers und allen erdenklichen http-Verben, den Zugriff auf unsere Ressource.

Abbildung 64 - CORS

CORS wurde bei der Implementierung der WebAPI verwendet, um Cross-Origin-Requests zu ermöglichen, damit von allen anderen Domains auf die WebAPI zugegriffen werden kann. Zum Beispiel von der Domain www.her-body.at. [14]

4.3.4. Bootstrap

Bootstrap ist ein sehr populäres CSS-Framework, welches zum Beispiel einem Webentwickler ermöglicht, responsive Anwendungen zu entwickeln. Das Framework ist kostenlos und wurde als Open-Source Projekt veröffentlicht.

Der Code muss dabei nur einfach verfasst werden und wird dann von Bootstrap automatisch an jedes Gerät angepasst.

Bootstrap wurde von den Twitter-Entwicklern Mark Otto und Jacob Thornton ins Leben gerufen und sollte ursprünglich dazu dienen intern ein einheitliches System zu verwenden, um Inkonsistenz zu vermeiden und sich damit den immensen Wartungsaufwand zu sparen. [15]

Bevor die Entwickler das Framework als Open-Source-Framework veröffentlichten war es als „Twitter Blueprint“ bekannt.
[16]

Bootstrap wurde bei der Implementierung der WebApp verwendet, um die Webseite responsive zu gestalten. Somit können die Kundinnen die Webseite auch problemlos vom Smartphone aus öffnen und erhalten dabei eine mobile-optimierte Ansicht.

4.3.5. Ionic

Ionic ist ein sogenanntes Frontend-Framework zur Entwicklung von mobilen Applikationen. Ähnlich wie das Bootstrap-Framework bringt es eine Vielfalt von fertigen GUI-Komponenten wie Buttons, Listen oder etwa Inputboxen mit sich.

Zusätzlich verfügt es über ein eigenes Kommandozeilentool, das den Code als App verpacken kann und sogar Simulatoren starten kann. Ein weiteres hilfreiches Tool das von den Herstellern des Frameworks kommt ist der online GUI-Designer Ionic Creator, mit dem ganzen Layout und Templates gestaltet werden können. [17]

Das Ionic-Framework wurde bei der Implementierung der App verwendet, um all die von Ionic zur Verfügung gestellten Komponenten zu verwenden, um die App ansprechend zu gestalten.

4.4. Entwicklungsumgebungen

4.4.1. Visual Studio

Visual Studio ist eine von Microsoft entwickelte IDE und zählt derzeit zu den führenden Entwicklungsumgebungen am IDE-Markt. VS15 bietet eine breite Auswahl an Erweiterungen und Tools wie zum Beispiel „Xamarin“ oder „Tools for ApacheCordova“.
Die Entwicklungsumgebung kann sehr spezifisch konfiguriert werden und stellt einen ausgezeichneten Debugg-Modus zur Verfügung.

Hardwareanforderungen [18]

- 1,6-GHz-Prozessor oder schneller
- 1 GB RAM (1,5 GB bei Ausführung auf einem virtuellen Computer)
- 10 GB verfügbarer Festplattenspeicher
- Festplattenlaufwerk mit 5400 U/min
- DirectX 9-kompatible Grafikkarte (Auflösung von 1024x768 oder höher)

Softwareunterstützung

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7 SP1
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 SP1

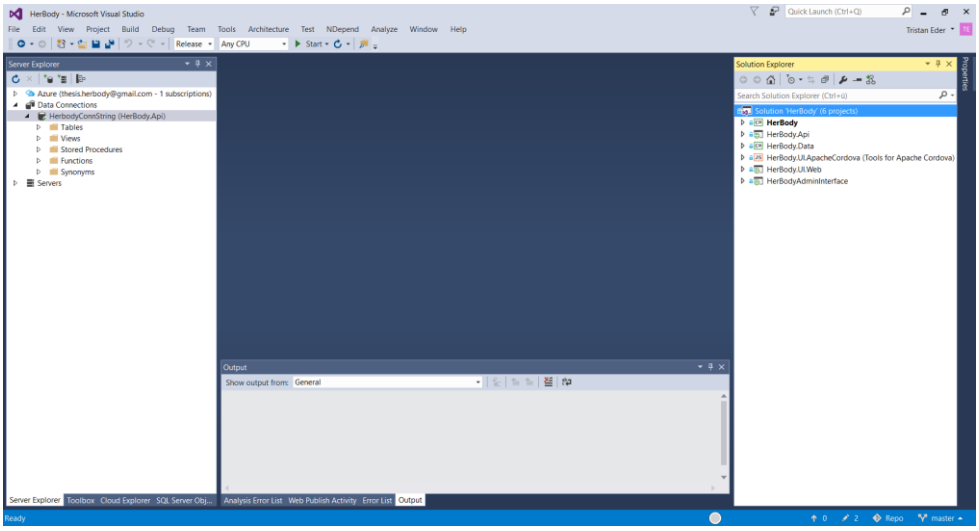


Abbildung 65 – Microsoft Visual Studio

4.4.2. Microsoft SQL Management Studio

Das Microsoft SQL Management Studio ist ein von Microsoft entwickeltes Datenbanken-Tool, dass entweder einzeln oder in Verbindung mit einem MS-SQL-Server installiert werden kann. Es bringt spezielle Datenbank-Funktionen mit sich um die Konfiguration und Wartung einer Datenbank zu erleichtern. Als ersten Schritt wird die Verbindung zu Datenbank hergestellt, wo man sich mittels Administrationszugang anmelden kann. Anschließend kann die Datenbank definiert, konfiguriert und gewartet werden.

Hardwareanforderungen [19]

- X86-Prozessor: 1.0GHz oder X64-Prozessor: 1.4GHz
- 1GB verfügbaren Festplattenspeicher

Softwareunterstützung

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7 SP1
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 SP1

4.5. Versionisierung

Zur Versionsverwaltung ist ein Git Repository bei Bitbucket.org angelegt worden. In diesem Git Repository ist das Projekt abgelegt. Somit ist es für jeden zugänglich und bei Änderungen sofort gesichert.

Als Git-Client wurde SmartGit verwendet. Somit war eine einfache Handhabung der Versionisierung garantiert.

Da jedes Projektmitglied ein eigenes Teil-Projekt in der HerBodySolution gehabt hat, und somit meist nur in seinem eigenen Teil-Projekt gearbeitet hat, funktionierte die Git-Synchronisation problemlos, da so keine Merge-Konflikte auftreten konnten.

4.6. Systemarchitektur

Die Systemarchitektur setzt sich aus den folgenden Komponenten zusammen:

- Mobile plattformübergreifende Apache Cordova App
- Angular JS Single Page Web Application
- APS.NET WebAPI
- Microsoft SQL Server
- Microsoft Azure Cloud
- WebApp Ressource
- Admininterface

Beschreiben lässt sich die Architektur am besten durch folgende Grafik:

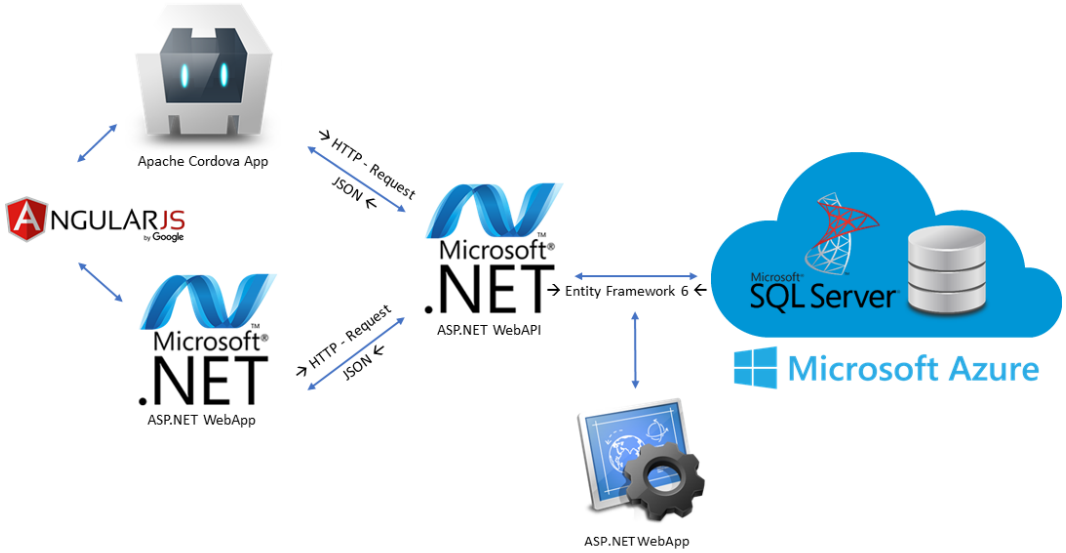


Abbildung 66 - Systemarchitektur

Als Frontend dienen einerseits eine Apache Cordova App, andererseits eine ASP.NET WebApp, die als Webseite fungiert. Die Schnittstelle zur Datenbank stellt eine ASP.NET WebAPI dar, mit der die beiden Frontends via HTTP-Requests kommunizieren und als Antwort Daten im JSON Format zurückgeschickt bekommen. Die WebAPI referenziert auf das Datenmodell, dass mit Hilfe des Entity Frameworks in der Version 6 erzeugt wurde und kommuniziert so mit dem Microsoft SQL Server. Der SQL Server wurde wiederrum auf der Cloud Plattform Microsoft Azure angemietet. Weiteres wurde auf MS Azure eine WebApp angelegt auf die WebAPI veröffentlicht wird und somit im World Wide Web zugänglich gemacht wird.

4.7. Schnittstellen

4.7.1. WebAPI

Als Schnittstelle zwischen dem Backend und den beiden Frontends wurde eine ASP.NET WebAPI verwendet, da diese eine sehr geeignete Lösung darstellte. Eine ASP.NET WebAPI ist ein Framework für die Erstellung von verschiedensten http-Diensten, auf welche eine Vielzahl an Clients bedient werden können. WebAPIs können mittels der Http-Methoden (GET, POST, PUT, DELETE) abgefragt beziehungsweise aufgerufen werden. Außerdem ersparen RESTful-Services gegenüber SOAP-Services das Konvertieren von Programmdaten zu einem XML Datenschema, weil RESTful-Service meistens mittels JSON kommuniziert. Diese Einsparung des Konvertierens bringt performancetechnische Vorteile mit sich.

4.7.1.1. REST

Ein Server stellt einen Dienst zu Verfügung, welcher bei Bedarf angefragt werden kann. REST Services bieten hierbei eine einfache Lösung – die nötige Infrastruktur ist meistens überall bereits vorhanden und die Web-Dienste haben REST grundsätzlich implementiert, beziehungsweise verwenden diesen Aufsatz bereits unbewusst. Im Grunde genommen basiert das ganze World Wide Web auf REST-Services – den REST besagt, dass die URI die Daten bestimmt. So kann der der Client die Daten vom Server mittels http-Verben anfordern.

4.7.1.2. HTTP-Requests

- GET (READ)

  Ist eine, über die URI definierte, Abfrage an einen Server. Sie soll keine Werte verändern, sondern diese nur auslesen. Die Abfrage kann mittels Übergabeparametern gesteuert werden.

```
GET /api/customer?Email=max@muster.com HTTP/1.1
Host: https://herbodyapi.azurewebsites.net
```

Abbildung 67 - HTTP GET-Request

- POST (CREATE)

  Kann ebenfalls über die URI definiert werden, jedoch auch über gewisse Header Fields. POSTS werden hauptsächlich dazu verwendet, neue Datensätze zu erzeugen. Außerdem können Daten im Body mitübergeben werden. Beispielsweise ein neuer Customer im JSON Format.

- PUT (UPDATE)

  Wird normalerweise verwendet, um Dateien auf einem Server hochzuladen. Im Zusammenhang mit der API wird PUT verwendet um Datensätze zu erneuern. Ein PUT funktioniert ähnlich zu einem POST. Die zu erneuernden Daten werden im Body übergeben. Um den Datensatz, welcher erneuert werden soll zu identifizieren, wird beispielsweise die ID in der URI mitgesendet.

- DELETE

  Löscht einen vorhandenen Datensatz, welcher über die in der URI übergebende ObjectID ermittelt wird.

4.7.1.3. HTTP-Response-Codes

Jede Anfrage auf den Server wird mit einem jeweiligen Response Code und eventuellen Daten, welche angefragt wurden, beantwortet. Der Server teilt somit dem Client mit, ob jeweilige Ressourcen beispielswiese nicht verfügbar sind, er eine Authentifizierung benötigt oder ob etwas erfolgreich funktioniert hat.

1xx – Informationen

Wenn die Bearbeitung einer Anfrage länger als normal dauert, gibt es Server, die eine Zwischenantwort senden. Dies hat den Grund, dass viele Clients nach einer zulangen Zeitspanne zwischen Anfrage und Antwort davon ausgehen, dass ein Fehler bei der Bearbeitung und/oder Übertragung entstanden ist.

2xx – Erfolgreiche Operation

Wenn die Bearbeitung beziehungsweise die gesamte Anfrage erfolgreich war, schickt ein Server eine 2xx Antwort.

3xx – Umleitung

Wenn es erforderlich ist, um eine Anfrage zu beantworten, einen anderen Server beziehungsweise eine andere Ressource zu erreichen, teilt der bereits angefragte Server dem Client mittels einer 3xx Nachricht eine Um- oder Weiterleitungsantwort mit. Die neue Adresse der gesuchten Ressource wird im Location-Header-Field eingetragen.

4xx – Client-Fehler

Entsteht bei der Bearbeitung einer Anfrage ein Fehler, dessen Schuld sich den Clients zuweisen lässt, sendet der Server eine 4xx Antwort. Beispielsweise, wenn ein angefordertes Dokument nicht vorhanden ist, diese eine Authentifizierung benötigt oder ein bestimmter Aufruf (GET, POST, PUT, DELETE, …) nicht zulässig ist.

5xx – Server-Fehler

Entsteht während der Bearbeitung ein serverseitiger Fehler, wird eine 5xx Antwort gesendet. Die Ursache kann beispielsweise sein, dass der Server gewisse Funktionen nicht besitzt (501).

4.7.1.4. JSON

Als Rückgabeformat der Daten wählten wir JSON, weil JSON eine Erweiterung von JavaScript ist und somit am besten mit der Apache-Cordova App, als auch mit der Website zusammenpasst. JSON wird hauptsächlich zur Speicherung und als Mittel der Übertragung von strukturierten Daten verwendet. Es ist ein kompaktes Datenformat, das einen sogenannten „Value“ einen „Key“ zuordnet. Es können alle mögliche Datentypen, Objekte, also auch Collections in Form von Arrays gespeichert werden (siehe 4.3.2. Newtonsoft). [20]

4.7.1.5. SQL

Structured Query Language ist eine Datenbanksprache, welche sowohl zur Beschreibung von Datenstrukturen, als auch zum Bearbeiten und Abfragen von Daten in der Datenbank verwendet wird. Fast alle Datenbanksystem unterstützen SQL, weil diese von ISO und ICE standardisiert wurde. SQL setzte sich aufgrund seiner leichten und verständlichen Syntax durch.

Sprachelemente:

- DML – Datenmanipulation (Einfügen, Löschen, Ändern, Lesen)
- DDL – Beschreibung des Datenbankschemas, praktisch deren Aufbau
- DCL – Verwaltung der Rechte und Transaktionen

4.7.1.6. MSSQL

Als Datenbank verwenden wir einen, auf Azure angemieteten, Microsoft SQL-Server. Dieser verfügt über ein relationales Datenbankmanagementsystem.
Dies bietet viele Vorteile gegenüber konventionellen Servern, da von Microsoft beste Erreichbarkeit und niedrige Ausfallwahrscheinlichkeit (wird durch global verteilte Serverfarmen erreicht) garantiert wird. Dazu bietet der Microsoft SQL-Server eine gute Kompatibilität mit anderen Microsoft Produkten und eine sehr preiswerte Alternative im Vergleich zu den Anschaffungskosten eines eigenen Servers.

4.7.1.7. Blob Storage

Um effizient große Daten wie Textfiles oder Bilder zu persistieren, ist es keine gute Idee diese direkt in einer SQL Datenbank abzubilden. Microsoft SQL Server unterstützen nur CLOBs (Character Large Object), aber keine BLOBs (Binary Large Objects). Um dieses Problem zu lösen, wurde ein BLOB Storage gewählt, ebenfalls auf der Azure Cloud gehostet. Die Referenz-URLs der Bilder und PDFs werden in der Datenbank gespeichert.

4.7.2. Controller-Katalog

Der folgende Controllerkatalog gibt einen Überblick über alle verfügbaren Controller der WebAPI mit den dazugehörigen Input- und Output Parametern. Weiteres sind zu jedem Controller der Controller-Type und ein Beispiel-Aufruf angeführt.

Controller - Name	Type	Inputparam.	Outputparam. (l. positive, r. negative)		Aufruf
BasketController	GET	ObjectID	200 OK, Array inkl. aller sich im Warenkorb befindlichen Pakete	401 Not Authorized, 404 Not Found	Source/Basket?ObjectID=
BasketController	PUT	Basket, ObjectID	200 OK	400 Bad Request, 401 Not Authorized, 404 Not Found	Source/Basket?ObjectID= Requestbody: Basket
CustomerController	GET	ObjectID	200 OK, Customer	401 Not Authorized, 404 Not Found	Source/Customer?ObjectID=
CustomerController	PUT	Customer	200 OK	400 Bad Request, 401 Not Authorized, 404 Not Found	Source/Customer Requestbody: Customer
ExerciseController	GET	ObjectID	200 OK, Array inkl aller Pakete und dazugehörigen Übungen	401 Not Authorized, 404 Not Found	Source/Exercise?ObjectID=
GetTotalController	GET	ObjectID	200 OK, TotalAmount	401 Not Authorized, 500 Internal Server Error	Source/Login?ObjectID=
LoginController	GET	EmailAddress	200 OK, Customer	404 Not Found	Source/Login?EmailAddress=
LoginController	GET	ObjectID, LogInOrOut	200 OK, Message	404 Not Found	Source/Login?ObjectID=&LogInOrOut=
LogonController	POST	Customer	200 OK	409 Conflict, 501 Internal Server Error	Source/Logon Requestbody: Customer
PackageController	GET		200 OK, Array inkl. aller Pakete	404 Not Found	Source/Package
PaymentController	GET	ObjectID	200 OK	401 Not Authorized, 404 Not Found, 500 Internal Server Error	Source/Payment?ObjectID=
RecipeController	GET	ObjectID	200 OK, Array inkl. aller Rezepte	401 Not Authorized, 404 Not Found	Source/Recipe?ObjectID=
SearchTagController	GET	ObjectID	200 OK, Array inkl. aller Rezeptschlagworte	401 Not Authorized, 404 Not Found	Source/SearchTag?ObjectID=

Tabelle 7 - Controller-Katalog

Die „Source“ in den Beispielaufrufen ist dabei: ***http://herbodyapi.azurewebsites.net/api***

4.8. Implementierung

4.8.1. ERD

Die Anforderung an das Datenmodell war es, eine Onlineshop ähnliche Umgebung für die Bedürfnisse des Verkaufs von Paketen, welche sowohl Übungen als auch Rezepte beinhalten, zu schaffen. Diese Pakete werden von sogenannten Customers (zu Deutsch: Kunden) gekauft und je nach georderter Länge in der Datenbank persistiert. Es muss dem Kunden möglich sein sich einzuloggen und zu registrieren. Da das Datenmodell auch als Datengrundlage der App und der Website dient, müssen zusätzliche Informationen über Trainingszyklen und Wiederholungen der jeweiligen Übungen gespeichert werden. Dies führt zu einem sehr komplexen Model, welches man besser anhand der Abbildung auf der nächsten Seite erklären kann. Um die Ansicht zu erleichtern ist eine Beschreibung der Hauptentitäten mit Zahlen und Assoziativtabellen mit Buchstaben gegeben.

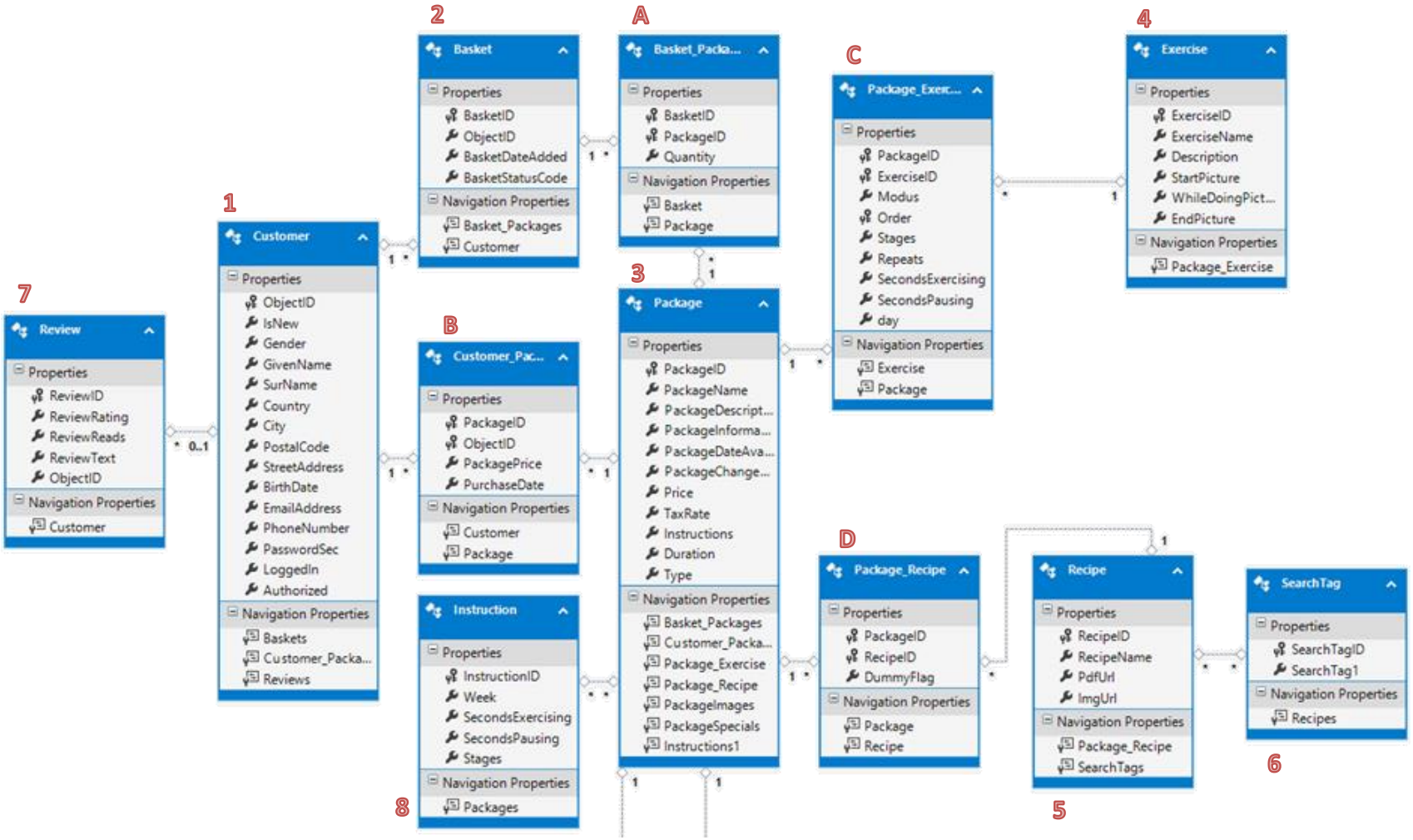


Abbildung 68 - ERD

1. In der Customer-Tabelle sind alle spezifischen Nutzerdaten – besonders ist dabei der verschlüsselte Passworthash.
2. In der Basket-Tabelle sind Status und Erstellungsdatum gespeichert, diese zwei Werte sind für den zukünftigen Ausbau des Projekts gedacht. Weiteres wird die ObjectID des Kunden gespeichert um den Warenkorb richtig zuordnen zu können – dies hat eigentlich zur Folge, dass ein Kunde mehrere Baskets haben kann, erleichtert aber die Abfrage. Die 1-1 Beziehung wird intern über Trigger geregelt.
3. Die Package-Tabelle umfasst Information und eine Beschreibung über das jeweilige Package wie zum Beispiel Preis, Laufzeit, oder Name.
4. Die Exercise-Tabelle umfasst eine Beschreibung der Übung, sowie die Referenzen der Trainingsbilder auf dem BLOB-Storage.
5. Die Recipe-Tabelle umfasst zwei Referenzen auf den BLOB-Storage
6. Die SearchTag-Tabelle erweitert die Rezepte um Schlagworte für eine effiziente Suche.
7. Die Review-Tabelle ermöglicht es von verifizierten Usern Meinungen, Kritiken und dergleichen zu speichern.
8. Die Instructions-Tabelle speichert für das Wedding-Package die Durchführung eines Zirkels.

A. Speichert die im Warenkorb befindlichen Pakete, sowie die Anzahl der Pakete.

B. Speichert die Pakete, die der User bereits gekauft hat.

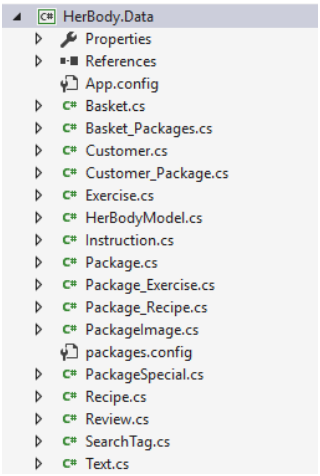
C. Speichert sowohl die Zugehörigkeit von Übungen zu Paketen, als auch die paketspezifische Übungsausführung.

D. Speichert die Zugehörigkeit von Rezepten zu Paketen, das Dummy Flag dient dazu, dass die Entitätsklasse erstellt wird. Dies ist hat den Grund, dass ansonsten die Abfrage und das Erstellen von Verbindungen komplexer gewesen wäre.

4.8.2. WebAPI

Wie bereits in der Funktionalität erwähnt, wird als Schnittstelle zwischen Backend und Frontend eine ASP.NET WebAPI verwendet. Diese wurde so implementiert, dass sie mit http-Aufrufen gesteuert werden kann.

Zuerst wurde ein leeres ASP.NET Projekt der gesamten VS15 Solution hinzugefügt. Mittels des Entity Frameworks und dem vorhandenen Wizard wurden die von der Datenbank bereits vorgegebenen Entitäts-Klassen (Code First From Database) generiert.



Erstellen der Entitäts-Klassen:

Rechtsklick auf Projekt

Add New Item → ADO.NET Entity Data Model

Code First From Database

Datenbankverbindung wählen/erzeugen

Datenbanktabellen auswählen und fertigstellen

Abbildung 69 - Entitätsklassen

Da wir in unserer Visual Studio Solution mehrere Projekte haben, haben wir das HerBody.Data - Projekt als Referenz zu der WebAPI hinzugefügt. Dies wird ebenfalls mittels Rechtsklick auf das Projekt – Add – Reference ermöglicht. Nun kann im WebAPI Projekt auf die Entitäts-Klassen zugegriffen werden. Zu jeder Entitäts-Klasse wurden passende Collections erstellt, um die Datenbankinhalte abfragen zu können. Diese Collections sind fest mit der Datenbank verbunden, das heißt, wird ein Objekt gelöscht, aktualisiert oder eingefügt, wird auch die Datenbank folglich verändert. Um Datenbankinhalte beziehungsweise die Inhalte der Collections abzufragen, wird LINQ verwendet. Je nach geforderten Inhalten wird der Controller mittels Rechtsklick auf das Projekt – Add – Controller – Web API 2 Controller-Empty erstellt.

Die Abfrage einer Collection, die nur einen eindeutigen Datenwert zurückliefern darf, wird mittels dem Primär-Schlüssel einer Entität durchgeführt. Die folgende Query liefert genau einen „Customer“ mit der ObjectID (Primärschlüsselattribut) zurück oder „null“ im Falle einer nichtexistierenden ObjectID.

```
Customer cust = (from c in db.Customers
                 where c.ObjectID.Equals(ObjectID)
                 select c).SingleOrDefault();
```

Abbildung 70 - Single Abfrage

Verschachtelte Abfragen von Tabellen, mit der Rückgabe mehrerer Werte, müssen mittels Schleifen durchlaufen werden um mehrere Objekte daraus abzufragen. In dieser Abfrage wird die Tabelle Package-Exercise (die assoziative Verbindung zwischen Paketen und Übungen) zuerst nach der „PackageID“ gefiltert und folglich nach den darin befindlichen „Exercises“ abgefragt. Die Abfrage liefert also die Übungen zu einem bestimmten Paket.
Durch LINQ ist es möglich solche Abfrage immens zu erleichtern, weil LINQ eine SQL ähnliche Struktur aufweist und diese mit Sprachelementen von C# verbindet. Wie im Codeausschnitt ersichtlich, wird einfach die Equals-Methode verwendet, um Elemente zu vergleichen.

```
var query3 = from pack_exec in db.Package_Exercise
                 where pack.PackageID.Equals(pack_exec.PackageID)
                 select pack_exec;
            foreach (var pack_exec in query3)
            {
                    var query4 = from exec in db.Exercises
                                 where
pack_exec.ExerciseID.Equals(exec.ExerciseID)
                                 select exec;
                                 foreach (var execise in query4)
                                 {

                                      …
```

Abbildung 71 - Verschachtelte Abfrage

Die bearbeiteten Ergebnisse werden mittels „HttpResponseMessage“ zu dem jeweiligen Client zurückgesendet. Hier wurden die http-Statuscodes und das Newtonsoft-Framework verwendet. Eine Antwort muss zwingend einen Statuscode besitzen und kann, wenn dies erforderlich ist, weitere Daten mitsenden. Diese Funktion wurde durch folgende Methode implementiert:

```
private HttpResponseMessage prepareResponse(HttpStatusCode statuscode, String
content)
        {
            if (content == null)
            {
                response = Request.CreateResponse(statuscode);
                response.Headers.CacheControl = new CacheControlHeaderValue()
                {
                    MaxAge = TimeSpan.FromMinutes(20)
                };
                return response;
            }
            else
            {
                response = Request.CreateResponse(statuscode);
                response.Content = new StringContent(content, Encoding.Unicode);
                response.Headers.CacheControl = new CacheControlHeaderValue()
                {
                    MaxAge = TimeSpan.FromMinutes(20)
                };
                return response;
            }
        }
```

Abbildung 72 - HttpResponseMessage

Diese Statuscodes werden von der WebAPI verwendet:

- 200 OK – Wird bei jeder erfolgreichen Anfrage an die Clients zurückgesendet.
- 201 Created – Wird beim erfolgreichen Registrieren eines Users gesendet.
- 400 Bad Request – Wird als Standardfehler zurückgesandt, wenn der API Aufruf nicht möglich war.
- 401 Unauthorized – Wird zurückgesendet, wenn eine Anfrage über eine ID eines Users gemacht wird, der nicht eingeloggt ist.
- 404 Not Found – Wird zurückgesandt, wenn die Angefragte Ressource nicht vorhanden ist in der Datenbank.
- 500 Internal Server Error – Wird zurückgesendet, wenn bei der Abfrage ein interner API Fehler auftritt.

Im Content eines http-Response verbirgt sich die eigentliche Nachricht, also der Payload der Nachricht. Weil als Datenformat JSON gewählt wurde, bietet sich die Newtonsoft Bibliothek gut für das Serialiseren und Deserialisieren von Objekten in und von JSON an.

Im folgenden Codeausschnitt wird die Verwendung der bereits erwähnten Methode „prepareResponse“, sowie die Verwendung von Newtonsoft („JsonConvert“) ersichtlich. Hierbei wird ein komplettes Customer-Object serialisiert und der Methode als Parameter übergeben.

```
public HttpResponseMessage GetCustomer(int ObjectID)
        {
            using (var db = new HerBodyModel())
            {
                if (db.isAuthorized(ObjectID))
                {
                    Customer cust = (from c in db.Customers
                                     where c.ObjectID.Equals(ObjectID)
                                     select c).SingleOrDefault();
                    if (cust == null)
                    {
                        return prepareResponse(HttpStatusCode.NotFound, null);
                    }
                    return prepareResponse(HttpStatusCode.OK,
JsonConvert.SerializeObject(cust));
                }
                return prepareResponse(HttpStatusCode.Unauthorized, null);
            }
        }
```

Abbildung 73 - Serialisierung eines Customer-Objects

In der nächsten Methode wird geprüft, ob der User mit der angeforderten ObjectID auch wirklich autorisiert ist. Bei der Registrierung wird sowohl das „loggedIn-Flag“, als auch das „authorized-Flag“ auf einen positiven Wert gesetzt. Fordert man nun Daten über die ObjectID an, prüft die API, ob der User, der die besagte ObjectID besitzt auch wirklich dazu berichtigt ist Daten anzufordern.

```
public Boolean isAuthorized(int ObjectID)
        {
            var query = from cust in Customers
                        where cust.ObjectID.Equals(ObjectID)
                        select cust;
            foreach (var cust in query)
            {
                if (cust.Authorized.Equals("y"))
                {
                    return true;
                }
                else
                {
                    return false;
                }
            }
            return false;
        }
```

Abbildung 74 - isAutorized()

Standard Abfragen wie GET-Requests, also das Anfordern von Daten unterscheiden sich nicht großartig voneinander. Sie sind immer wieder durch das gleiche Muster gekennzeichnet, werden jedoch komplexer, wenn diese mit einer oder sogar mehreren Beziehungen verknüpft sind. Um das Handeln dieser Assoziationen zu erleichtern bzw. gewisse Berechnung überhaupt möglich zu machen, wurden eigene Klassen geschrieben, die das angeforderte Objekt exakt so beschreiben, wie es für die Clients von Nöten ist.

```
public class PackageDBO
{
    public int PackageID { get; set; }
    public string PackageName { get; set; }
    public string PackageDescription { get; set; }
    public string PackageInformation { get; set; }
    public DateTime? PackageDateAvailable { get; set; }
    public DateTime? PackageChangeDate { get; set; }
    public decimal Price { get; set; }
    public decimal TaxRate { get; set; }
    public string Instructions { get; set; }
    public int Duration { get; set; }
    public List<ExerciseDBO> Exercises { get; set; }
    public List<Recipe> Recipes { get; set; }
    public string Type { get; set; }

}
```

Abbildung 75 - Klassen

4.8.3. WebApp

4.8.3.1. Allgemeines

Die Webseite wurde als ASP.NET Web Application angelegt und als AngularJS Single Page Web Application implementiert.

Eine AngularJS Single Page Web Application besteht im Prinzip nur aus einer Index-Seite. Der eigentliche Inhalt wird in Form von html-Seiten je nach Benutzerinteraktion dynamisch in den Content-Bereich der Index-Seite geladen.

4.8.3.2. Aufbau

Der Aufbau der HerBody Index-Seite beginnt mit der Initialisierung der AngularJS App.

Im <html> Tag wird dafür mit der ng-app Angular Notation der App Bereich der AngularJS App festgelegt.

```
<!DOCTYPE html>
<html ng-app="app">
```

Abbildung 76 - Initialisierung der Angular App

In der Headerdefinition der Seite befinden sich Meta-Informationen, die benötigten JavaScript-Dateien und die benötigten CSS-Dateien.

Am wichtigsten sind die Dateien des AngularJS-Frameworks und die Dateien des Bootstrap-Frameworks. Die Dateien des AngularJS-Frameworks sind zuständig für die Funktionalität der WebApp. Die Dateien des Bootstrap-Frameworks sorgen für die responsive Darstellung der WebApp.

```
<script src="Scripts/angular.min.js"></script>
<link href="Content/bootstrap.min.css" rel="stylesheet" />
```

Abbildung 77 - Framework-Dateien

Weitere wichtige Dateien die im Head-Bereich eingebunden werden:

- app.js
- controllers.js
- herbody.css (Stylesheet für die Webseite)
- jQuery—1.9.1.min.js (Wird von Bootstrap benötigt)
- bCrypt.js (Für die Passwortverhashung)
- ngStorage.min.js (Für den SessionStorage)

Gleich nach der Bodydefinition befindet sich eine responsive Navigationsleiste. Das bedeutet, dass die Navigationsleiste auf größeren Bildschirmen desktop-optimiert dargestellt wird und auf kleineren Bildschirmen mobile-optimiert dargestellt wird. Diese Navigationsleiste fungiert als Webseitenheader.

Darunter befindet sich ein Tag mit der AngularJS Notation ng-view. Diese Notation kennzeichnet den Bereich in den der Webseiteninhalt dynamisch geladen wird.

```
<div autoscroll="true" class="container" ng-view></div>
```

Abbildung 78 - Dynamischer Webseiteninhalt

Unter dem Content-Bereich befindet sich noch ein Webseiten-Footer, welcher Verlinkungen zu den Seiten Impressum, AGB, Datenschutz, Liefer- und Zahlungsbedingungen und Kontakt enthält.

4.8.3.3. app.js

Die app.js Datei wird in der Index-Seite als erste AngularJS-Datei eingebunden. In der app.js Datei wird als erstes eine app-Variable angelegt und mit „angular.module“ die nötigen Dateien eingebunden.

```
var app = angular.module('app', ['app.controllers', 'ngRoute', 'ngStorage']);
```

Abbildung 79 - Initialisierung der app Variable

Darunter befindet sich die „app.run“-Methode. Diese Methode wird beim App-Start aufgerufen. In dieser Methode befindet sich eine JavaScript-Funktion. In der Funktion werden Variablen bekannt gemacht, um diese im Code verwenden zu können.
In der run Methode befindet sich ein Codeteil, um den Seiten-Titel dynamisch zu ändern, damit der Benutzer das Gefühl bekommt, zwischen verschiedenen Seiten zu navigieren.

```
$rootScope.$on('$routeChangeSuccess', function () {
    document.title = $route.current.title + " - HerBody";
});
```

Abbildung 80 - Dynamischer Seitentitel

Darunter befindet sich die „app.config“-Methode. Dort werden die App-Routen bestimmt. Mithilfe des „routeProviders“, der ganz oben in der app.js-Datei mit „ngRoute“ eingebunden wird, können für verschiedene Routen verschiedene Dinge konfiguriert werden. Mit „$routeProvider.when“ kann man für eine bestimmte Route (im Beispiel „/private“) zum Beispiel den Titel, eine TemplateUrl und einen Controller definieren.

```
.when('/private', {
    title: 'Mitgliederbereich',
    templateUrl: 'pages/private.html',
    controller: 'PrivateController'
})
```

Abbildung 81 - Routen der WebApp

Außerdem wurde in der „app.config“-Methode das „HTTP-Response caching“ deaktiviert. Das sorgt dafür, dass jeder HTTP-Request neu durchgeführt wird und nicht auf gecachte Daten zurückgegriffen wird.

```
if (!$httpProvider.defaults.headers.get) {
    $httpProvider.defaults.headers.get = {};
}
$httpProvider.defaults.headers.get['If-Modified-Since'] = 'Mon, 26 Jul 1997 05:00:00 GMT';
$httpProvider.defaults.headers.get['Cache-Control'] = 'no-cache';
$httpProvider.defaults.headers.get['Pragma'] = 'no-cache';
```

Abbildung 82 - Deaktivieren des HTTP-Cachings

4.8.3.4. controllers.js

In der controllers.js Datei befinden sich für die Routen, denen ein Controller zugewiesen wurde, der zugehörige Controller. Wenn in der jeweiligen Route ein Controller definiert wurde, können alle Funktionen und Variablen des Controllers in der jeweiligen Seite der Route verwendet werden.

In der controllers.js wird als erstes das „angular.module“ definiert.

```
angular.module('app.controllers', [])
```

Abbildung 83 - Definition des AngularJS Moduls

Dieses Angular-Modul wird in der app.js ganz oben eingebunden.

```
var app = angular.module('app', ['app.controllers', 'ngRoute', 'ngStorage']);
```

Abbildung 84 - Einbindung des AngularJS Moduls

Darunter werden alle Controller implementiert, die in den App Routen definiert werden.
Ein Controller ist wie folgt aufgebaut:

```
.controller('Controller', function ($scope, $rootScope) {
    // Some Controller-Code

})
```

Abbildung 85 - Controller Aufbau

4.8.3.5. Konsumieren von Daten

Zum Konsumieren von Daten setzt die WebApp HTTP-Requests an die WebAPI ab. Um den „http-Scope“ verwenden zu können, muss er dem Controller überreicht werden.

Um zu verdeutlichen, wie die Daten konsumiert werden, folgt ein Code-Ausschnitt.

```
$scope.httpConfig = {
        headers: {
            'Content-Type': 'application/json'
        },
        params: parameter
};


$http.get($rootScope.apiString + '/controller, $scope.httpConfig)
.success(function (data, status, headers, config) {
        // success
})
.error(function (data, status, header, config) {
       // error
});
```

Abbildung 86 - Konsumieren von Daten mittels HTTP-Requests

Als erstes werden der Header und die Parameter des HTTP-Requests definiert. Diese werden beim Aufruf in der Config mitübergeben. Der „apiString“ ist die URL unserer gehosteten WebAPI. Wenn der Aufruf mit Statuscode 200 beantwortet wird, können im „.success“-Bereich die geforderten Daten aus „data“ ausgelesen werden und weiterverarbeitet werden. Auf einen Fehler kann im „.error“-Bereich reagiert werden. Zum Beispiel mit einem Bootstrap-Alert.

4.8.3.6. Seite aktualisieren

Damit der Benutzer die Webseite problemlos mit „F5“ oder mit dem „Seite neu laden“-Icon aktualisieren kann, ist das AngularJS-Modul „ngStorage“ eingebunden. Mit diesem Modul kann man den AngularJS SessionStorage nutzen und somit alle Benutzerdaten in den SessionStorage speichern. Wenn der Benutzer die Seite neu lädt und die WebApp neu initialisiert wird, werden die gespeicherten Daten vom SessionStorage wieder in den RootScope geladen. Somit kann der Benutzer die Seite neu laden, ohne dass sich der Benutzer immer neu anmelden muss.

```
$rootScope.localUser = $sessionStorage.localUser;
$rootScope.loggedin = $sessionStorage.loggedin;
```

Abbildung 87 - Verwendung des SessionStorage

4.8.3.7. PayPal Instant Payment Notification (IPN)

Bei „PayPal Instant Payment Notification“ handelt es sich um ein Tool von PayPal, welches ermöglicht Informationen einer PayPal-Transaktion zu erhalten und diese später zu verarbeiten.

Dabei sind vom Entwickler beim PayPal Geschäftskonto einige Einstellungen vorzunehmen: Die Einstellungen warden in der Einstellungsseite „My Selling Tools“ (Profile – More Options – My Selling Tools) vorgenommen. Unter dem Reiter “Getting paid and managing my risk” wird die Einstellungsseite für Instant Payment Notification gefunden, welche unter „Update“ aufgerufen werden kann.

Auf der Update-Seite für Instant Payment Notification ist die Möglichkeit gegeben, die Einstellungen für Instant Payment Notification zu ändern, beziehungsweise die Instant Payment Notification zu deaktivieren. Unter „Edit Settings“ wird ein Link zu einem, am Webserver gehosteten, Script hinterlegt, an welches PayPal die Transaktionsdaten nach einer Transaktion weiterleitet.

Abbildung 88 - PayPal IPN Einstellungen

Wenn ein Kunde auf der HerBody-Webseite über den PayPal-Button seine Bestellung bezahlt, sendet PayPal alle relevanten Daten an das listener.php Script, welches auf dem Webserver gehostet ist.

In diesem Script wird genauestens überprüft, ob die Transaktion erfolgreich gewesen ist und ob die Transaktion verifiziert ist.

Die von PayPal erhaltenen Daten werden an PayPal zurückgesendet, wenn das Ergebnis darauf „VERIFIED“ ist, sind die Transaktionsdaten verifiziert. Wenn die Transaktion verifiziert ist, können die Transaktionsdaten ausgelesen und vom listener.php – Script weiterverarbeitet werden.

Der PayPal-Button, welcher sich auf der Warenkorb-Seite der Webseite befindet, enthält ein custom-input Feld, in welchem die ObjectID des Users gespeichert ist, welchem der Warenkorb gehört.

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post" target="_top">
    <input type="hidden" name="cmd" value="_xclick">
    <input type="hidden" name="business" value="payments@her-body.at">
    <input type="hidden" name="lc" value="DE">
    <input type="hidden" name="item_name" value="HerBody Bestellung">
    <input type="hidden" name="amount" value="{{total}}">
    <input type="hidden" name="currency_code" value="EUR">
    <input type="hidden" name="custom" value="{{localUser.ObjectID}}">
    <input type="submit" class="btn btn-success" ng-disabled="checkEmpty()"
value="Bezahlen" />
</form>
```

Abbildung 89 - PayPal Button

Dieser custom Input kann im listener.php Script ausgelesen und weiterverarbeitet werden. Die ObjectID wird mit einem HTTP-Post-Request zur Web API gesendet, welche die nötigen Datenbankaktionen erledigt, die nach einem Paketverkauf nötig sind.
Der Kunde erhält nach der PayPal-Bezahlung eine Bestätigungsnachricht, welche direkt im php-Script abgesendet wird.

```
$email = 'payments@her-body.at';
$header = 'From: '.$email."\r\n".'Reply-To: '.$email."\r\n";
mail( $payer_email, 'Zahlungsbestätigung: HerBody Bestellung', $msg, $header );
```

Abbildung 90 - Bestätigungsmail mittel PHP mail()-Funktion

4.8.3.8. Newsletter

Der Newsletter wurde beim Anbieter Joletter.de aufgesetzt und auf der Webseite eingebunden. Die Newsletter-Anmeldung besteht aus einem HTML-Formular, welches die Daten an Joletter abschickt.

```
<form action='http://www.joletter.de/add_newsletter.php' method='post' >
```

Abbildung 91 - Newsletter-Formular

4.8.4. App

Zu Beginn wurde die Apache Cordova App als Projekt mit dem Typ „Blank Cordova App“ zur Visual Studio Solution hinzugefügt. Zum Aufbau der Cordova App lässt sich sagen, dass es sich wie bei der Webseite um eine AngularJS Single Page Application handelt.

Dabei wird beim Start der App vorerst die Index-Seite (HTML5 Datei) geladen und je nach Benutzerinteraktion sogenannte Templates die in den „Content-Bereich“ der Index-Seite geladen werden. Das bedeutet die einzelnen Templates, die den speziellen Seiteninhalt definieren, werden dynamisch in die Index-Seite geladen und angezeigt.

Der Aufbau der Index-Seite beginnt mit der Headerdefinition wo alle benötigten JavaScript-Dateien und CSS-Styles geladen werden und die Meta-Daten verankert sind. Allen voran natürlich die benötigten Dateien für das Ionic-Framework, wo einerseits die benötigte Funktionalität über die „ionic.bundle.js“ Datei zu Verfügung gestellt wird und andererseits die Ionic-Styles über die „ionic.app.css“ Datei bereitgestellt werden.

```
<script src="lib/ionic/js/ionic.bundle.js"></script>
<link href="css/ionic.app.css" rel="stylesheet">
```

Abbildung 92 - Framework-Dateien

Nach den Headerdefinitionen folgt der Body, der die „ng-app=“app““ Information enthält und somit den folgenden Inhalt als App definiert. Im Body kommt, dann schon zum ersten Mal das Ionic-Framework zum Einsatz. Es wird die seitliche Navigationsbar, die bis auf der Loginseite, überall verwendbar und ersichtlich ist, über das Tag <ion-side-bar> definiert. Weiteres wird durch das <ion-content> Tag jener Bereich erzeugt, wo anschließend die Templates hineingeladen werden.

Ein solches Template ist folgendermaßen aufgebaut:

```
<ion-view title="exampleTemplate" id="page01" style="background-color:#FFB2BB;">
  <ion-content padding="true" class="has-header">

  </ion-content>
</ion-view>
```

Abbildung 93 - Ionic Template

Im <ion-view> Tag werden Titel, Id und die Hintergrundfarbe definiert. Weiteres inkludiert es ein weiteres Tag, nämlich das <ion-content> Tag, dass schlussendlich den eigentlichen Inhalt wie zum Beispiel Listen, Buttons oder Inputfelder beinhaltet. Zusätzlich werden im <ion-content> Tag die Attribute padding und class gesetzt um den Header, der den Titel beinhaltet, zu berücksichtigen.

Weitere wichtige JavaScript Dateien die im Header der Index-Datei geladen werden sind:

- cordova.js
- app.js
- controllers.js
- routes.js

Die cordova.js Datei legt sozusagen den Grundbaustein der Cordova App und inkludiert die gesamte Funktionalität einer Apache Cordova App. Um die Funktionalität später noch weiter ausbauen zu können, können sogenannte Plugins über den Plugin-Manager in der config.xml Datei installiert werden.

In der app.js Datei wird die App genau angelegt und spezifiziert. Dabei wird zu Beginn eine Variable für die App angelegt, die die Module enthält die geladen werden. Darüber hinaus kann über die Option „config“ die App genau konfiguriert werden. In unserem Fall wird in der Konfiguration ein sehr wichtiger Punkt implementiert. Nämlich die Deaktivierung des Caching von http-Responses.

```
if (!$httpProvider.defaults.headers.get) {
    $httpProvider.defaults.headers.get = {};
}
$httpProvider.defaults.headers.get['If-Modified-Since'] = 'Mon, 26 Jul 1997 05:00:00
GMT';
$httpProvider.defaults.headers.get['Cache-Control'] = 'no-cache';
$httpProvider.defaults.headers.get['Pragma'] = 'no-cache';
```

Abbildung 94 - Deaktivierung des HTTP-Caching

Die Deaktivierung ist deshalb so wichtig, weil sonst die von der WebAPI kommenden http-Response Messages gecached werden würden und die neuen Requests nicht mehr abgesetzt werden würden. Das Problem wird durch folgendes Beispiel klar:
Der Benutzer startet die App und loggt sich erfolgreich mit seinen korrekten Benutzerdaten ein. Dabei übermittelt die WebAPI einen positiven 200OK http-Response. Dieser wird anschließend gecached. Loggt sich der User später erneut ein, aber mit falschen Benutzerdaten würde der Request zum Einloggen gar nicht abgesetzt werden da die positive Response-Message vom letzten Login noch gecached ist und der User hätte sich „erfolgreich“ mit falschen Benutzerdaten eingeloggt.

Weiteres enthält die app.js Datei enthält die „run“ Funktion die die erste Funktion ist, die beim App-Start ausgeführt wird. Diese enthält die Implementierung der „ready“ Funktion, welche ausgeführt wird, sobald das Ionic-Framework erfolgreich geladen wurde.
In dieser „ready“ Funktion werden die globalen Variablen wie zum Beispiel die API-URL, der lokale User oder der User-Warenkorb angelegt und initialisiert.

In der controllers.js Datei wurde für jedes Template, sprich grafische Komponente ein Controller angelegt der die Hintergrundfunktionalität bereitstellt. Im Template kann der benötigte Controller zum Beispiel im <ion-content> Tag über das setzten des „ng-controller=“controller““ Attributes zugewiesen werden.
Innerhalb dieses Tags stehen anschließend alle Funktionen und Variablen, die im Umfang des Controllers inkludiert sind, zur Verfügung.
Wichtige Controller sind:

- LoginCtrl
- UserCtrl
- LiveWorkoutCtrl

Dabei ist der LoginCtrl mit allen nötigen Funktionen, die mit der Registrierung und dem Login in Verbindung gebracht werden, ausgestattet. Auch die Ver- und Entschlüsselung des Benutzerpassworts mittels der bcrypt-Funktion übernimmt der LoginCtrl. Weiteres wurde der Controller abstrahiert und so implementiert, dass er sowohl von der App, als auch von der Webseite verwendet werden kann.

4.8.4.1. bcrypt

Die kryptologische Hashfunktion bcrypt wurde mit dem Ziel Passwörter zu hashen und zu speichern entwickelt. Die Funktion baut auf dem symmetrischen Blockverschlüsselungsalgorithmus Blowfish auf. Ist das Passwort einmal gehasht worden, kann nicht mehr auf den Klartext zurückgefolgert werden. Zur Authentifizierung des Benutzers wird das eingegebene Passwort mit der gleichen Funktion gehasht und anschließen mit dem gespeicherten Passwort Hash verglichen. Im Gegensatz zu den Hashfunktionen SHA1 und MD5 ist bcrypt nicht entwickelt worden, um Daten schnell und effizient zu hashen, sondern die Sicherheit der Daten zu gewähren. Das macht sich natürlich in der Geschwindigkeit bemerkbar, bietet aber eine weitaus höhere Sicherheit gegen Brute-Force-Attacken. [21]

Der UserCtrl hingegen bietet alle Funktionen die mit dem User zusammenhängen, wie zum Beispiel die Änderung bestimmte Benutzerdaten oder die Änderung des Benutzerpassworts. Und der LiveWorkoutCtrl ist der zuständige Controller für das absolvieren des Live-Trainings über die App.

Das Anlegen eines Controllers wurde wie folgt implementiert:

```
.controller('exampleCtrl', ['$scope', '$stateParams',
function ($scope, $stateParams) {


}])
```

Abbildung 95 - Controller Aufbau

Im Beispiel wurde ein Controller mit dem Namen ‚exampleCtrl' und den Funktionsparametern ‚$scope', ‚$stateParams' und der Controller-Funktion angelegt. Die übergebenen Parameter können anschließend im gesamten Controller verwendet werden.
Alle weiteren Funktionen werden üblicherweise innerhalb jener Funktion implementiert, die als Parameter übergeben wurde und auf den aktuellen ‚$scope' gesetzt, was bedeutet, dass die Funktion nur innerhalb dieses Controllers verfügbar ist.

Soll die Funktion jedoch global aufrufbar gemacht werden, muss als Parameter der ‚$rootScope' Parameter übergeben werden und die Funktion im ‚$rootScope' angelegt werden.
Auch wird der ‚$rootScope' Parameter benötigt, wenn innerhalb des Controllers auf globale Variablen zugegriffen werden soll.

In der routes.js Datei werden sämtliche Pfade zwischen den einzelnen Templates definiert um anschließend zwischen den Templates navigieren zu können. Eine Route wird dabei folgendermaßen angelegt:

```
.state('exampleRoute', {
      url: '/page01',
      templateUrl: 'templates/exampleTemplate.html',
      controller: 'exampleCtrl'
  })
```

Abbildung 96 - Routen der App

Im Beispiel wird der Route der Name „exampleRoute" verliehen. Weiteres werden die Parameter url, templateUrl und controller gesetzt. Die url beschreibt die gesetzte Id des Template. Die templateUrl hingegen beinhaltet den Pfad inklusive Dateinamen und Dateiendung zur Template-Datei. Und der controller-Parameter weist der Route bzw. dem Template einen Controller zu.

Wurden die benötigten Routen implementiert kann man in den Controllern über den ‚$state' Parameter zwischen den einzelnen Templates navigieren.

```
$scope.changeState = function (pagename) {
      $state.go(pagename);
}
```

Abbildung 97 - Navigation zwischen den Templates

Wie bereits erwähnt wurde, setzt die App zum Konsumieren der Daten http-Requests an die WebAPI ab. Um diese Requests in den einzelnen Controllern zu ermöglichen muss der ‚$http' Parameter dem Controller überreicht werden. Anschließend kann im Controller der Header des Requests konfiguriert werden und dem Request mitgegeben werden.

```
$scope.httpConfig = {
        headers: {
            'Content-Type': 'application/json'
        },
        params: parameter
};
$http.get($rootScope.apiString + '/controller, $scope.httpConfig)
     .success(function (data, status, headers, config) {
             //success: do something
     })
     .error(function (data, status, header, config) {
            //error: do something
     });
```

Abbildung 98 - Konsumieren von Daten mittels HTTP-Requests

Im Code-Ausschnitt wird eine solche http-Konfiguration mit Header und Parameter angelegt und anschließend beim Absetzen des Requests an den WebAPI-Controller mit dem Namen „controller“ übergeben. Von der WebAPI wird als Rückgabewert entweder ein positiver oder negativer Response-Code übermittelt. Abhängig von diesem Rückgabewert wird der Code entweder im .success Zweig oder im .error Zweig weiter abgehandelt.

Für die Gestaltung der Templates wird in der Index-Datei das Stylesheet customfonts.css eingebunden um die von den Auftraggebern gewünschten Schriftarten und Farben verwenden zu können. In diesem Stylesheet werden diese benutzerdefinierten Schriftarten aus den jeweiligen .ttf Dateien geladen, angelegt und zur Verfügung gestellt.

```
@font-face {
  font-family: "AutographScriptLight";
   src: url('../lib/ionic/fonts/Autograph Script/ufonts.com_autographscriptef-
        light.ttf') format("truetype");
   font-weight: normal;
   font-style: normal;
}
```

Abbildung 99 - Verwendung von benutzerdefinierten Schriftarten

4.8.5. Admininterface

Um die Datenbankinhalte möglichst einfach warten zu können wurde eine ASP.NET MVC Anwendung implementiert. Diese wurde ebenfalls, wie auch die anderen Projekte, zu unserer HerBody Solution hinzugefügt und dementsprechend referenziert, um auf die Entitätsklassen zugreifen zu können. Die GUI kann leicht mittels eines Wizards erstellt werden – Rechtsklick – Add – New Scaffolded Items. Jedoch werden nicht nur die graphischen Elemente erzeugt, sondern auch die Controller, welche die GUI mit voller Funktionalität (CRUD – Operationen) erweitert.

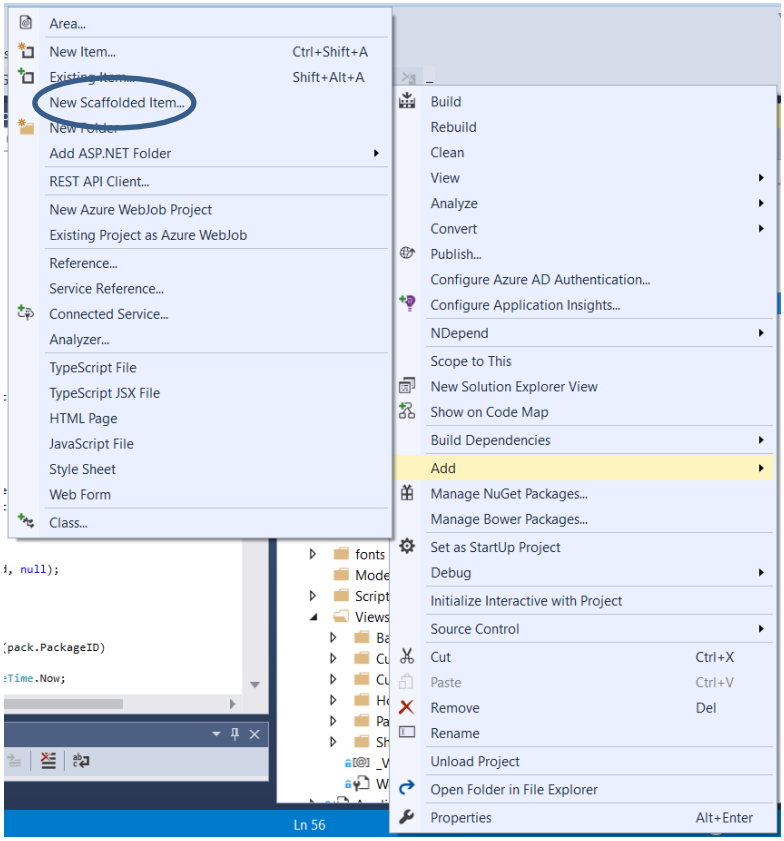


Abbildung 100 - Erstellung des GUIs

Dieses Muster zieht sich durch alle Seiten durch.

Was sich als deutlicher Vorteil bemerkbar macht ist, dass im Datenmodell keine reinen assoziativen Tabellen sind. Dies würde die Implementierung des Admininterfaces auf diese Weise erschweren. Denn dann müssten die Views für die assoziativen Tabellen eigenhändig ausprogrammiert werden. Andersrum muss nur zuerst die Primärtabelle mit Inhalten befüllt werden und anschließend die Sekundärtabelle. Dieses Problem wird auch „Henne-Ei Problem“ genannt.

4.9. Online-Marketing Maßnahmen

4.9.1. Suchmaschinenoptimierung

Die Suchmaschinenoptimierung beschäftigt sich mit Techniken, die einer Webseite dazu verhelfen, bei Suchmaschinen einen besseren Rang zu erhalten. Da Google mit über 90% Marktanteil Marktführer der Suchmaschinen ist, beschäftigen sich die meisten Suchmaschinen-Optimierer hauptsächlich mit Google und dessen Page Rank Algorithmus. Die Optimierung bei Verkaufswebseiten ist besonders wichtig, denn je weiter oben eine Webseite bei Google gelistet ist, desto mehr Menschen besuchen die Webseite und umso mehr Menschen kaufen sich dann Produkte auf der Seite.

Um eine Webseite für Google attraktiver zu machen, kann man verschiedene Dinge der Webseite optimieren. Einerseits können On-Page Optimierungen durchgeführt werden, die sich mit inhaltlichen Verbesserungen direkt auf der Webseite beschäftigen. Anderseits können Off-Page Optimierungen durchgeführt werden, welche aber nicht direkt auf der Webseite vorgenommen werden.

4.9.1.1. On-Page Optimierungen

Die On-Page Optimierung befasst sich mit allen inhaltlichen Anpassungen auf der Webseite. Für Google ist es besonders wichtig, dass die Webseite nicht nur eine reine Verkaufsseite ist, sondern eine Seite, die den Internetnutzern einen inhaltlichen Mehrwert bietet. Bei einer Fitnessseite ist es daher wichtig, dem Webseitenbesucher kostenlosen Fitness-Content zu bieten. Dieser Content kann zusätzlich suchmaschinenoptimiert gestaltet sein, in dem bestimmte Keywords (Suchbegriffe) geschickt im Text platziert sind. Wenn diese Keywords dann gegoogelt werden, wird die Seite mit dem suchmaschinenoptimierten Content bei Google weiter oben angezeigt.

4.9.1.2. Off-Page Optimierungen

Die Off-Page Optimierung befasst sich hauptsächlich mit dem Aufbau von qualitativ hochwertigen Links. Ein Link ist dann qualitativ hochwertig, wenn er von einer Webseite kommt, die Google mit einem hohen Page Rank eingestuft hat. Diese Links werden oft auch Backlinks genannt. Je mehr qualitativ hochwertige Backlinks auf die eigene Webseite führen, desto besser wird sie von Google bewertet. Am besten sind Backlinks von Webseiten, die sich mit dem selben Thema befassen wie die eigene Webseite. Backlinks von sozialen Netzwerken werden von Google besonders berücksichtigt, da eine Webseite, die oft in sozialen Netzwerken geteilt wird, viel relevanter und somit wichtiger ist, als Webseiten ohne Backlinks sozialer Netzwerke.

4.9.2. Lead-Generierung

Als Lead bezeichnet man im Online-Marketing, einen potentiellen Kunden beziehungsweise einen Kontakt, der mit einer Marketing-Maßnahme gewonnen worden ist. Diese Leads werden meist mit einem Newsletter auf der Verkaufswebseite eingesammelt. Webseitenbesucher, die sich kein Produkt kaufen, aber trotzdem Interesse haben, haben die Möglichkeit ihre E-Mail-Adresse und ihren Namen eintragen. Der Webseitenbetreiber findet dann in seinem Newsletter-Tool eine Liste an Personen vor, die sich genau für eine gewisse Thematik interessieren. Diese potentiellen Kunden können dann mit gezieltem E-Mail-Marketing in zahlende Kunden verwandelt werden.

4.9.3. Umsetzung

Die Webseite enthält viele zusätzliche Informationen zum Thema Fitness und Ernährung, um den Webseitenbesuchern kostenlosen, inhaltlichen Mehrwert zu bieten.
Außerdem werden nach Projektabschluss Links in unserer Facebook-Gruppe platziert, um einige Backlinks zu generieren.

Um Leads zu generieren wurde ein Newsletter-Tool aufgesetzt. Die gesammelten Leads können von unseren Auftraggebern dann für E-Mail-Marketing-Maßnahmen verwendet werden.

5. Qualitätssicherung

Um ein qualitatives Produkt zu schaffen, haben wir einen besonders großen Wert auf Qualitätssicherung gelegt. Um die Qualität des Produkts zu gewährleisten sind regelmäßig Produkttests durchgeführt worden.

5.1. Qualitätsmerkmale

Die Qualitätsmerkmale nach ISO 25010 werden im folgenden Teil beschrieben.

5.1.1. Zuverlässigkeit

Unser System bestehend aus Webseite, App und Backend und Admin-Interface läuft zuverlässig ohne auftretende Probleme.

5.1.2. Benutzbarkeit

Die Webseite und die App wurden kundenfreundlich programmiert, sodass das System sehr einfach verwendet werden kann. Das System kann problemlos bedient werden.

5.1.3. Effizienz

Durch das Speichern der Übungen und der Rezepte auf einem Azure Blob-Storage stehen die Daten schnell zur Verfügung.

Da JavaScript Dateien der Webseite direkt am Webserver gespeichert sind, kann die Webseite schnell aufgerufen werden.

5.1.4. Wartbarkeit

Unser System kann vollständig mit dem HerBody Admin-Interface gewartet werden.

5.1.5. Sicherheit

Um geschützte Daten im Mitgliederbereich ansehen zu können, muss der User angemeldet sein. Das Passwort des Users wird bei der Übertragung mit bCrypt verschlüsselt.

5.1.6. Kompatibilität

Unser System ist mit vielen Geräten kompatibel. Die Webseite kann sowohl am PC, am Laptop, Tablet und am Smartphone geöffnet werden, da sich die Webseite an das Client-Gerät anpasst.

5.1.7. Portabilität

Die HerBody App wurde mit dem Apache Cordova Framework entwickelt. Dieses Framework ermöglicht es dem Entwickler, die App einmal zu entwickeln und diese dann für alle Plattformen zu veröffentlichen, zum Beispiel für Android oder iOS. [22]

5.2. Maßnahmen

5.2.1. Ausführliche Planung

Jeder Arbeitsschritt wurde sowohl projektorganisatorisch als auch technisch geplant.

5.2.2. Regelmäßige Auftraggeber-Meetings

Um den Auftraggebern aktuellen Projektfortschritte präsentieren zu können, haben wir regelmäßig Projektmeetings in den Räumlichkeiten der HTL-Perg abgehalten. Dort wurden erledigte und ausstehende Aufgaben besprochen und Vereinbarungen getroffen.

5.2.3. Regelmäßige Projekt-Meetings

Um aufgetretene Fehler schnell und effektiv beheben zu können, wurden regelmäßig projektinterne Meetings abgehalten, um unser System zu testen und Programmierbugs schnell aus der Welt zu schaffen.

5.2.4. Ausführliches Testen

Unser System wurde ausführlich getestet, um die Qualität des Gesamtsystems zu gewährleisten (siehe 5.3. Testen).

5.3. Testen

Um die Qualität des Gesamtsystems zu sichern wurde das System natürlich ausreichend getestet und die Programmabläufe im Debugg-Modus kontrolliert.

Um das Testen zu erleichtern wurde die WebAPI nicht jedes Mal auf MS Azure veröffentlicht, sondern als Startup – Projekt in Visual Studio definiert und auf dem Localhost auf Port 31529 laufen gelassen. In der WebAPI konnten anschließend gezielt in jenen Controllern die getestet werden sollten Breakpoints gesetzt werden.

Zusätzlich zur WebAPI konnten sowohl die WebApp, als auch die mobile App als neue Instanz, parallel zur WebAPI gestartet werden. Natürlich wurde in den beiden Client-Applikationen die jeweilige API-URL im Vorhinein auf den Localhost geändert. Weiteres laufen die beiden Client-Applikationen auf einem anderen Port.

Schlussendlich konnten in den beiden Frontends jene http-Request abgesetzt werden, in denen vorher in der WebAPI die Breakpoints gesetzt wurden. Dies ermöglichte ein sehr ausführliches Testen.
Darüber hinaus konnten die JavaScript- Dateien in den Client-Applikationen mit dem „debugger;“ Keyword versehen werden um ein Debugging in der Browser-Konsole zu ermöglichen.

5.4. Probleme & Lösungen

5.4.1. Technische Probleme

Während der Entwicklung der App ergab sich ein Problem bei der Implementierung des Logins mit dem Azure Active Directory, kurz AAD. Das AAD ist ein von Microsoft zur Verfügung gestelltes Verzeichnis, dass in der Microsoft Azure Cloud angelegt und konfiguriert werden kann. Es übernimmt dem Programmierer die Implementierung des Login Systems. Sowohl die grafische Oberfläche, als auch die Funktionalität im Hintergrund kann online konfiguriert werden. Für die spätere Verwendung des AADs muss ein Token mit gewissen Konfigurationsparametern generiert werden, der in unserem Fall auf Grund eines nicht identifizierbaren Fehlers nicht verwendbar war. Da wir vergebens sehr viel Zeit in die Fehlerbehebung investierten, entschieden wir uns das Login System mit dazugehöriger Datensicherheit selbst zu implementieren.

Ein weiterer Fehler trat beim Testen der API auf, wo mittels einer Test-App die ersten HTTP-Requests abgesetzt wurden, aber jedes Mal als Responsemessage der Response Error 404 Not Found angenommen wurde. Grund dafür waren falsch gesetzte Headerattribute beim Bauen des http-Requests, welche mittels des Visual Studio Debuggers identifiziert werden konnten und anschließend richtig gesetzt werden konnten.

Ein weiteres technisches Problem war, dass ein Browser-Client zum Konsumieren von Daten von der WebAPI erst berechtigt werden muss. Dieses Problem wurde mit Hilfe von CORS gelöst, welches serverseitig in der WebAPI implementiert wurde.

5.4.2. Organisatorische Probleme

Es war nicht immer einfach einen Überblick über Fehler, die noch behoben werden mussten und Arbeitsaufgaben zu behalten. Gelöst werden konnte dieses Problem nur mit klassischen TODO-Listen und Fehlerlisten auf die jeder Diplomand Einsicht hatte.

Ein sehr heikles Problem aus organisatorischer Hinsicht war die Kommunikation mit den Auftraggebern, die sich nur nach sehr langer Wartezeit wieder meldeten, auf Fragen reagierten und uns die angeforderten Daten wie zum Beispiel Bilder zukommen haben lassen. Leider waren diese Daten oft nicht korrekt, worauf wieder nach den korrekten Daten angefragt werden musste. Lösung für dieses Problem war das erneute Kontaktieren der Auftraggeber bis eine Rückmeldung mit den richtigen Daten kam.

Wenn im Team oder außerhalb organisatorische Probleme aufgetreten sind, konnten wir uns immer an Frau Prof. Mag. Danner wenden, welche uns dann beraten hat und uns immer eine Lösung für unsere organisatorischen Probleme bieten konnte.

5.4.3. Fehlermanagement

Aufgetretene Fehler wurden in einem Excel-Sheet mitgeschrieben, um diese bis zur Behebung für alle sichtbar zu machen. Wenn Fehler nicht direkt behoben werden konnten, wurden diese im Team diskutiert und gemeinsam beseitigt.

5.5. Risikomanagement

Projektrisiko	Eintrittswahr-scheinlichkeit	Risikoauswirkung	Maßnahmen	Verantwortlichkeit
Fehlende Ressourcen	15%	terminlich	Fehlende Ressourcen werden beschafft	Auftraggeber (finanziell) & Projektteam (Umsetzung)
Schnittstellenprobleme	25%	technisch, terminlich	Gemeinsames Lösen der Probleme im Projektteam	Projektteam
Mangelnde technische Erfahrung	20%	technisch, terminlich	Aneignen neuer Technologien, Aufsuchen eines Fachlehrers	Projektteam
Risiken aus Zusammenarbeit mit dem Auftraggeber	10%	Terminlich, finanziell	Absprachen mit Auftraggebern & Festlegung fixer Termine	Auftraggeber

Tabelle 8 - Risiken

6. Release Verwaltung

6.1. Microsoft Azure

Die jeweils aktuellste Version der WebAPI und der WebApp können bequem über die Projektlösung in VS15 geöffnet werden und auf die im Vorhinein angelegte Ressource auf der Azure Cloud veröffentlicht werden.

Anhand dieses Beispiels kann die Veröffentlichung verdeutlicht werden:
Als Beispiel wird die Veröffentlichung der WebAPI auf die Azure WebApp Ressource herangezogen. Bevor mit der Veröffentlichung begonnen werden kann, muss sich der Entwickler mit jenem Microsoft Account in Visual Studio anmelden, auf dem der Azure Account läuft.

Wurde dieser Schritt erledigt kann mittels eines Rechtsklicks auf das Projekt der WebAPI im Kontextmenü der Punkt „Publish“ ausgewählt werden.

Anschließend kann ein Veröffentlichungsprofil angelegt werden wo der Programmierer diese Inputs tätigen muss:

- Abonnement des Azure Account
- Ressourcengruppe
- Ressource auf die die API veröffentlicht werden soll

Das angelegte Profil wird anschließend gespeichert und kann bei erneuter Veröffentlichung geladen werden und den gesamten Vorgang zu erleichtern.

Hat der Entwickler alle Eingaben getätigt, hat er die Möglichkeit eine Vorschau zu starten. Ist diese Vorschau erfolgreich muss abschließend nur mehr der „Publish-Button“ gedrückt werden um die Veröffentlichung zu starten.

Die folgende Abbildung zeigt jenes Fenster, mit dem die WebAPI, nach erfolgreichem Anlegen eines Profils, veröffentlicht werden kann.

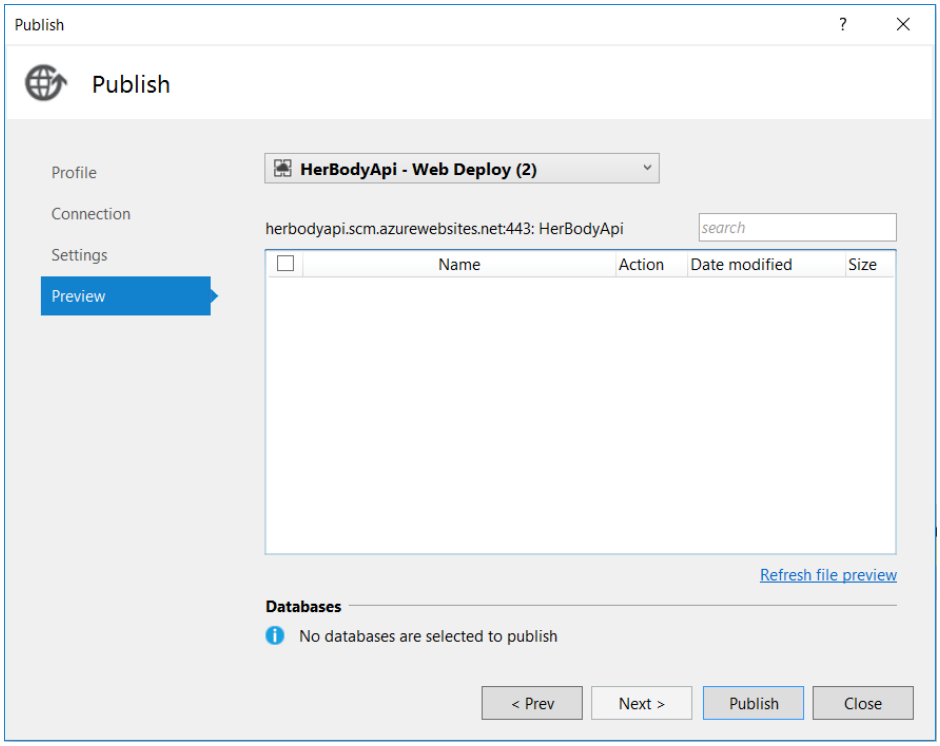


Abbildung 101 - Publish-Window

In der Visual Studio Konsole kann der Vorgang mit Ergebnis verfolgt werden:

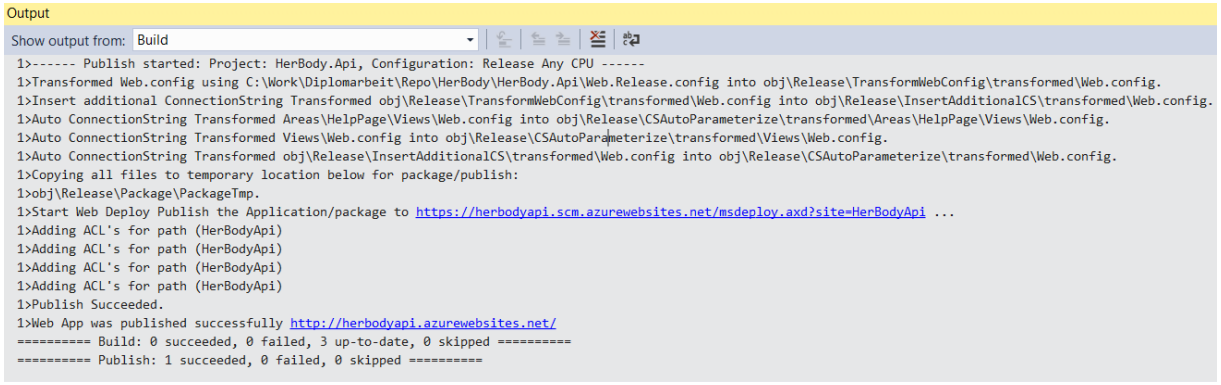


Abbildung 102 - Publish-Ergebnis

6.2. Stores

Die mobile plattformübergreifende App wurde nach fertiger Implementierung und Testung im Windows Store und im Google Play Store veröffentlicht.
Da sowohl für die Veröffentlichung einer App im Play Store, als auch im Windows Store ein Developer-Account benötigt wurde, wurde dieser mit der Google Mailadresse des Projektteams angelegt. Die einmaligen Aktivierungskosten übernahmen die Auftraggeber.

6.2.1. Google Play Store

Die Veröffentlichung einer App im Google Play Store erfolgt über die „Google Developer Console“ wo vorerst die App mit einem Namen angelegt werden muss. Dann wird die App mit Attributen wie Preis, Kategorie, Beschreibung, Icon, Screenshots und noch vielen weiteren genau spezifiziert. Schlussendlich kann eine. apk Datei, sprich eine für Android Geräte verpackte Anwendung, hochgeladen werden.

Diese. apk Datei kann in Visual Studio über die Build-Release Funktion für die Zielplattform Android erzeugt werden.

6.2.2. Windows Store

Im Windows Store wird genaugenommen nicht für den Account im Windows Dev Center, wo die App veröffentlicht werden kann, bezahlt, sondern für das Reservieren des App-Namen. Dieser muss natürlich eindeutig sein und darf nicht bereits existieren. Hat man den App-Namen reserviert und die App angelegt, wird die mobile Anwendung ebenso wie im Google Playstore genauestens spezifiziert.

In Visual Studio kann für eine Veröffentlichung im Windows Store ein sogenanntes App-Package erzeugt werden. Dies wird ebenfalls durch die Build-Release Funktion ermöglicht, allerdings mit der Zielquelle Windows.

Hochgeladen wird nach Generierung des App-Packages jene Datei im Release Ordner mit Dem Dateityp „.appxupload“.

Im Anschluss folgt eine Zertifizierung von Microsoft die mehrere Tage in Anspruch nehmen kann. Dabei wird zum Beispiel überprüft ob die hochgeladenen Screenshots der App mit der wirklichen App übereinstimmen. Wird die Anwendung von Microsoft freigegeben ist sie schlussendlich im Windows Store beziehbar.

6.3. Veröffentlichung der Webseite

Für das Unternehmen HerBody wurden folgende Domains reserviert:

- her-body.at
- her-body.ch
- her-body.com
- her-body.de
- her-body.it
- her-body.ru
- her-body.uk
- her-body.us

Das Projekt wird jedoch nur auf der Domain her-body.at veröffentlicht. Für die anderen erworbenen Domains ist eine Weiterleitung eingerichtet worden, die auf die Domain her-body.at weiterleitet.

6.3.1. Webhosting

Für das Unternehmen HerBody wurde beim Webhosting Anbieter „All-Inkl“ das Webhosting Paket „All-Inkl Privatplus“ gebucht. Dieses Paket enthält genügend Speicherplatz für die Webseite und unterstützt auch alle anderen benötigten Dinge wie zum Beispiel eine Webmail-Funktion.

6.3.2. Veröffentlichung

Die Webseite ist mit dem FTP-Tool „FileZilla“ auf dem Webspace veröffentlicht worden und kann unter folgender URL besucht werden: http://www.her-body.at.

Eine Beschreibung zur Veröffentlichung der WebAPI und der mobilen App kann unter dem Punkt Release Verwaltung eingesehen werden (siehe 6. Verwaltung).

7. Evaluierung

7.1. Planung vs. Realisierung

Da jedes Mitglied des Projektteams schon zu Beginn des Projektes gewisse Qualitäten auf Grund von verschiedensten absolvierten Schul- und Privatprojekten mitbrachte und diese auch im zugeteilten Modul einsetzen konnte, stellte das Einarbeiten in die Technologien nicht wirklich ein Problem dar. Darüber hinaus standen uns für fachliche Fragen die Fachlehrer der HTL – Perg und die Firma Softaware zur Verfügung. Trotz diesem durchaus positiven Projektstart wurde das Projekt nicht auf die leichte Schulter genommen, da das Gesamtsystem nach der Planung für unsere Verhältnisse doch sehr komplex ausfiel.

Zu Beginn des Projekts tauchten Fragen zum AAD auf, darum wurde ein Treffen mit einem Spezialisten in der Softwareentwicklung vereinbart, der uns nach Analyse der Gesamtsituation des Projekts auch dazu riet die Systemarchitektur zu überdenken. Das Projektteam folgte diesem Rat und überarbeitete die Architektur und setzte anstatt eines WCF-Data-Services eine WebAPI ein.

Da schlussendlich, trotz allem die Probleme mit dem AAD nicht behoben werden konnten und sehr viel Zeit in die Fehlersuche investiert wurde, entschied sich das Entwicklerteam für die Implementierung eines eigenen Login-Systems mit dazugehörigem Sicherheitsaspekt, was im Nachhinein gesehen sicherlich schon weit früher hätte beschlossen werden sollen.

Eine weitere Änderung die von der eigentlichen Planung abweicht, ist die WebApp, die in der Planung eigentlich als normale HTML5 Webseite geplant war, jedoch auf Grund von Funktionalitätsmangel als .NET WebApp und AngularJS als Single Page Application implementiert wurde. Daraus ergibt sich der Vorteil der gemeinsamen Verwendung gewisser JavaScript Controller mit der mobilen App, da diese ebenfalls unter Verwendung des AngularJS Frameworks implementiert wurde.

Zu Projektabschluss kann man trotz vielen Höhen und Tiefen während der Projektdurchführung sagen, dass sowohl die Auftraggeber vom Gesamtsystem profitiert haben, als auch das Entwicklerteam, dass sehr viel Erfahrung sammeln durfte.

7.2. Stundenverteilung

Die folgende Grafik beschreibt die Aufteilung der Arbeitsstunden pro Person, die sowohl für die Implementierung der Diplomarbeit, als auch für die Verschriftlichung aufgewendet worden sind.

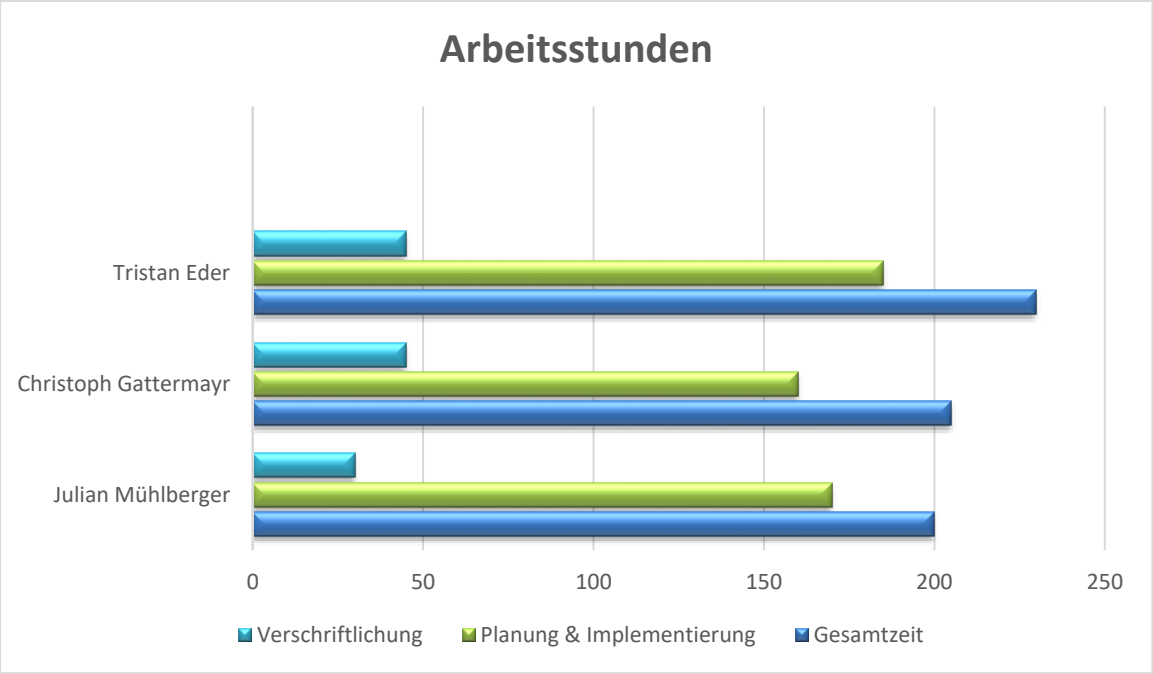


Abbildung 103 - Stundenverteilung

Aus dieser Grafik ergibt sich eine Gesamtanzahl von 635 erreichten Arbeitsstunden.

Da wir bei der Implementierung und bei der Verschriftlichung doch mehr Zeit verwendet haben, sind wir am Projektende 95 Arbeitsstunden jenseits der Aufwandsschätzung.

7.3. Persönliche Resümees

7.3.1. Resümee - Tristan Eder

Da mich das Thema „Automatisierter Vertrieb von digitalen Produkten im Internet“ schon länger interessiert hat, und plötzlich die Möglichkeit vorgelegen ist ein solches System in Form eines Schulprojekts selbst zu programmieren, habe ich sofort gewusst, dass ich diese Aufgabenstellung annehmen möchte und dazu meine Diplomarbeit verfassen möchte.

Um dieses Aufgabe zu meistern, musste ich mir einige neue Technologien aneignen und sehr viel meiner Zeit investieren. Wenn ich jetzt auf die Zeit der Projektentwicklung zurückblicke, muss ich sagen, dass es genau die stressigen Momente waren, die mich als Person und uns als Team voranbrachten. Dieses Projekt ermöglichte es mir in einer kurzen Zeit sehr viele Dinge zu lernen und viele Erfahrungen zu sammeln. Die Erfahrungen, die ich im projektorganisatorischen Bereich sammeln konnte und auch die neu erlernten technischen Fähigkeiten, werden mir sicher in Zukunft noch weiterhelfen können.

Nach Fertigstellung dieses Projekts kann ich behaupten mit meinem Projektteam ein tolles Produkt geschaffen zu haben, auf das wir stolz sein können.

7.3.2. Resümee - Christoph Gattermayr

Schon seit Beginn der fünfjährigen Ausbildung ist die Mobile Computing Thematik jenes Themengebiet, dass die größte Begeisterung rund um die Informatik in mir hervorrief und jenes, dass das wahre „Programmier-Feuer“ in mir entfachte. Darum war es für mich ein sehr wichtiger Aspekt auch in der Diplomarbeit eine mobile Anwendung zu realisieren. Durch das HerBody Projekt wurde mir dieses Vorhaben ermöglicht. Nicht nur das Entwickeln einer neuartigen, plattformübergreifenden mobilen Anwendung mit einer neuen Technologie, sondern auch die Fitness Thematik rund um das Projekt, das Projektumfeld mit den Auftraggebern, dem Betreuungslehrer und den Entwicklerkollegen bot einen extrem hohen Lernfaktor durch den es mir möglich war in kurzer Zeit sehr viel Erfahrung zu sammeln und mich dementsprechend weiterzuentwickeln. Ich bin davon überzeugt, dass sich diese Erfahrung im späteren Leben nur positiv auswirkt und mir bei etwaigen Fragen weiterhilft. Darüber hinaus war es für mich eine erneute Bestätigung, dass ich in der Berufswelt des Softwareentwicklers und des Projektenwicklers richtig bin, mich wohlfühle und diesen Zweig im späteren Berufsleben mit Sicherheit weiterführen werde. Und wer weiß, vielleicht treibt mich dieser Fluss an Bestätigung sogar einmal zur erfolgreichen Selbstständigkeit.

7.3.3. Resümee - Julian Mühlberger

Nicht nur, dass Fitness einen sehr wichtigen Punkt in meinem Leben darstellt machte dieses Projekt sehr reizvoll, sondern auch die Herausforderung ein Produkt zu entwerfen, welches die Standards der heutigen Zeit erfüllt, um gewinnbringend auf dem Markt durchzustarten. Darüber hinaus ist mir der Teil des Backends zugesprochen worden, was auch meine Erwartungen des Programmierens völlig erfüllt. Grund dafür ist, dass für mich nicht das Design beziehungsweise generell das Designen, sondern viel mehr die Funktionalität in einem Programm im Vordergrund steht. Somit war die Entwicklung, gemeinsam mit meinem Projektteam und dem Projektumfeld, als auch unserer Betreuungslehrerin, sowie auch unseren Auftraggebern, für mich sehr komfortabel und bot zudem noch einen riesigen Mehrwert. Einerseits im Thema Organisation und Kommunikation, als auch in technischer Hinsicht bot mir dieses Projekt viel Lernpotential, welches ich meiner Meinung nach gut ausschöpfen konnte. Ich freue mich darauf, dieses Projekt, nach Schulabschluss weiter betreuen zu dürfen.

8. Anhang

8.1. Glossar

Admininterface	Ein Administratorinterface erlaubt es dem Administrator einen Überblick über das System zu haben und es zu verwalten.
Backend	Jene Funktionalität die der User nicht zu Gesicht bekommt und meist von einem Server aus fungiert.
Bitbucket / SmartGit / GitHub	Bitbucket ist der verwendete Online-Anbieter für die Versionsverwaltung. SmartGit ist das zugehörige Client-Programm und GitHub ist eine Plattform zur Bereitstellung von Open-Source-Projekten.
Brute-Force-Attacke	Ein Angriff, bei dem alle Möglichkeiten so lange ausprobiert werden, bis die richtige Lösung für ein Problem gefunden wird (Angriff der rohen Gewalt).
Bug	Ein Bug ist ein Fehler im Programm.
Build / Release	Eine Build-Software ist eine sich noch in Entwicklung befindliche Software, wohingegen eine Release-Software die finale, fehlerfreie Software bezeichnet.
Client	Der Client ist nichts anderes als der Kunde oder der Benutzer.
Controller	Ein Controller stellt eine gewisse Funktionalität zur Verfügung die jederzeit aufgerufen werden kann. Controller werden sowohl in den beiden Frontends, als auch im Backend eingesetzt.
Conversion	Mit Conversion wird die Umwandlung von einem Webseitenbesucher zu einem zahlenden Kunden gemeint.
CSS	Eine Stylesheet-Sprache, um HTML-Dokumente zu formatieren.
Debugging	Ist ein schrittweiser Prozess zum Beheben von Codefehlern und zur zeitlichen Analyse der Variablen.

Domain	Eine Domain ist der Name einer Webseite, der weltweit eindeutig ist. Die Domain ist ein wesentlicher Bestandteil der URL.
Footer	Der Footer ist der Bereich der Fußzeile. Dort befinden sich meist Verlinkungen zum Impressum und zu den AGBs.
Framework	Eine Bibliothek an Funktionen die zusätzlich eingebunden und verwendet werden kann.
Frontend	Jene Funktionalität, die der User sieht und mit der er interagieren kann. Die Umsetzung erfolgt meist durch verschiedene Clients.
GUI	Ist die grafische Benutzeroberfläche mit der der User interagieren kann.
Hashfunktion	Eine Funktion zur Verschlüsselung eines Klartextes. Grundsätzlich kann von einem Hash nicht mehr auf den Klartext zurückgeschlossen werden.
HTML	Eine Textbasierte Auszeichnungssprachen die eine Struktur zur digitalen Darstellung von Inhalten bietet.
http-Caching	Unter http-Caching versteht man das temporäre Speichern von http-Responses.
JavaScript	Eine Skriptsprache, die es Entwicklern ermöglicht, Webseiten dynamisch zu gestalten.
Kick-Off-Meeting	Meeting zum Projektstart mit dem Auftraggeber und allen Projektbeteiligten. Bei diesem Meeting werden die Projektziele und die Schritte zur Erreichung besprochen. Meistens dient dieses Meeting als Feier und als Symbol für den Projektstart.
Localhost	Beschreibt das lokale, momentan genutzte System.
Mobile Computing	Der Bereich Mobile Computing umfasst das das gesamte Spektrum in der Informatik welches sich mit der Entwicklung mobiler Anwendungen (Apps) beschäftigt.

Nativ	Eine Native Entwicklung erfordert die genaue Anwendung einer Programmiersprache für eine spezifische Technologie (Bsp. Java → Android). Das Gegenteil wäre die Cross-Platform-Programmierung.
Popup, Modal-Popup	Ein visuelles Anzeigefenster, welches nach einer bestimmten Benutzeraktion aufspringt und über dem eigentlichen Webseiteninhalt dargestellt wird.
Razor Seite	Eine Razor Seite ist eine ASP.NET Seite, die das Razor Markup, also das Einbetten von Server – Code (C# oder VB) verwendet.
Release	Die Veröffentlichung eines Programmes.
Scrum	Scrum ist ein agiles Vorgehensmodell im Projektmanagement und wird meist in der Softwareentwicklung eingesetzt.
Server	Ein Computer der gewissen Funktionalitäten für Clients bereitstellt.
Tag	Tags werden häufig in Websprachen verwendet und grenzen bestimmte Bereiche mittels „<“ und „>“ ab.
WCF-Data-Service	Ein WCF-Data Service ermöglicht es einem, Daten über das Open Data Protocol (ODATA) zu konsumieren. Dieser Service ist Bestandteilt des .NET Frameworks.

Tabelle 9 - Glossar

8.2. Quellenverzeichnis

[1]	Training http://sport1.uibk.ac.at/lehre/lehrbeauftragte/Huber%20Reinhard/Dehnen_%DCbungen.pdf
[2]	Kalorien Beispiel https://www.medizinpopulaer.at/archiv/seele-sein/details/article/gesund-abnehmen-was-und-wie-viel-darf-ich-essen.html
[3]	Ernährung http://www.netdoktor.at/laborwerte/makronaehrstoffe-6684722
[4]	Ernährung http://www.fitforfun.de/abnehmen/gesunde_ernaehrung/gesunde-ernaehrung-die-rolle-der-naehrstoffe-eiweiss-fett-und-kohlenhydrate_aid_10133.html
[5]	Ernährung https://www.vitamine.com/mineralstoffe/ratgeber/tagesbedarf/
[6]	Projektorganisation Projektentwicklungsbuch (Schulbuch) http://www.wissenistmanz.at/produkte/htl-mtl/systemplanung-und-projektentwicklung/catalog_package_view?packageUID=8700c3366eadab4efab609e4891ad875&b_start=0
[7]	ASP.NET https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx
[8]	Apache Cordova https://cordova.apache.org/docs/en/latest/
[9]	Apache Cordova http://www.dynamogold.at/dg/info/blog/detail/id/apache-cordova-einfuehrung-native-mobile-apps-mit-html5-css-und-javascript-2/
[10]	Apache Cordova https://de.wikipedia.org/wiki/PhoneGap
[11]	Microsoft Azure https://azure.microsoft.com/en-us/overview/what-is-azure/

[12]	AngularJS https://angularjs.de/artikel/was-ist-angularjs
[13]	Newtonsoft http://www.newtonsoft.com/json/help/html/Introduction.htm
[14]	CORS https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/enabling-cross-origin-requests-in-web-api
[15]	Bootstrap https://www.prestashop.com/blog/de/was-ist-bootstrap-die-geschichte-und-der-hype-teil-1-von-2/
[16]	Bootstrap http://holdirbootstrap.de/ueber-bootstrap/
[17]	Ionic http://t3n.de/magazin/entwicklung-ionic-237246/
[18]	Visual Studio https://www.visualstudio.com/en-us/productinfo/vs2015-sysrequirements-vs
[19]	Microsoft SQL Management Studio https://msdn.microsoft.com/de-de/library/ms143506(v=sql.120).aspx#hwswr
[20]	JSON https://en.wikipedia.org/wiki/JSON
[21]	bCrypt https://en.wikipedia.org/wiki/Bcrypt
[22]	Qualitätsmerkmale http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Management-der-Systementwicklung/Software-Qualitatsmanagement/Qualitatsmerkmale-von-Software/index.html

Tabelle 10 - Quellen

8.3. Abkürzungsverzeichnis

.APK	Android Application Package
AAD	Azure Active Directory
AC	Antagonist-Contract
API	Application Programming Interface
ASP. NET	Active Server Pages .NET
CORS	Cross-Origin-Resource-Sharing
CR	Contract-Relax
CSS	Cascading Style Sheets
DCL	Data Control Language
DDL	Data Definition Language
DIE	Integrated Development Environment
DML	Data Manipulation Language
EF(6)	EntityFramework(Version 6)
ERD	Entity Relationship Diagramm
GB	Gigabyte
GHz	Gigahertz
GnbR	Gesellschaft nach bürgerlichem Recht
GUI	Graphical User Interface
HTBLA	Höhere technische Bundeslehranstalt
HTL	Höhere technische Lehranstalt
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identifikationsnummer
IEC	International Electrotechnical Commission
ISO	Internationale Organisation für Normung
JS	JavaScript
JSON	JavaScript Object Notation
MS	Microsoft Azure
OS	Operating System
POJO	Plain Old Java Object

REST	Representational State Transfer
U/min	Umdrehungen pro Minute
URL	Uniform Resource Locator
VS(15)	VisualStudio (2015)
WCF	Windows Communication Foundation
XML	eXtensible Markup Language

Tabelle 11 - Abkürzungen

8.4. Abbildungsverzeichnis

8.5. Tabellenverzeichnis

8.6. Protokollvorlage

Sitzungsprotokoll – HerBody	
Datum:	Uhrzeit: __:__
Ort:	Raum:
Sitzungsleiter:	
Teilnehmer:	
Protokollführer:	Vertraulichkeit: öffentlich / privat
Thema:	
Ergebnisse (Besprechungsinhalte und Vereinbarungen):	
Kundenversprechen:	
Anhang:	
Ort, Datum, Unterschrift	