

HTBLA Perg
Machlandstraße 48
4320 Perg

Tel.: +43 7262 53926



Diplomarbeit

EatFit - Ernährungsplaner im Web



Projektteam: Erkol Beytullah
Voß Oliver

Projektbetreuung: Mag. Danner Gabriela

Bearbeitungszeitraum: 16.06.2015 -
08.04.2016

Eidesstattliche Erklärung

Perg, am _____

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.

Erkol Beytullah

Voß Oliver

Zusammenfassung

„Man ist, was man isst.“ Um gesundheitliche und sportliche Ziele, wie Fettabbau, Muskelaufbau oder körperliches Wohlbefinden zu erreichen, ist deshalb eine bilanzierte Ernährung fundamental.

Die Diplomarbeit befasst sich mit der Implementierung einer Webanwendung für jene, die sich bilanziert ernähren wollen. "EatFit" ermöglicht dem Anwender plattformunabhängig die Planung von Mahlzeiten und unterstützt ihn darüber hinaus, durch das automatische Erstellen einer Einkaufsliste mit den nötigen Lebensmitteln, bei der Umsetzung seines Vorhabens. Die Einkaufsliste gibt es darüber hinaus als Android-App für unterwegs.

Alles beginnt damit, dass sich der Anwender ein persönliches Ziel setzt und die entsprechende Kalorienmenge, sowie Nährstoffverteilung auf Fette, Kohlenhydrate und Proteine kennt, die dafür nötig ist. Diese Werte werden am besten im Rahmen eines Gespräches mit einem Ernährungsberater eruiert.

Nun kommt "EatFit" ins Spiel. In der Webanwendung findet der Anwender viele gewöhnliche Standardlebensmittel, mit allen elementaren Nährstoffangaben, kann diese Datenbank aber auch noch selbst durch eigene Lebensmitteleinträge erweitern.

Mahlzeiten können aus beliebigen vielen Lebensmitteln und jeweiliger Angabe der Menge zusammengestellt werden. Bei der Mahlzeitplanung sieht der Benutzer neben seinen selbst definierten Zielwerten, auch immer die Summe der Nährstoffe und Kalorien der bisher geplanten Mahlzeiten des Tages. Ebenso wird die Differenz dieser Tages- und Zielwerte angezeigt, wodurch der Benutzer auf das Gramm exakt planen kann.

Für jede Woche wird automatisch eine Einkaufsliste mit allen geplanten Lebensmitteln erstellt. Nach dem Einkaufen kann der Anwender den Einkauf in Webanwendung oder App bestätigen. Auch Teileinkäufe sind möglich.

Abstract

"You are what you eat." To achieve health and athletic goals like fat reduction, muscle building or physical wellbeing, it's fundamental to have balanced nutrition

The thesis has to do with the implementation of a web application, for all who want to eat a balanced diet. "EatFit" enables the user to schedule meals platform independent by using any web browser and supports him furthermore by automatically creating a grocery list with the needed items to execute his plan. The grocery list is also available to go as an Android app.

It all starts with the user setting himself a personal goal and knowing the corresponding amount of calories as well as the matching nutrient distribution. It's best to evaluate these values as part of a conversation with a nutritionist.

Now "EatFit" comes in. There are many usual groceries with all basic nutrition information in the web application, but the user can extend this database with his own entries.

Meals can consist of any number of groceries and particular indication of quantity. When planning a meal, the user always sees his target values next to the sums of the nutrients and calories of the meals he has planned on that day. The difference of these values is shown as well, which allows the user to plan exactly.

For each week a grocery list is created automatically. When done shopping the user can confirm the purchase in the web application or the app. Even separating the shopping is possible.

Danksagung

Wir bedanken uns bei all jenen, die uns über den gesamten Projektverlauf hinweg in jeglicher Hinsicht unterstützt haben.

Ein besonderer Dank gilt unserer Betreuungslehrerin Mag. Gabriela Danner, die uns mit großem Engagement und Einsatz geholfen hat. Wir bedanken uns für ihre Unterstützung in Planung und Umsetzung des Projektes.

Darüber hinaus bedanken wir uns bei allen Lehrkräften, die uns bei Fragen immer hilfsbereit unterstützt haben und so am Erfolg dieser Diplomarbeit teilhaben.

Herzlichen Dank!

Inhaltsverzeichnis

Eidesstattliche Erklärung.....	3
Zusammenfassung.....	5
Abstract	7
Danksagung	9
Inhaltsverzeichnis.....	11
1 Stakeholder	15
1.1 Projektteam.....	15
1.2 Schule	15
1.3 Projektbetreuung	15
2 Entstehung und Hintergrund.....	17
2.1 Ausgangssituation & Projektidee	17
2.1.1 Konkurrenzprodukte & Preis	17
2.1.1.1 Fddb Premium Preise	17
2.1.1.2 MyFitnessPal Premium Preise	17
2.1.2 EatFit vs. Konkurrenzprodukte	17
2.1.3 Motivation für EatFit	19
2.2 Grundlagen Ernährung	21
2.2.1 Kalorienbilanz & Makronährstoffe	21
2.2.2 EatFit Nährstoffübersicht	22
2.2.3 Festlegen der Zielwerte	23
2.2.4 Beispiel Zielwerte & Mahlzeitplanung.....	23
3 Projektziele.....	25
3.1 Anwenderziele.....	25
3.1.1 Bilanzierte Ernährung.....	25
3.1.2 Gesundheit & Sport	25
3.2 Ökonomische Ziele	25
3.3 Ökologische Ziele.....	25
4 Grundlagen und Technologien	27
4.1 Technologien zur Erstellung	27
4.1.1 HTML	27
4.1.2 CSS	27
4.1.3 PHP	27
4.1.4 MySQL.....	28
4.1.5 MySQLi.....	28

4.1.6	JavaScript.....	28
4.1.7	jQuery.....	29
4.1.8	AJAX.....	29
4.1.9	Windows Server 2012.....	29
4.1.10	Android.....	29
4.2	Software zur Umsetzung.....	30
4.2.1	phpMyAdmin.....	30
4.2.2	Notepad ++.....	30
4.2.3	Eclipse Mars PHP Developer.....	30
4.2.4	Firebug.....	30
4.2.5	Google Drive.....	30
4.2.6	VMware vSphere Client.....	31
4.2.7	IIS.....	31
4.2.8	FTP.....	31
4.2.9	Android Studio.....	31
4.2.10	DB Designer Fork.....	31
4.2.11	FileZilla Client.....	32
5	Umsetzung.....	33
5.1	Webanwendung.....	33
5.1.1	GUI Planung.....	33
5.1.2	GUI Umsetzung.....	33
5.1.3	Verschlüsselung.....	34
5.1.4	Aufbau.....	35
5.1.5	connection.php.....	37
5.1.6	auth.php.....	37
5.1.7	Session Variablen.....	38
5.1.8	Datenaustausch.....	38
5.1.9	Verwendung JQuery.....	40
5.1.10	Datetimepicker.....	41
5.2	Android-Applikation.....	42
5.3	Datenbank.....	44
5.3.1	Standard Daten.....	44
5.3.2	Datenmodell.....	44
5.3.3	Beschreibung der Datenbanktabellen.....	46
5.4	Server.....	50
5.4.1	Webserver (IIS).....	50

5.4.2	File Transfer Protocol (FTP)	51
6	Ergebnis	53
6.1	Komponentenübersicht.....	53
6.2	Beispiel Mahlzeitplanung mit EatFit.....	54
6.3	Webanwendung	57
6.3.1	Pages & Funktionen.....	57
6.3.1.1	Benutzerauthentifizierung.....	57
6.3.1.2	Wochenübersicht	57
6.3.1.3	Tagesbilanz	58
6.3.1.4	Mahlzeit planen	58
6.3.1.5	Meine Lebensmittel – Vorrat	59
6.3.1.6	Meine Lebensmittel – Verwaltung	60
6.3.1.7	Meine Lebensmittel - Einkaufsliste	61
6.3.1.8	Profil	61
6.3.1.9	Zielsetzung.....	62
6.4	Android-Applikation	63
6.4.1	Login	63
6.4.2	Einkaufsliste.....	64
7	Evaluierung.....	65
7.1	Herausforderungen & Lösungen	65
7.1.1	Schulalltag & Schulabschluss.....	65
7.1.2	Einhaltung des Zeitplans.....	65
7.1.3	Lösung.....	65
7.2	Persönliches Resümee.....	66
7.2.1	Erkol Beytullah.....	66
7.2.2	Voß Oliver	67
8	Verzeichnisse	69
8.1	Abbildungsverzeichnis.....	69
8.2	Tabellenverzeichnis	70
8.3	Literaturverzeichnis.....	71
9	Anhang.....	73
9.1	Projektdokumentation	73
9.1.1	Projektstrukturplan	73
9.1.2	Meilensteine.....	75
9.1.3	Zeitplan.....	75
9.1.4	Arbeitsaufteilung.....	79
9.1.5	Kommunikation	79

9.1.5.1	Teamintern	79
9.1.5.2	Projektteam & Projektbetreuung	79
10	Leistungsumfang	80
10.1	Leistungsumfang hinsichtlich der Technologien	80
10.2	Leistungsumfang hinsichtlich der Applikationen	80
10.3	Leistungsumfang hinsichtlich der Geschäftsprozesse	81

1 Stakeholder

1.1 Projektteam



Abbildung 1: Projektteam

	Erkol Beytullah	Voß Oliver
Adresse	Bäckerstraße 29 4482 Ennsdorf	Struden 45a 4381 St. Nikola
Klasse	5AHIF	5AHIF
Geburtsdatum	17.06.1996	28.12.1995
E-Mail-Adresse	b.erkol@gmx.at	voss.oliver28@gmail.com

Tabelle 1: Personendaten Projektteam

1.2 Schule

Die Diplomarbeit wurde im Rahmen der Reife- und Diplomprüfung an der HTL Perg, Schwerpunkt Informatik, im Schuljahr 2015/16 durchgeführt.

1.3 Projektbetreuung

Betreut wurde die Diplomarbeit von Mag. Gabriela Danner. Sie fungierte als beratende Stelle und kontrollierte in regelmäßigen Abständen den Projektfortschritt.

2 Entstehung und Hintergrund

Voß Oliver

2.1 Ausgangssituation & Projektidee

Es gibt viele Softwareprodukte zum Thema Ernährungskontrolle, doch diese sind meist kostenpflichtig, unübersichtlich aufgebaut und es fehlen nützliche Funktionen.

2.1.1 Konkurrenzprodukte & Preis

EatFit wird in diesem Abschnitt mit zwei Konkurrenzprodukten verglichen. Beide Kontrahenten bieten eine kostenfreie Version und eine Premium-Mitgliedschaft. In den kostenfreien Versionen wird Werbung geschaltet und es stehen nur begrenzt Funktionen zur Verfügung. Zunächst betrachten wir die Preise der möglichen Premium-Optionen.

2.1.1.1 Fddb Premium Preise

- € 3,95 für 3 Monate € 1,32 pro Monat
- € 5,95 für 6 Monate € 1,00 pro Monat
- € 9,95 für 12 Monate € 0,83 pro Monat

2.1.1.2 MyFitnessPal Premium Preise

- € 9,99 monatlich
- € 49,99 jährlich

2.1.2 EatFit vs. Konkurrenzprodukte

In den folgenden fünf Tabellen wird EatFit mit diesen zwei bereits angebotenen Produkten hinsichtlich wichtiger Kriterien und Funktionen verglichen.

Kriterium	Fddb	MyFitnessPal	EatFit
Anzahl maximal möglicher Mahlzeiten pro Tag	6	6	unbegrenzt
Angabe einer Portionsgröße	uneingeschränkt	uneingeschränkt, aber umständlich	uneingeschränkt
Mahlzeitbezeichnung	vorgegeben	frei wählbar	frei wählbar

Tabelle 2: Kriterien EatFit vs. Konkurrenz

Vergleich: Angabe einer Portionsgröße von 150 g

MyFitnessPal	EatFit
<p>Spar - Mager Topfen (0,2%)</p> <p>Wie viel?</p> <p>150 Portion(en) 1 g</p> <p>Zu welcher Mahlzeit?</p> <p>Abende</p> <p>100 g</p> <p>1 g</p> <p>1 container (250 gs ea.)</p> <p>3.5 ounce</p> <p>1 ounce</p>	<p>Produkt</p> <p>Reis, geschält</p> <p>Menge</p> <p>150</p> <p>Einheit</p> <p>g</p> <p>Hinzufügen</p>

Abbildung 2: MyFitnessPal Angabe Portionsgröße

Abbildung 3: EatFit Angabe Portionsgröße

Angabe erfolgt verwirrend in 150
Portionen zu je 1 g

Angabe erfolgt einfach in 150 g

Tabelle 3: Vergleich: Angabe einer Portionsgröße

Vergleich: Anzahl der verfügbaren Nährstoffe

Nährstoff	Fddb	MyFitnessPal	EatFit
Vitamine	8	2	12
Mineralstoffe	13	2	15

Tabelle 4: Vergleich: Anzahl verfügbarer Nährstoffe

Vergleich: Zugriffsberechtigungen eigener Lebensmitteleinträge

Funktion	Fddb	MyFitnessPal	EatFit
einsehen	jeder	jeder	nur Autor
erstellen	jeder	jeder	jeder
ändern	jeder	jeder	nur Autor
löschen	jeder	jeder	nur Autor

Tabelle 5: Vergleich: Zugriff eigener Lebensmitteleinträge

Funktion	Fddb	MyFitnessPal	EatFit
automatisch erstellte Einkaufsliste	nein	nein	ja
automatisch erstellter Wochenplan der Mahlzeiten	nein	nein	ja

Tabelle 6: Funktionsvergleich EatFit vs. Konkurrenz

2.1.3 Motivation für EatFit

Die Motivation zur Implementierung von EatFit ist, ein Softwareprodukt zur Ernährungskontrolle zu erstellen, welches nicht die Schwachstellen der bisher auf dem Markt erhältlichen Produkte aufweist.

Darüber hinaus wird auf Folgendes besonderer Wert gelegt:

- Kostenfreiheit des Produkts
- intuitive Bedienung
- übersichtlicher Aufbau der grafischen Benutzeroberfläche
- praxisorientierte Funktionen
 - automatisch erstellte Einkaufsliste
 - automatisch erstellter Wochenplan der geplanten Mahlzeiten

2.2 Grundlagen Ernährung

2.2.1 Kalorienbilanz & Makronährstoffe

Lebensmittel enthalten viele verschiedene Nährstoffe. Unterteilen kann man diese in Makronährstoffe, Vitamine (Mikronährstoffe) und Mineralstoffe.

Folgende 3 Nährstoffe werden als Makronährstoffe bezeichnet:

Makronährstoff	Kaloriendichte
Fette	9,3 kcal/g
Kohlenhydrate	4,1 kcal/g
Proteine	4,1 kcal/g

Tabella 7: Makronährstoffe Kaloriendichte

Aus der Summe der zugeführten Makronährstoffe errechnet sich die Kalorienbilanz.

Vitamine, auch Mikronährstoffe genannt, sowie Mineralstoffe haben keinen Brennwert. Alkohol liefert 7,1 kcal pro Gramm und Ballaststoffe können mit 2 kcal/g vernachlässigt werden, aufgrund der geringen zugeführten Menge. Empfohlen wird die tägliche Aufnahme von ungefähr 30 g Ballaststoffen, was unwesentlichen 60 kcal entspricht.

Vereinfacht kann man sagen:

Führt man mehr Kalorien zu sich, als man benötigt, nimmt man zu. Nimmt man weniger auf, als man benötigt, bedient sich der Körper des Körperfetts als Energiequelle und man nimmt ab.

Wie viel Energie man täglich benötigt und in welchem Verhältnis die Makronährstoffe zur Kaloriensumme beitragen sollten ist von verschiedenen Faktoren abhängig.

Plant man die täglich zugeführten Nährwerte entsprechend an diese Faktoren angepasst, um ein gewünschtes Ziel bzw. eine gewünschte Veränderung des Körpers zu erreichen, spricht man von einer bilanzierten Ernährung.

Solche Faktoren sind:

- Alter
- Geschlecht
- Größe
- Gewicht
 - Verteilung des Gewichts auf die verschiedenen Gewebetypen
 - Körperfett, Muskelmasse etc.
 - sonstige Körperbestandteile (in untergeordnetem Maß)
 - Knochengewicht, Wasseranteil etc.
- körperliche & geistige Aktivität
- Ziel
 - Was möchte ich erreichen?
 - z. B. Muskelaufbau oder Fettabbau

2.2.2 EatFit Nährstoffübersicht

Folgende Tabelle zeigt die in EatFit verfügbaren Nährstoffe und sonstige Werte, sowie die dazugehörige Einheit.

Makronährstoffe	Mineralstoffe	Vitamine
Fett (g)	Chlor (mg)	Vitamin A (mg)
davon gesättigte Fettsäuren (g)	Eisen (mg)	Vitamin B1 (mg)
davon einfach ungesättigte Fettsäuren (g)	Fluor (mg)	Vitamin B2 (mg)
davon mehrfach ungesättigte Fettsäuren (g)	Jod (µg)	Vitamin B3 (mg)
Kohlenhydrate (g)	Kalium (mg)	Vitamin B5 (mg)
davon Zucker (g)	Kalzium (mg)	Vitamin B6 (mg)
Protein (g)	Kupfer (mg)	Vitamin B7 (µg)
Sonstige Stoffe	Magnesium (mg)	Vitamin B9 (µg)
Brennwert (kcal)	Mangan (mg)	Vitamin B12 (µg)
Ballaststoffe (g)	Molybdän (µg)	Vitamin C (mg)
Salz (g)	Natrium (mg)	Vitamin D (µg)
Wasser (ml)	Phosphor (mg)	Vitamin E (mg)
Cholesterin (mg)	Schwefel (mg)	
Broteinheiten (BE)	Selen (µg)	
Purin (mg)	Zink (mg)	
Alkohol (g)		

Tabelle 8: Nährstoffübersicht

EatFit berücksichtigt all diese Nährstoffe, um dem Benutzer bei der individuellen Planung möglichst viele Parameter zu bieten. Der Anwender ist natürlich nicht dazu gezwungen, auch all diese Nährwerte zu verwenden.

2.2.3 Festlegen der Zielwerte

Im Internet findet man zur Berechnung des Energiebedarfs, sowie der Nährstoffverteilung, verschiedene Formeln. Auch einige Softwareprodukte erlauben es sich, aufgrund weniger Eingaben eines Benutzers, unter Angabe seines Ziels, Zielwerte vorzuschlagen, welche häufig so auch blind übernommen werden.

EatFit verlangt vom Benutzer weder die Eingabe persönlicher Daten, noch werden Zielwerte vorgeschlagen. Das Projektteam besitzt keine Ausbildung bezüglich Ernährungsberatung, weshalb das Programm ausschließlich zur Planung und Dokumentation der Ernährung dient. Zielwerte können nur vom Benutzer manuell eingegeben werden. Ratsam ist es diese Zielwerte zusammen mit einem Experten zu eruieren.

2.2.4 Beispiel Zielwerte & Mahlzeitplanung

Wir nehmen an, aus einem Beratungsgespräch kennen wir folgende Zielwerte:

	Fett	Kohlenhydrate	Protein	Kalorien
Menge	59 g	235 g	147 g	2114,9 kcal

Tabelle 9: Beispiel Zielwerte

Die Berechnung von Kalorien erfolgt folgendermaßen:

59 g Fett	*	9,3 kcal	=	548,7 kcal
235 g Kohlenhydrate	*	4,1 kcal	=	963,5 kcal
147 g Protein	*	4,1 kcal	=	602,7 kcal
				<u>2114,9 kcal</u>

Nun planen wir folgende Mahlzeit:

300 g Putenbrust
150 g Reis

Für den nächsten Schritt muss man die Nährwerte pro 100 g wissen:

Produkt	Fett	Kohlenhydrate	Protein	Kalorien
Putenbrust	0,8 g	0 g	23,2 g	102,56 kcal
Reis	2,2 g	74,1 g	7,8 g	356,25 kcal

Tabelle 10: Beispiel Nährwerte pro 100 g

Jetzt rechnen wir diese Nährwerte entsprechend der geplanten Menge hoch und ziehen diese von unseren Zielwerten ab, um zu wissen, wie viel zur weiteren Planung übrig bleibt:

	Fett	Kohlenhydrate	Protein	Kalorien
Zielwert	59 g	235 g	147 g	2114,9 kcal
- Putenbrust	2,4 g	0 g	69,6 g	307,68 kcal
- Reis	3,3 g	111,15 g	11,7 g	534,375 kcal
Rest (Differenz)	53,3 g	123,85	65,7	1272,845 kcal

Tabella 11: Beispiel Zielwerte & Differenz

Nun müsste man für die nächste Mahlzeit wieder aufwändig Rechnungen durchführen, doch entsprechende Programme wie EatFit übernehmen diesen Rechenaufwand. Wie die Planung bei EatFit abläuft wird später in einem Beispiel durchexerziert im Punkt 6.2.

3 Projektziele

3.1 Anwenderziele

3.1.1 Bilanzierte Ernährung

Das Primärziel von EatFit ist es, dem Anwender eine einfache und schnelle Ernährungsplanung zu ermöglichen. Durch die Möglichkeit von exakten Werteingaben bei Produkt-, Mahlzeit und Zielwerten, kann der Anwender eine bilanzierte Ernährung umsetzen.

3.1.2 Gesundheit & Sport

Durch eine entsprechend geplante und ausgelegte Ernährung kann der Anwender seine gesundheitlichen & sportlichen Ziele erreichen. Dazu gehören beispielsweise das bewusste Erhöhen des täglichen Gemüse- & Obstkonsums, Reduktion des Fleischkonsums, Muskelaufbau und Fettabbau.

3.2 Ökonomische Ziele

Durch die automatisch generierte Einkaufsliste, zeigt EatFit dem Anwender genau, welche Produkte zur Umsetzung des Ernährungsplans einzukaufen sind. Somit wird nur eingekauft, was gebraucht wird, was die Menge von Lebensmittelabfällen einschränken kann, welche entstehen können, wenn ohne Plan eingekauft wird. Dadurch kann der Anwender seine Ausgaben für Lebensmittel senken.

3.3 Ökologische Ziele

Lebensmittelabfälle reduzieren ist darüber hinaus nicht nur schonend für den Geldbeutel, sondern auch umweltbewusst und ressourcenschonend.

4 Grundlagen und Technologien

Erkol Beytullah

4.1 Technologien zur Erstellung

4.1.1 HTML

Die **Hypertext Markup Language** dient zur Darstellung von bestimmten Inhalten auf einer Webseite. Vorteil dabei ist die Plattformunabhängigkeit, jedoch erfolgt keine Formatierung des Inhaltes.



Abbildung 4: HTML 5 Logo

4.1.2 CSS



Abbildung 5: CSS3 Logo

Cascading Style Sheets ist eine Stylesheet-Sprache und gehört zu den Kernsprachen des World Wide Webs. In Verbindung mit HTML ermöglicht es eine Gestaltungsmöglichkeit der Webseite.

4.1.3 PHP

Die Sprache **Hypertext Preprocessor** ist eine serverseitige Programmiersprache. Eine Anwendung erfolgt in der Webprogrammierung, wo es in eine HTML-Seite eingebettet wird. Diese Sprache führt Anfragen auf dem Server aus und liefert die Ergebnisse dem Entwickler zurück. Der Client merkt die Anfragen im Hintergrund dabei nicht und sieht nur das Ergebnis.



Abbildung 6: php Logo

4.1.4 MySQL



Abbildung 7: MySQL Logo

MySQL ist ein relationales Datenbankverwaltungssystem und eine Open-Source Software, jedoch gibt es auch kostenpflichtige Versionen, die mehr Funktionalitäten bieten. Oft kommt es bei Webapplikationen zur Anwendung.

4.1.5 MySQLi

MySQLi ist eine Erweiterung für den Zugriff auf MySQL Datenbank, dies kann objektorientiert oder prozedural erfolgen. Der riesen Vorteil dieser Erweiterung sind die Prepared Statements, welches *SQL-Injections verhindert

*SQL-Injection bezeichnet das Ausnutzen einer Sicherheitslücke in Zusammenhang mit SQL-Datenbanken, die durch mangelnde Maskierung oder Überprüfung von Metazeichen in Benutzereingaben entsteht.

[Quelle: MySQL 5.1 Referenzhandbuch, Kapitel 20]

4.1.6 JavaScript



Abbildung 8: JavaScript Logo

JavaScript ist eine Programmiersprache die mit Verbindung von HTML eine dynamische Webseite bildet. Dabei können Elemente aktualisiert, verändert oder ausgelesen werden ohne die Seite neu zu laden.

4.1.7 jQuery

jQuery ist eine kostenlose JavaScript Bibliothek die eine *DOM-Navigation und –Manipulation zur Verfügung stellt.

*Document Object Model (DOM) ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente.

[Quelle: de.wikipedia.org/wiki/Document_Object_Model]



Abbildung 9: jQuery Logo

4.1.8 AJAX

AJAX (Asynchronous JavaScript and XML) bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden.

[Quelle: Hannes Pfeiffer: Ajax: Modernes Webscripting. Data Becker, August 2006]

4.1.9 Windows Server 2012



Abbildung 10: Windows Server Logo

Windows Server 2012 entwickelt vom Großkonzern Microsoft ist eine Server-Version angelehnt an das Betriebssystem Windows 8. Aktuelle Version ist der Windows Server 2012 R2 (Stand 30.3.2016).

4.1.10 Android

Das Betriebssystem **Android** ist eine Software-Plattform für mobile Geräte wie Smartphones, Netbooks und Tablets. Die Entwickler dieses Betriebssystems ist das Unternehmen Google Inc., als Basis dient der Linux-Kernel.



Abbildung 11: Android Logo

4.2 Software zur Umsetzung

4.2.1 phpMyAdmin

Die Open-Source Webanwendung **phpMyAdmin** dient zur Administration von MySQL-Datenbanken. Generelle Funktionen wie z.B. Tabellen anzeigen/löschen, benötigt der Nutzer keine SQL-Anweisungen zu schreiben.



Abbildung 12: phpMyAdmin Logo

4.2.2 Notepad ++

Notepad++ ist ein Texteditor für Windows und andere kompatible Betriebssysteme.

4.2.3 Eclipse Mars PHP Developer

Eclipse Mars PHP Developer ist eine Entwicklungsumgebung von Eclipse Foundation, welche ins besondere auf die Programmiersprache PHP spezialisiert ist.



Abbildung 13: Eclipse Logo

4.2.4 Firebug



Firebug ist ein kostenloses Add-on für den Webbrowser Mozilla Firefox, welches dem Entwickler bei der Fehlersuche und Monitoring das Leben vereinfacht.

Abbildung 14: Firebug Logo

4.2.5 Google Drive

Google Drive ermöglicht Benutzern das Speichern von Dokumenten in der Cloud, das Teilen von Dateien und das gemeinsame Bearbeiten von Dokumenten.



Abbildung 15: Google Drive Logo

[Quelle: Walter S. Mossberg: Google Stores, Syncs, Edits in the Cloud, The Wall Street Journal. 25. April 2012]

4.2.6 VMware vSphere Client

Der **VMware vSphere-Client** ist eine Windows-Anwendung, welche interaktiv den Zugang zu den Ressourcen eines vSphere-Servers ermöglicht. Ermöglicht Funktionen wie Start/Stopp virtueller Maschinen, Erstellen/Löschen von virtuellen Maschinen etc.



Abbildung 16: VMware Logo

[Quelle: Bertram Wöhrmann, Carsten Schäfer, Günter Baumgart, Urs Stephan Alder, Marcel Brunner: VMware vSphere 6: Das umfassende Handbuch. Rheinwerk-Verlag, Reihe Computing]

4.2.7 IIS

Die **Internet Information Services** ist ein Webserver, welches durch die Kommunikationsprotokolle HTTP, HTTPS, FTP, SMTP, POP3 und WebDAV, Dokumente und Dateien im Netz zugänglich macht.

4.2.8 FTP

Der **File Transfer Protocol** dient zur Übertragung von Dateien über die IP-Netzwerke. FTP ermöglicht Dateien vom Server Herunterzuladen und auch Hochzuladen. Es können auch Verzeichnisse erstellt und ausgelesen werden.

4.2.9 Android Studio



Abbildung 17: Android Studio Logo

Android Studio ist eine kostenlose Entwicklungsumgebung für mobile Applikationen, welche von Google bereitgestellt wird.

4.2.10 DB Designer Fork

Die Software **DB Designer Fork** ist ein Datenmodell Designer, welches Entity Relationship Design und SQL Export integriert. Es können DDLs für SQL Server, MySQL, FireBird, SQLite und PostgreSQL erstellt werden.

4.2.11 FileZilla Client

FileZilla ist ein FTP-Client Software, es erstellt eine Verbindung mit dem FTP-Server. Es können mehrere Verbindungen abgespeichert werden, die Benutzeroberfläche ist übersichtlich und daher einfach zu bedienen.



Abbildung 18: FileZilla Logo

5 Umsetzung

5.1 Webanwendung

5.1.1 GUI Planung

Die ersten Entwürfe zur grafischen Benutzeroberfläche entstanden auf Papier. Um Änderungen schneller und besser durchführen zu können und um die Entwürfe mobiler zu haben, wurde in weiterer Folge Moqups verwendet.

The image shows a wireframe of a web application interface. At the top, there is a navigation bar with the following items: 'Wochenübersicht', 'Tagesbilanz', 'Mahlzeit planen', 'Meine Lebensmittel', 'Drucken', 'Profil & Einstellungen', 'Abmelden', and 'Benutzername'. Below the navigation bar, the main content area is titled 'Profil bearbeiten'. On the right side of this area, there is a dropdown menu with 'Profil' selected and 'Zielsetzung' as an option. The main content area contains a form titled 'Passwort' with three input fields: 'altes Passwort', 'neues Passwort', and 'neues Passwort wiederholt'. Below these fields is a button labeled 'Passwort ändern'. At the bottom of the form area, there is a button labeled 'Profil löschen'.

Abbildung 19: GUI Planung

5.1.2 GUI Umsetzung

Die Umsetzung der grafischen Benutzeroberfläche erfolgte mit HTML und CSS. HTML gibt die Struktur und Anordnung der Elemente vor. CSS ist für das Design, Farben etc. verantwortlich.

The image shows the implemented version of the web application interface. It features a logo at the top left consisting of a red heart shape with a green figure inside, and the text 'EatFit' below it. The navigation bar is now a solid orange color with white text for the items: 'Wochenübersicht', 'Tagesbilanz', 'Mahlzeit planen', 'Meine Lebensmittel', 'Profil & Einstellungen', 'Abmelden', and 'Benutzername'. The main content area has a yellow background. The title 'Profil' is displayed in bold. The form fields are now white with black text and borders. The buttons are orange with white text. The overall design is clean and modern.

Abbildung 20: GUI umgesetzt

5.1.3 Verschlüsselung

Wie jede Anwendung mit einem Login sollte das Benutzerpasswort nicht im Klartext in der Datenbank abgespeichert sein, sondern mit einer Methode verschlüsselt und erst danach gespeichert werden. Da ansonsten die Arbeit der Angreifer enorm erleichtert wird und man grundsätzlich Passwörter nie im Klartext abspeichert.

Die Verschlüsselungsmethode *MD5 kam bei der Verschlüsselung und Entschlüsselung der Benutzer Passwörter zum Einsatz. In PHP existiert die Methode „md5()“, es errechnet den MD5-Hash eines Strings.

```
$password = md5($_POST['password']);
```

Bei der Erstellung des Nutzers wird das Passwort direkt verschlüsselt in die Datenbank gespeichert und beim Einloggen gleicht die Anwendung das eingegebene Passwort vom Benutzer mit dem aktuellen Passwort in der Datenbank ab. Falls das Passwort falsch ist wird der Benutzer wieder zum Login Fenster verlinkt. Bei korrekter Eingabe des Passwortes wird der Benutzer erfolgreich eingeloggt.

```
//Create query
$qry = "SELECT * FROM benutzer WHERE email='$email' AND password='$password' AND verfuegbar=0";

//Check whether the query was successful or not
if($result = mysqli_query($link, $qry)) {
    if(mysqli_num_rows($result) > 0) {
        //Login Successful
        session_regenerate_id();
        $member = mysqli_fetch_assoc($result);
        $_SESSION['SESS_MEMBER_ID'] = $member['idBenutzer'];
        $_SESSION['SESS_EMAIL'] = $member['email'];
        $_SESSION['SESS_PW'] = $member['password'];
        header("location: wochenuebersicht.php");
        exit();
    }else {
        //Login failed
        echo "User not found";
        $errflag = true;
        if($errflag) {
            header("location: login.php");
            exit();
        }
    }
}else {
    die("Query failed");
}
```

*MD5 -Message-Digest Algorithm 5 ist eine weit verbreitete kryptographische Hashfunktion, die aus einer beliebigen Nachricht einen 128-Bit-Hashwert erzeugt.

[Quelle: Hans Dobbertin: Cryptanalysis of MD5 compress. Announcement on Internet, Mai 1996]

5.1.4 Aufbau

Die Webseite unterteilt sich in drei Teile, diese sind einmal der Header, die Navigation und der Section Bereich. Diese werden auch die semantischen HTML5 Elemente genannt. Als Basis diente ein unsere Ansprüche erfüllender Template, auf diesem wurden die Farben angepasst und Elemente die unbrauchbar sind entfernt. Jede einzelne Seite ist mit HTML5 und CSS aufgebaut, was viel Mühe und Zeit gekostet hat.

Nach erfolgreichem Login erscheint die Wochenübersicht des Benutzers. Auf jeder einzelnen Seite existiert eine Überschrift der aktuellen Seite.

```
<header>
  
  <a href="logout.php">
    
  </a>
</header>
```

Die Navigation ist als ein DropDown Menü realisiert worden, da einige Funktionalitäten unter einem einzigen Menüpunkt zusammengefasst werden können. Realisierung eines DropDowns erfolgt ausschließlich über CSS, da HTML kein vorgefertigtes Element hat.

```

<nav>
  <ul class="drop_menu">
    <li><a href="wochenuebersicht.php">Wochenuebersicht</a></li>
    <li><a href="tagesbilanz.php">Tagesbilanz</a></li>
    <li><a href="mahlzeit_planen.php">Mahlzeit planen</a></li>
    <li><a href="#">Meine Lebensmittel</a>
      <ul>
        <li><a href="meine_lebensmittel_vorrat.php">Vorrat</a></li>
        <li><a href="meine_lebensmittel_verwaltung.php">Verwaltung</a></li>
        <li><a href="einkaufsliste.php">Einkaufsliste</a></li>
      </ul>
    </li>
    <li><a href="#">Profil & Einstellungen</a>
      <ul>
        <li><a href="profil.php">Profil</a></li>
        <li><a href="profil_zielsetzung.php">Zielsetzung</a></li>
      </ul>
    </li>
  </ul>
</nav>

```

Je nach Auswahl eines Menüpunktes erfolgt die Anzeige des Inhaltes im Section Bereich, dieser wird öfters auch in kleineren Bereichen aufgeteilt, um die Übersicht zu verbessern.

In der folgenden Abbildung ist der Aufbau des Untermenüpunktes „Profil“ zu sehen.

```

<section>
  <h1>Profil</h1>
  <form name="passwordform" action="change_pw.php" method="POST">
    <table>
      <tr>
        <td class="standart_font">Altes Passwort:</td>
        <td><input type="password" name="old_pw" required="required" autofocus/></td>
      </tr>
      <tr>
        <td class="standart_font">Neues Passwort:</td>
        <td><input type="password" name="nev_pw" required="required"/></td>
      </tr>
      <tr>
        <td class="standart_font">Neues Passwort WH:</td>
        <td><input type="password" name="nev_pw_wh" required="required"/></td>
      </tr>
      <tr>
        <td><input type="submit" name="save" value="Passwort speichern" class="btn"/></td>
      </tr>
    </table>
  </form>
  <table>
    <tr>
      <td><input type="submit" name="dactive" value="Profil deaktivieren" class="btn" onclick="window.location.href='dactive_user.php'"/></td>
    </tr>
  </table>
</section>

```

5.1.5 connection.php

Diese PHP Datei dient als eine Verbindungsbrücke zu der Datenbank. Sie muss in jeder Datei inkludiert werden, um ein Zugriff auf die Datenbank zu haben. Die Variable „\$link“ ist der Connection-String zur Datenbank und wird daher in Methoden die Queries ausführen z.B. „mysql_query()“ als Übergabe-Parameter benötigt.

Die folgende Abbildung zeigt den Code in connection.php.

```
<?php
$link = mysqli_connect('172.16.12.42', 'root', 'EatFit1S%', 'eatfit_db');

if (!$link) {
    echo "Error: Unable to connect to MySQL." . PHP_EOL;
    echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
    echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
    exit;
}
?>
```

5.1.6 auth.php

Das nicht Angreifer auf eine bestimmte Seite mit einem direkten Link Zugriff haben wird die „auth.php“ Datei auf diese Seiten inkludiert. Beim Einloggen des Users werden Session-Variablen erzeugt und „auth.php“ kontrolliert ob diese gesetzt sind. Falls nein wird man auf die Login Seite geleitet.

In der nachfolgenden Abbildung ist die Überprüfung der Session Variablen zu sehen.

```
<?php
//Start session
session_start();
//Check whether the session variable SESS_MEMBER_ID is present or not
if (!isset($_SESSION['SESS_MEMBER_ID']) || (trim($_SESSION['SESS_MEMBER_ID']) == '')) {
    header("location: login.php");
    exit();
}
?>
```

5.1.7 Session Variablen

Es gibt insgesamt drei Session Variablen: `SESS_MEMBER_ID` (Identifikation des Benutzers), `SESS_FIRST_NAME` (Vorname des Benutzers), `SESS_LAST_NAME` (Nachname des Benutzers).

Mit der PHP Methode „`session_start()`“ startet die Session und man bekommt ein Zugriff auf diese drei Variablen. Sie werden beim Einloggen des Benutzers angelegt und beim Ausloggen gelöscht. Der Zugriff auf eine Session-Variable erfolgt durch „`$_SESSION[]`“.

5.1.8 Datenaustausch

Größtenteils findet die Datenabfrage auf die Datenbank mittels AJAX statt, weil dadurch die aktuelle Seite nicht neu geladen werden muss. Der Benutzer merkt daher von dieser Abfrage gar nichts und die Wartezeiten sind auch gering. Jeder Datenaustausch erfolgt nach einem Event, welches durch Interaktionen des Nutzers ausgelöst wird.

In einem AJAX Aufruf erfolgt ein „POST“ auf eine PHP Seite, in diesem übergibt es jegliche Parameter mit. PHP übernimmt diese Parameter mittels der Funktion „`$_POST[]`“ und speichert sie in eine PHP Variable.

Abfragen auf die Datenbank erfolgen mittels den MySQLi Methoden „`mysqli_query()`“ oder „`mysqli_prepare()`“. Um SQL-Injection zu verhindern erfolgten die Abfragen eher mit Prepared Statements.

Beispielsweise bei einem Select-Query erhält man Daten von der Datenbank, diese werden dann in ein PHP Array gespeichert. Nach der erfolgreichen Ausführung der SQL-Query, wird der Array als ein JSON String zurück geliefert. Dies geschieht mittels der Methode „`json_encode()`“.

In der folgenden Abbildung handelt es sich um die Suchfunktion nach einem Produkt in der Datenbank.

```
<?php
session_start();
require_once('connection.php');

// Parameter
$user_id = $_SESSION['SESS_MEMBER_ID'];
$searchstring = $_POST['p'];
$products = array();

// Query "auswählen"
if(strlen($searchstring) == 0) {
    $sql_search_product = "SELECT * FROM `produkt`";
} else {
    $sql_search_product = "SELECT * FROM `produkt` WHERE `produktName` LIKE '%$searchstring%' AND\n"
        . "`Benutzer_idBenutzer` = '$user_id' OR `Benutzer_idBenutzer` IS NULL AND\n"
        . "`produktName` LIKE '%$searchstring%'";
}

if($search_result = mysqli_query($link, $sql_search_product))
{
    while($row=mysqli_fetch_array($search_result))
    {
        $products[] = $row;
    }
    echo json_encode($products);
} else {
    printf("Errormessage: %s\n", mysqli_error($link));
}

?>
```

In dieser Abbildung sind der Aufruf von AJAX und die Behandlung des zurückgegebenen JSON Strings zu sehen. Mit der Methode „JSON.parse()“ wird der JSON String in ein Javascript Object umgewandelt.

```
//Produkte werden in eine Liste geladen
var products;
var loadData = function() {
    var product = $("#searchstring").val();
    $.ajax({
        type:"POST",
        url:"searchproduct.php",
        data: {
            'p': product
        }

    }).done(function(data) {
        products = JSON.parse(data);
        var row = "";
        row += "<select id='produkte' size='15'>";
        for(var i in products){
            row += "<option value="+ products[i].idProdukt+">";
            row += products[i].produktName;
            row += "</option>";
        }
        row += "</select>";
        $('#produkte').html(row);
    });
}
```

5.1.9 Verwendung JQuery

Die JavaScript Bibliothek JQuery kam öfters zur Verwendung, da es das Arbeiten an der DOM deutlich erleichtert. Dynamische Aktualisierungen auf der Webseite sind daher ohne Problem durchzuführen. JQuery ermöglicht Elemente zu selektieren und sie zu bearbeiten bzw. auszulesen, natürlich kann man sie auch entfernen oder neue hinzufügen.

5.1.10 Datetimepicker

Da HTML ein Eingabefeld mit dem Typ Datum nur für Google Chrome Browser zur Verfügung stellt, kam ein Datetimepicker Plugin zum Einsatz, welches JQuery nutzt um ein Datum auszuwählen. Damit kann man jetzt in jedem Browser ein Datum im Eingabefeld auswählen.

Angabe der Dateipfade vom Plugin:

```
<link rel="stylesheet" type="text/css" href="plugins/datetimepicker-master/jquery.datetimepicker.css"/ >  
<script src="plugins/datetimepicker-master/jquery.js"></script>  
<script src="plugins/datetimepicker-master/build/jquery.datetimepicker.full.min.js"></script>
```

Initialisierung eines Eingabefeldes mit einer Id:

```
<tr>  
  <td class="standart_font">Datum: <input id="datetimepicker" type="text"></td>  
</tr>
```

Anwendung des Datetimepicker auf das Eingabefeld:

```
//Datumauswahl JQuery Plugin  
$.datetimepicker.setLocale('de');  
$('#datetimepicker').datetimepicker({  
  timepicker:false,  
  format: 'Y-m-d'  
});
```

5.2 Android-Applikation

Die Applikation stellt die mobile Einkaufsliste dar und ist somit relativ kompakt gehalten. Es gibt nur zwei grafische Oberflächen. Nach dem Starten der Applikation erscheint das Login-Formular. Nach erfolgreichem Login, sieht der Benutzer seine Einkaufsliste, entsprechend dem ausgewählten Zeitraum. Im Folgenden sieht man Teile von Planung und Umsetzung der Applikation.

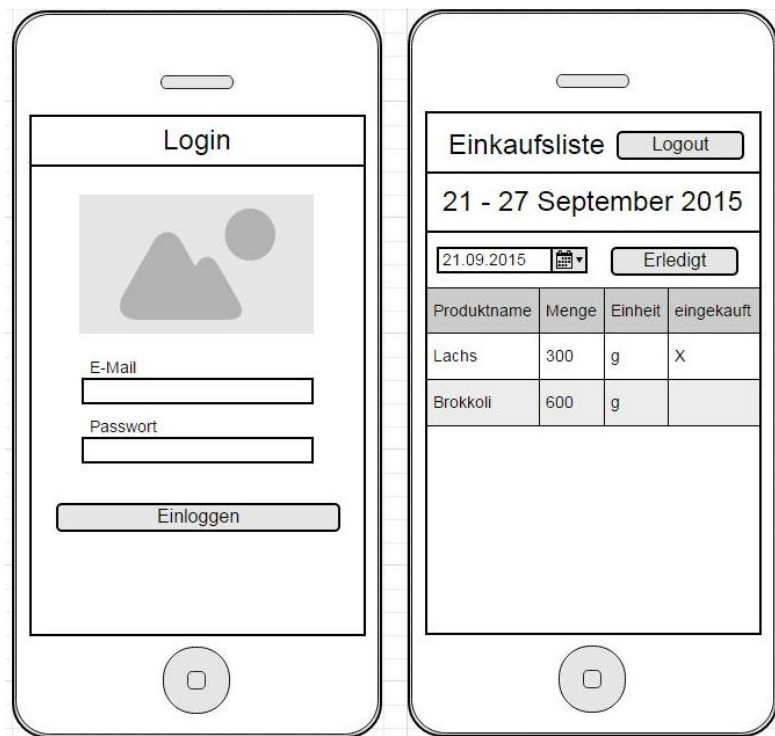


Abbildung 21: App Planung

Die Struktur der Pages wird mittels XML festgelegt. Folgender Codeausschnitt aus der Datei activity_login.xml zeigt die Struktur der Login-Page, im speziellen die Eingabefelder für E-Mail und Passwort.

```
<android.support.design.widget.TextInputLayout android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <AutoCompleteTextView android:id="@+id/email" android:layout_width="match_parent"
        android:layout_height="wrap_content" android:hint="E-Mail-Adresse"
        android:inputType="textEmailAddress" android:maxLines="1"
        android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText android:id="@+id/password" android:layout_width="match_parent"
        android:layout_height="wrap_content" android:hint="Passwort"
        android:imeActionId="@+id/login"
        android:imeActionLabel="Sign in"
        android:imeOptions="actionUnspecified" android:inputType="textPassword"
        android:maxLines="1" android:singleLine="true" />

</android.support.design.widget.TextInputLayout>
```

Der jetzige Codeausschnitt aus LoginActivity.java gibt einen Einblick in die Funktionalität, der Login-Page. **onCreate** wird beim Starten der Applikation aufgerufen und ist im Weiteren dafür verantwortlich, dass ein Login ermöglicht wird.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    // Set up the login form.
    mEmailView = (AutoCompleteTextView) findViewById(R.id.email);
    populateAutoComplete();

    mPasswordView = (EditText) findViewById(R.id.password);
    mPasswordView.setOnEditorActionListener(new TextView.OnEditorActionListener() {
        @Override
        public boolean onEditorAction(TextView textView, int id, KeyEvent keyEvent) {
            if (id == R.id.login || id == EditorInfo.IME_NULL) {
                attemptLogin();
                return true;
            }
            return false;
        }
    });

    Button mEmailSignInButton = (Button) findViewById(R.id.email_sign_in_button);
    mEmailSignInButton.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View view) {
            attemptLogin();
        }
    });

    mLoginFormView = findViewById(R.id.login_form);
    mProgressView = findViewById(R.id.login_progress);
}
```

attemptLogin "versucht" den Login mit den Benutzereingaben durchzuführen und überprüft dabei deren Richtigkeit. Weitere Methoden überprüfen auch noch, ob zum Beispiel überhaupt ein @-Zeichen in der E-Mail-Adresse vorkommt, welches ja Pflicht ist.

5.3 Datenbank

Als Datenbank haben wir uns für MySQL entschieden, welche die Login Daten des Benutzers und zugleich die Daten von einzelnen Benutzern der Webanwendung speichert. Für die Verwaltung und Pflege der Datenbank ist phpMyAdmin zuständig, welche eine übersichtliche GUI und diverse Funktionalitäten anbietet.

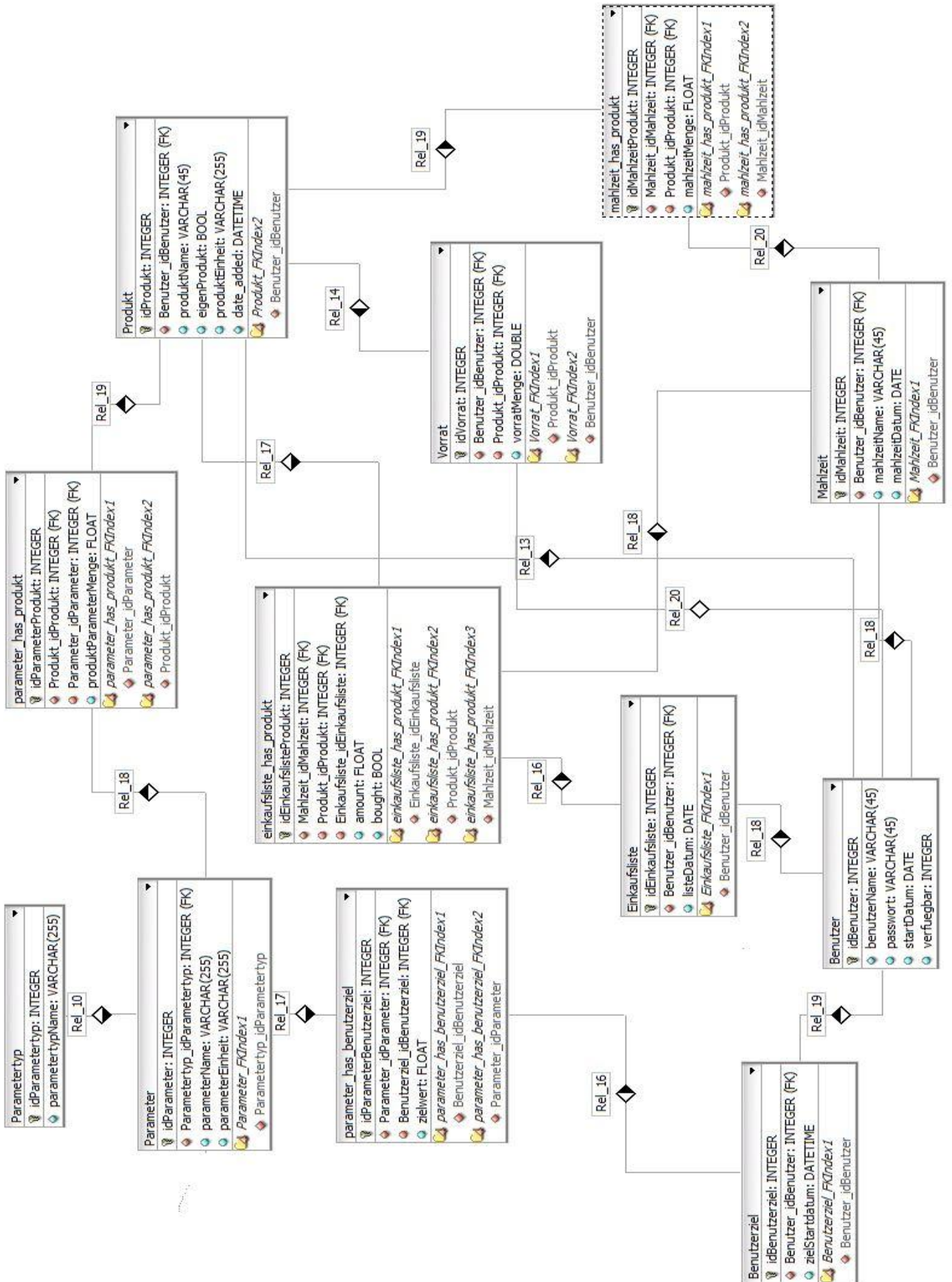
5.3.1 Standard Daten

Die Datenbank enthält von „EatFit“ ausgewählte Produkte mit ihren genauen Nährwerten, diese können vom Benutzer verwendet und geplant werden. Der eingeloggte Benutzer darf nur seine eigene Produkte ändern und löschen, aber nicht die ursprünglichen Produkte der Webanwendung.

5.3.2 Datenmodell

Für die Abbildung der Datenstruktur haben wir ein Entity-Relationship-Modell (ERD) gebildet welches laufend aktualisiert wurde, da Inkonsistenzen entstanden sind und daher Verbesserung durchgeführt werden mussten.

In der nachfolgenden Abbildung ist der endgültige ERD zu sehen.



5.3.3 Beschreibung der Datenbanktabellen

Die Benutzer Tabelle beinhaltet Login Daten des Benutzers

Tabelle Benutzer	
Field	Erklärung
idBenutzer	Der Identifikationscode eines Benutzers
email	E-Mail Adresse des Benutzers, zum Einloggen benötigt
passwort	Das vom Benutzer festgelegtes Passwort
verfuegbar	Flag um ein Benutzer zu deaktivieren
startdatum	Erstelldatum des Benutzers

Tabelle 12: Tabelle Benutzer

Die Benutzerziel Tabelle enthält die Ziele eines Benutzers daher steht es in einer 1:n Beziehung mit der Tabelle Benutzer.

Tabelle Benutzerziel	
Field	Erklärung
idBenutzerziel	Der Identifikationscode eines Benutzerzieles
idBenutzer	Fremdschlüssel der auf den Benutzer verweist
zielStartdatum	Erstelldatum eines Benutzerzieles

Tabelle 13: Tabelle Benutzerziel

Die Parametertyp Tabelle dient zur Speicherung der drei Basis Nährwerte (Mineralstoffe, Vitamine und Makronährstoffe).

Tabelle Parametertyp	
Field	Erklärung
idParametertyp	Der Identifikationscode eines Parametertypes
parametertypName	Name eines Parametertypes

Tabelle 14: Tabelle Parametertyp

Die Parameter Tabelle beinhaltet einen bestimmten Nährwert wie z.B. Fett. Sie steht in einer 1:n Beziehung mit der Tabelle Parametertyp.

Tabelle Parameter	
Field	Erklärung
idParameter	Der Identifikationscode eines Parameters
idParametertyp	Fremdschlüssel der auf den Parametertyp verweist
parameterName	Name eines Parameters
parameterEinheit	Einheit eines Parameters

Tabelle 15: Tabelle Parameter

Die Produkt Tabelle beinhaltet die vorhandenen Produkte des Benutzers. Sie steht in einer 1:n Beziehung zum Benutzer, da jeder Benutzer beliebig viele Produkte anlegen kann.

Tabelle Produkt	
Field	Erklärung
idProdukt	Der Identifikationscode eines Produktes
Benutzer_idBenutzer	Fremdschlüssel der auf den Benutzer verweist
produktName	Name eines Produktes
eigenProdukt	Flag ob das Produkt selbst erstellt wurde
produktEinheit	Einheit des Produktes
date_added	Erstelldatum eines Produktes

Tabelle 16: Tabelle Produkt

Die Vorrat Tabelle beinhaltet die Menge von einem Produkt, welches zuhause lagernd ist. Sie steht in einer 1:n Beziehung mit die Tabellen Benutzer und Produkt.

Tabelle Vorrat	
Field	Erklärung
idVorrat	Der Identifikationscode eines Vorrates
Benutzer_idBenutzer	Fremdschlüssel der auf den Benutzer verweist
Produkt_idProdukt	Fremdschlüssel der auf ein Produkt verweist
vorratMenge	Menge des jeweiligen Produktes, welches zuhause vorhanden ist

Tabelle 17: Tabelle Vorrat

Die Mahlzeit Tabelle dient zur Speicherung der geplanten Mahlzeiten des Benutzers, daher steht sie in einer 1:n Beziehung mit der Benutzer Tabelle.

Tabelle Mahlzeit	
Field	Erklärung
idMahlzeit	Der Identifikationscode eines Mahlzeit
Benutzer_idBenutzer	Fremdschlüssel der auf den Benutzer verweist
mahlzeitName	Name der Mahlzeit
mahlzeitDatum	Erstelldatum eines Mahlzeit

Tabelle 18: Tabelle Mahlzeit

Die Einkaufsliste Tabelle beinhaltet vom jeweiligen Benutzer eine Einkaufsliste.

Tabelle Einkaufsliste	
Field	Erklärung
idEinkaufsliste	Der Identifikationscode einer Einkaufsliste
Benutzer_idBenutzer	Fremdschlüssel der auf den Benutzer verweist
listeDatum	Erstelldatum der Einkaufsliste

Tabelle 19: Tabelle Einkaufsliste

Die Parameter_has_Benutzerziel ist eine aus der m:n Beziehung zwischen den Tabellen Parameter und Benutzerziel entstandene Tabelle. Sie beinhaltet vom jeweiligen Parameter den Zielwert des Benutzers.

Tabelle Parameter_has_Benutzerziel	
Field	Erklärung
idParameterBenutzerziel	Der Identifikationscode der Parameter-Benutzerziele
p_idParameter	Fremdschlüssel der auf einen Parameter verweist
b_idBenutzerziel	Fremdschlüssel der auf ein Benutzerziel verweist
zielwert	Wert der als Ziel für einen Parameter festgelegt wird

Tabelle 20: Tabelle Parameter_has_Benutzerziel

Die Parameter_has_Produkt ist eine aus der m:n Beziehung zwischen den Tabellen Parameter und Produkt entstandene Tabelle. Sie beinhaltet vom jeweiligen Produkt die Parameter Menge (z.B. Fett 2,5g).

Tabelle Parameter_has_Produkt	
Field	Erklärung
idParameterProdukt	Der Identifikationscode der Parameter-Produkte
Parameter_idParameter	Fremdschlüssel der auf ein Parameter verweist
Produkt_idProdukt	Fremdschlüssel der auf ein Produktl verweist
produktParameterMenge	Menge für ein Parameter

Tabelle 21: Tabelle Parameter_has_Produkt

Die Mahlzeit_has_Produkt ist eine aus der m:n Beziehung zwischen den Tabellen Mahlzeit und Produkt entstandene Tabelle. Sie beinhaltet vom jeweiligen Produkt die Mahlzeit Menge (z.B. Banane 500g).

Tabelle Mahlzeit_has_Produkt	
Field	Erklärung
idMahlzeitProdukt	Der Identifikationscode der Mahlzeit-Produkten
mahlzeit_id	Fremdschlüssel der auf eine Mahlzeit verweist
produkt_id	Fremdschlüssel der auf ein Produkt verweist
mahlzeitMenge	Menge eines Produktes für die Mahlzeit

Tabelle 22: Tabelle Mahlzeit_has_Produkt

Die Einkaufsliste_has_Produkt ist eine aus der m:n Beziehung zwischen den Tabellen Einkaufsliste und Produkt entstandene Tabelle. Sie beinhaltet vom jeweiligen Produkt die einzukaufende Menge (z.B. Brokkoli 250g, Bananen 300g). Die Beziehung zwischen Mahlzeit und Einkaufsliste dient zur Identifikation, zur welcher Mahlzeit das einzukaufende Produkt gehört.

Tabelle Einkaufsliste_has_Produkt	
Field	Erklärung
idEinkaufslisteProdukt	Der Identifikationscode der Einkaufsliste-Produkte
e_idEinkaufsliste	Fremdschlüssel der auf eine Einkaufsliste verweist
p_idProdukt	Fremdschlüssel der auf ein Produkt verweist
m_idMahlzeit	Fremdschlüssel der auf eine Mahlzeit verweist
amount	Menge des zu kaufenden Produktes
bought	Flag ob das Produkt eingekauft ist oder nicht

Tabelle 23: Tabelle Einkaufsliste_has_Produkt

5.4 Server

Als Projekt-Server haben wir einen Schulserver zu Verfügung gestellt bekommen. Auf Ratschlag eines Professors fiel die erste Entscheidung auf einen Linux Server. Es entstanden jedoch später Probleme bei der Konfiguration des Servers. Diese Probleme regten zu einem Wechsel auf Windows an, da dieses Betriebssystem einfacher und übersichtlicher für uns ist.

5.4.1 Webserver (IIS)

Nach der Installation des Webservers war der nächste Schritt den IIS nach unseren Wünschen zu konfigurieren. Nebenbei wurde auch der PHP Manager installiert, welcher es ermöglicht Dateien mit der Endung „.php“ auf dem Server erfolgreich auszuführen.

Die installierte PHP Version ist 5.6.0 und es sind insgesamt 16 PHP Extensions aktiviert, welche uns zum Beispiel den Verbindungsaufbau zur Datenbank ermöglicht.

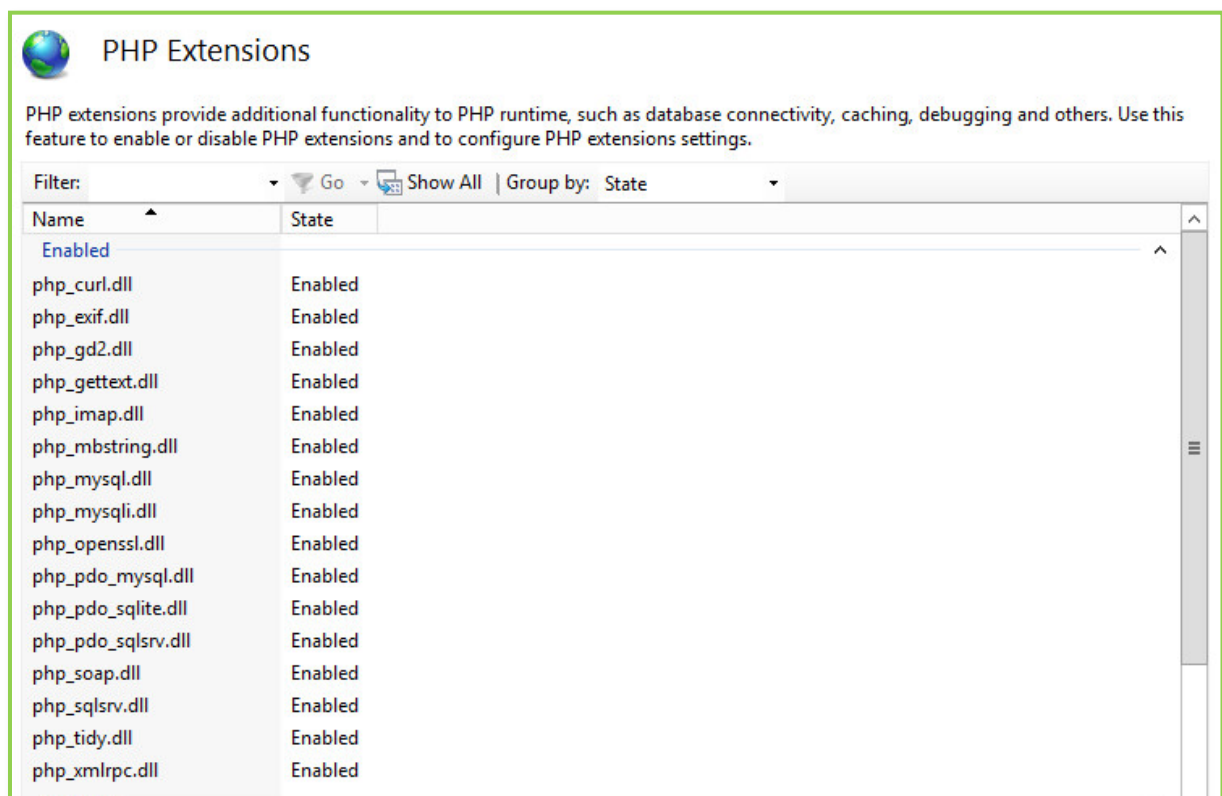


Abbildung 22: Webserver PHP Extensions

Standardmäßig ist „iisstart.html“ als Default Document gesetzt, welches beim Aufruf der Webseite als die Startseite erscheint. Diese wurde auf „login.php“ umgeändert, was die Login Seite als die Startseite nun anzeigt.

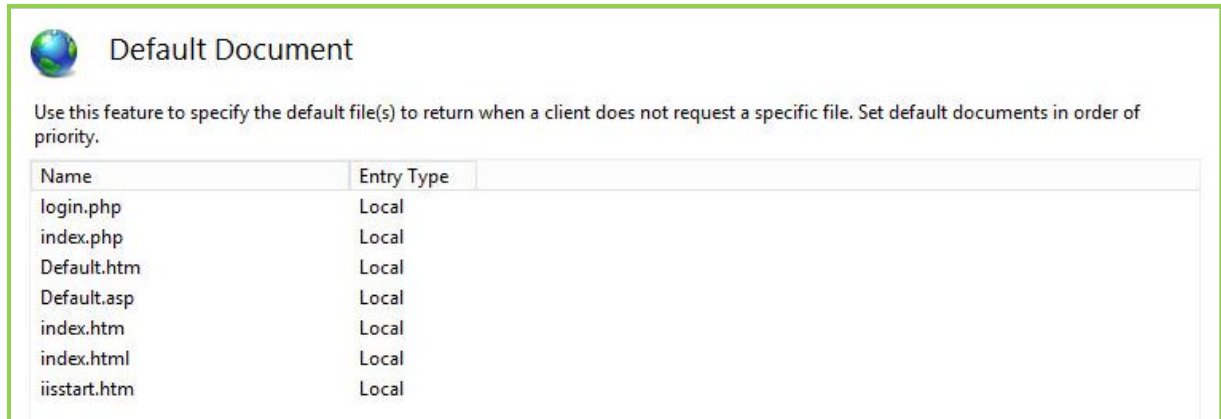


Abbildung 23: Webserver Default Document

5.4.2 File Transfer Protocol (FTP)

Um die Dateien nicht mühsam über andere Wege auf den Server zu transferieren, kam FTP zum Einsatz. Die Installation läuft über den Server Manager von Windows, was die Arbeit deutlich beschleunigt und vereinfacht.

Nach erfolgreicher Installation folgt die Konfiguration, welcher Ordner als Upload bzw. Download dienen soll. In unserem Fall ist es der Ordner „wwwroot“ welcher vom Webserver erstellt wurde.



Abbildung 24: EatFit FTP

6 Ergebnis

6.1 Komponentenübersicht

Folgende Übersicht zeigt die vier Komponenten von EatFit und worauf der Benutzer zugreift.

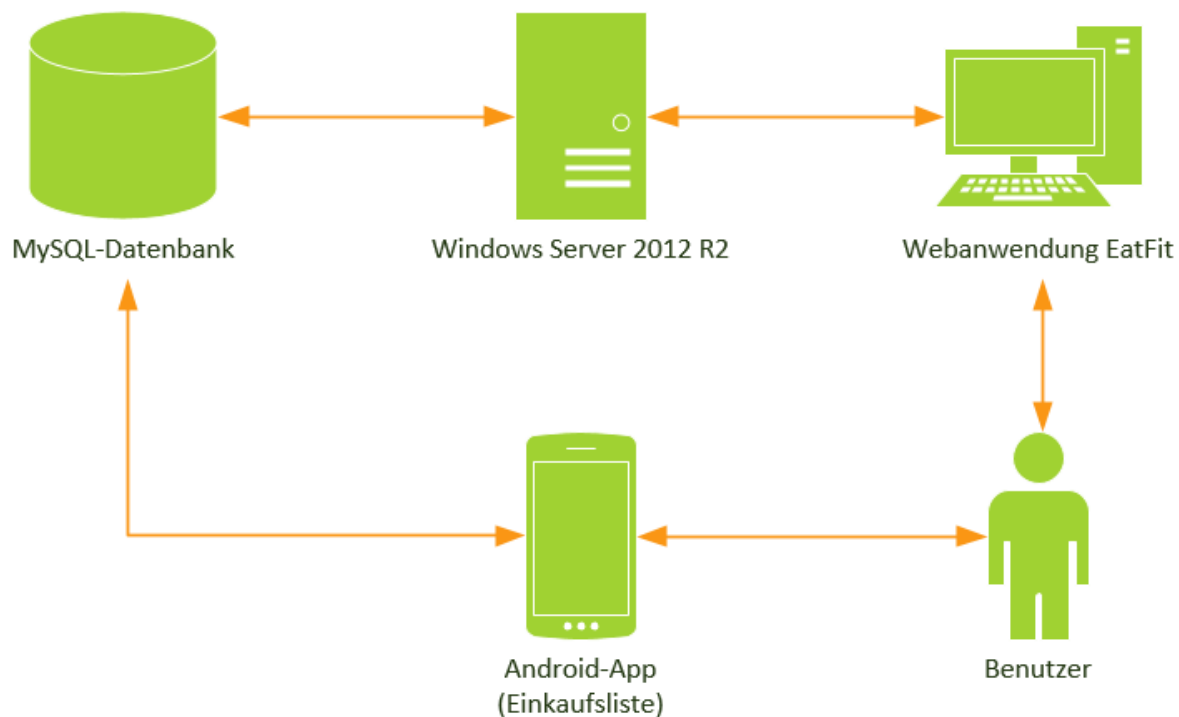


Abbildung 25: EatFit Komponentenübersicht

6.2 Beispiel Mahlzeitplanung mit EatFit

Dieses Beispiel entspricht der händisch gerechneten Mahlzeitplanung aus 2.2.4. Nun wird gezeigt, wie einfach die Planung einer Mahlzeit unter Berücksichtigung der Ziele mit EatFit verläuft.

Wir nehmen wieder an, aus einem Beratungsgespräch kennen wir folgende Zielwerte:

	Fett	Kohlenhydrate	Protein	Kalorien
Menge	59 g	235 g	147 g	2114,9 kcal

Tabelle 24: Beispiel Zielwerte

Diese Zielwerte tragen wir für unseren Benutzer ein, um später bei der Planung unsere Zielwerte immer im Auge zu haben.

Makronährstoffe			Mineralstoffe			Vitamine		
Name	Zielwert	Einheit	Name	Zielwert	Einheit	Name	Zielwert	Einheit
Brennwert	2115	kcal	Chlor	830	mg	Vitamin A	0,8	mg
Fett	59	g	Eisen	10	mg	Vitamin B1	1,3	mg
davon gesättigte Fettsäuren	0	g	Fluor	3,8	mg	Vitamin B2	1,8	mg
davon einfach ungesättigte Fettsäuren	0	g	Jod	200	µg	Vitamin B3	15	mg
davon mehrfach ungesättigte Fettsäuren	0	g	Kalium	2000	mg	Vitamin B5	8	mg
Kohlenhydrate	235	g	Kalzium	1000	mg	Vitamin B6	1,6	mg
davon Zucker	0	g	Kupfer	2	mg	Vitamin B7	0,25	µg
Protein	147	g	Magnesium	300	mg	Vitamin B9	0,16	µg
Ballaststoffe	30	g	Mangan	3	mg	Vitamin B12	3	µg
Salz	0	g	Molybdän	70	µg	Vitamin C	100	mg
Wasser	0	ml	Natrium	550	mg	Vitamin D	20	µg
Cholesterin	0	mg	Phosphor	700	mg	Vitamin E	10	mg
Broteinheiten	0	BE	Schwefel	0	mg			
Purin	0	mg	Selen	70	µg			
Alkohol	0	g	Zink	10	mg			

Abbildung 26: Beispiel EatFit Zielsetzung

Darüber hinaus haben wir auch noch Ziele für Mineralstoffe und Vitamine festgelegt.

Jetzt planen wir eine ähnliche Mahlzeit wie beim händisch gerechneten Beispiel. Die nötigen Nährwerte pro 100 g sind in der Datenbank gespeichert. In dieser Grafik sieht man die Seite zum Planen von Mahlzeiten. Links sind alle Mahlzeiten des entsprechenden Datums zu sehen. Rechts davon der Name der Mahlzeit und die genaue Uhrzeit kann auch angegeben werden. Ebenso sieht man pro Mahlzeit die Produkte inklusive geplanter Menge. Daneben kann man Produkte suchen und sieht bei Selektion die Makronährstoffe. Natürlich gibt es auch alle wichtigen Funktionen, wie Mahlzeiten und Produkte zu löschen oder zu ändern. Hat man sich für ein Produkt entschieden, gibt man die Menge ein und klickt auf "Hinzufügen".

Mahlzeit planen

Neue Mahlzeit erstellen

Datum: 2016-04-03

Mahlzeiten anzeigen

Mittag

Name: Mittag Datum: 2016-04-03 12:00:00

Mahlzeit Produkte			
Auswahl	Produkt	Menge	Einheit
<input type="checkbox"/>	Putenfleisch (Brust)	300	g
<input type="checkbox"/>	Reis, geschält	150	g
<input type="checkbox"/>	Brokkoli	200	g

Mahlzeit löschen

Produkt entfernen

Änderungen speichern

Produkt

Nährwerte		
Name	Wert	Einheit
Brennwert	26	kcal
Fett	0.2	g
Kohlenhydrate	2.51	g
davon Zucker	00.00	g
Protein	3.3	g

Produkt: Brokkoli

Menge:

Einheit: g

Hinzufügen

Abbildung 27: Beispiel EatFit Mahlzeit planen

Ganz rechts auf dieser Seite gibt es eine Tabelle, welche sich bei Änderungen sofort aktualisiert. Zur besseren Lesbarkeit wird diese hier nun gesondert abgebildet. Diese Tabelle ermöglicht genaues Planen, da man immer die Zielwerte im Auge hat. Änderungen in schon geplanten Mahlzeiten sind damit ein Kinderspiel, während es bei einer manuellen Berechnung zeitintensiv und demotivierend wäre, etwas nachträglich zu ändern.

Tagesbilanz Soll/Ist				
	Fett	KH	Protein	Kalorien
Soll	59g	235g	147g	2115kcal
Ist	4g	122g	89g	894kcal
Differenz	55g	113g	58g	1221kcal

Abbildung 28: Beispiel EatFit Tagesbilanz

In diesem Ausschnitt aus der Wochenübersicht, sehen wir unsere erfolgreich geplante Mahlzeit.

Wochenübersicht	
Von: 2016-04-03	Bis: 2016-04-09 Anzeigen
Sonntag	Montag
Mittag	
Putenfleisch (Brust) 300 g	
Reis, geschält 150 g	
Brokkoli 200 g	

Abbildung 29: Beispiel EatFit Wochenübersicht

Ein Blick auf die Einkaufsliste zeigt uns jetzt, was wir einkaufen müssen, um unseren Plan umsetzen zu können.

Einkaufsliste				
Von: 2016-04-03	Bis: 2016-04-09			
Anzeigen				Bestätigen
Produkt	Menge	Einheit	Gekauft	
Putenfleisch (Brust)	300	g	<input type="checkbox"/>	
Reis, geschält	150	g	<input type="checkbox"/>	
Brokkoli	200	g	<input type="checkbox"/>	

Abbildung 30: Beispiel EatFit Einkaufsliste

Damit steht zwischen unserem Ziel und uns dank EatFit jetzt nur noch Einkaufen, Kochen und Genießen, ohne stundenlanges Planen und Rechnen.

6.3 Webanwendung

6.3.1 Pages & Funktionen

6.3.1.1 Benutzerauthentifizierung

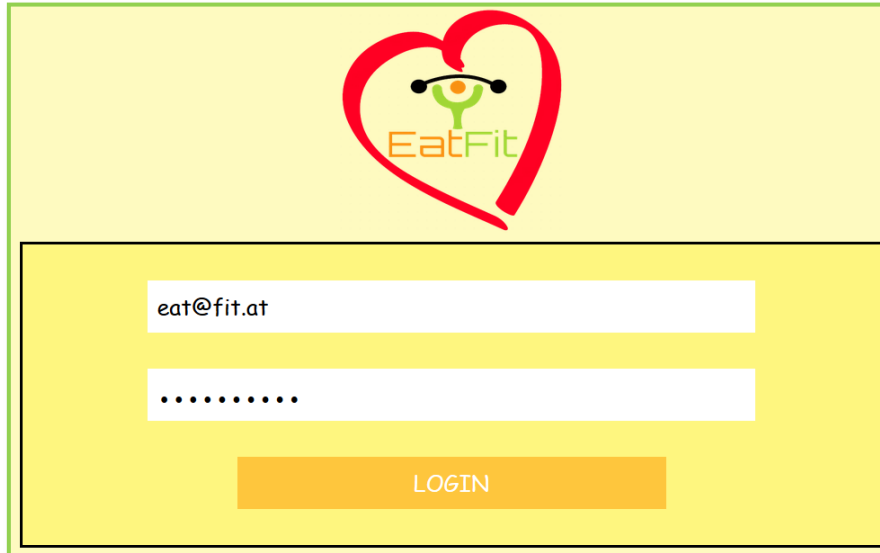


Abbildung 31: Login GUI

Funktion	Beschreibung
Einloggen	Login mit bestehendem Benutzer
Ausloggen	Logout mit eingeloggten Benutzer von der aktuellen Seite mittels Logout-Symbol

Tabelle 25: Funktionen Benutzerauthentifizierung

6.3.1.2 Wochenübersicht

Die Wochenübersicht zeigt alle geplanten Mahlzeiten von einem ausgewählten Zeitraum.

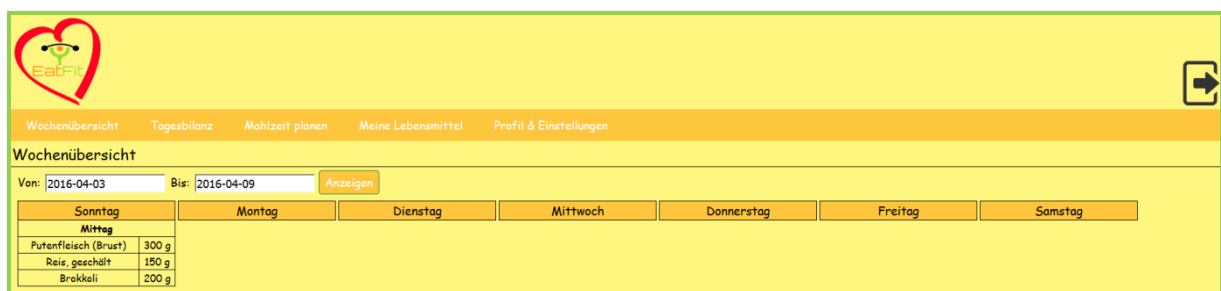


Abbildung 32: Wochenübersicht GUI

Funktion	Beschreibung
aktuelle Mahlzeiten anzeigen	Anzeigen der geplanten Mahlzeiten.

Tabelle 26: Funktionen Wochenübersicht

6.3.1.3 Tagesbilanz

In der Tagesbilanz sieht man für den ausgewählten Tag Ist- und Sollwerte aller Nährstoffe. Die Tabelle ähnelt jener aus der Zielsetzung, welche zuvor im Beispiel gezeigt wurde.

Funktion	Beschreibung
Tageswerte anzeigen	Werte zum entsprechenden Tag anzeigen

Tabelle 27: Funktionen Tagesbilanz

6.3.1.4 Mahlzeit planen

Wie die Mahlzeitplanung funktioniert wurde bereits im Beispiel gezeigt und erklärt. Hier sind alle Funktionen aufgelistet.

Funktion	Beschreibung
Produktsuche	Suche eines Produkts per Texteingabe
Produkt zur Mahlzeit hinzufügen	Produkt mit Startmenge zur ausgewählten Mahlzeit hinzufügen, Anzeigen der Makronährstoffe pro 100 g oder ml
Produktmenge in Mahlzeit ändern	Menge eines Produkts in Mahlzeit ändern
Produkt aus Mahlzeit entfernen	Entferne Produkt aus Mahlzeit (Produkt mit 0 g Rest soll nicht zurück bleiben)
neue Mahlzeit erstellen	Neue leere Mahlzeit erstellen
Mahlzeit umbenennen	Bezeichnung von Mahlzeit ändern
Mahlzeit löschen	Aktuell angezeigte Mahlzeit löschen
Mahlzeit speichern	Alle aktuellen Einträge zur Mahlzeit speichern
Mahlzeit wechseln	Mahlzeit von ausgewähltem Tag per Klick wechseln
aktuelle Tagesbilanz anzeigen	Tagesbilanz dynamisch aktualisieren

Tabelle 28: Funktionen Mahlzeit planen

6.3.1.5 Meine Lebensmittel - Vorrat

Im Vorrat kann man Produkte erfassen. Die manuelle Erfassung von Produkten macht speziell zu Beginn der Verwendung von EatFit Sinn, da man im Zuge einer Inventur alle brauchbaren Produkte von zu Hause aufnehmen kann, um sie sinnvoll zu verplanen.

Abbildung 33: Meine Lebensmittel - Vorrat GUI

Funktion	Beschreibung
Produktsuche	Suche eines Produkts per Texteingabe
Produkt zum Vorrat hinzufügen	Produkt mit Startmenge zum Vorrat hinzufügen
Produktmenge in Vorrat ändern	Menge eines Produkts in Vorrat ändern
Produkt aus Vorrat entfernen	Entferne Produkt aus Vorrat

Tabelle 29: Funktionen Meine Lebensmittel - Vorrat

6.3.1.6 Meine Lebensmittel - Verwaltung

In der Verwaltung hat der Benutzer die Möglichkeit die Produkte und ihre Nährwerte einzusehen, sowie zu verändern. Änderungen sind jedoch nur möglich, falls es sich um ein eigenes Produkt handelt. Ein eigenes Produkt kann bequem über "Produkte anlegen" erstellt werden.

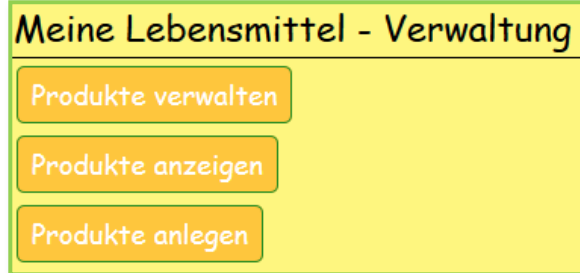


Abbildung 34: Meine Lebensmittel - Verwaltung Menüpunkte

Hier wird gerade unter "Produkte verwalten" das benutzerdefinierte Produkt "Bananen" angelegt. Der Benutzer kann selbst entscheiden, welche Quellen er wählt, um die Nährwerte für seine Produkte zu bestimmen.

Makronährstoffe			Mineralstoffe			Vitamine		
Name	Wert	Einheit	Name	Wert	Einheit	Name	Wert	Einheit
Brennwert	90	kcal	Chlor		mg	Vitamin A		mg
Fett		g	Eisen		mg	Vitamin B1		mg
davon gesättigte Fettsäuren		g	Fluor		mg	Vitamin B2		mg
davon einfach ungesättigte Fettsäuren		g	Jod		µg	Vitamin B3		mg
davon mehrfach ungesättigte Fettsäuren		g	Kalium		mg	Vitamin B5		mg
Kohlenhydrate	20	g	Kalzium		mg	Vitamin B6		mg
davon Zucker		g	Kupfer		mg	Vitamin B7		µg
Protein		g	Magnesium		mg	Vitamin B9		µg
Ballaststoffe		g	Mangan		mg	Vitamin B12		µg
Salz		g	Molybdän		µg	Vitamin C		mg
Wasser		ml	Natrium		mg	Vitamin D		µg
Cholesterin		mg	Phosphor		mg	Vitamin E		mg
Broteinheiten		BE	Schwefel		mg			
Purin		mg	Selen		µg			
Alkohol		g	Zink		mg			

Abbildung 35: Meine Lebensmittel - Verwaltung: Produkte verwalten

"Produkte anzeigen" und "Produkte anlegen" ähneln im Layout dieser Grafik. Der Unterschied liegt lediglich in den Funktionen.

Funktion	Beschreibung
Produktsuche	Suche eines Produkts per Texteingabe
Produkt zur Datenbank hinzufügen	Produkt mit leeren Parametern zur Datenbank hinzufügen
Produktparameter in Datenbank ändern	Parameter eines Produkts in Datenbank ändern (nur bei eigenen Produkten möglich)
Produkt aus Datenbank entfernen	Entferne Produkt aus Datenbank (nur bei eigenen Produkten möglich)
UPDATE: Produkt in Datenbank umbenennen	Bezeichnung von Produkt in Datenbank ändern (nur bei eigenen Produkten möglich)

Tabelle 30: Funktionen Meine Lebensmittel - Verwaltung

6.3.1.7 Meine Lebensmittel - Einkaufsliste

Die Einkaufsliste wurde bereits im Beispiel gezeigt. Hier sieht der Benutzer die einzukaufenden Produkte.

Funktion	Beschreibung
benötigte Produkte anzeigen	Einzukaufende Produkte anzeigen
eingekaufte Menge eingeben	eingekaufte Menge eingeben
eingekaufte Menge bestätigen	eingekaufte Menge bestätigen

Tabelle 31: Funktionen Meine Lebensmittel - Einkaufsliste

6.3.1.8 Profil

Im Reiter "Profil" kann der Benutzer sein Passwort ändern, sowie sein Profil deaktivieren.

Abbildung 36: Profil Passwort ändern

Funktion	Beschreibung
Passwort ändern	Passwort ändern
Profil deaktivieren	Profil deaktivieren

Tabelle 32: Funktionen Profil

6.3.1.9 Zielsetzung

In diesem Unterpunkt von "Profil & Einstellungen" kann der Anwender seine Zielwerte neu festlegen. Damit ist eine stetige Adaption an die Veränderungen des Körpers möglich.

Funktion	Beschreibung
UPDATE: Zielwerte ändern	Zielwerte ändern

Tabelle 33: Funktionen Zielsetzung

6.4 Android-Applikation

6.4.1 Login

Funktion	Beschreibung
Login	Login mit gültigen Daten

Tabelle 34: Funktionen App Login

Die Login-Page ermöglicht den Login mit gültigen Anmeldedaten. Nach erfolgreichem Login wird man zur Page für die persönlichen Einkaufslisten weitergeleitet.



Abbildung 37: App Login GUI

6.4.2 Einkaufsliste

Funktion	Beschreibung
Logout	Logout des angemeldeten Benutzers mit dem Button "Fertig"
Einkauf bestätigen	Der Button "Fertig" bestätigt den Einkauf und nimmt eine entsprechende Manipulation in der Datenbank vor. Außerdem wird der Logout damit durchgeführt. Nur Produkte, welche entsprechend unter "Status" als eingekauft selektiert wurden, werden auch als gekauft gespeichert.
Datum wählen	Es werden nach Wahl des Datums, alle benötigten Produkte dieser Woche unterhalb der Buttons aufgelistet.

Tabelle 35: Funktionen App Einkaufsliste



Abbildung 38: App Einkaufsliste GUI

7 Evaluierung

7.1 Herausforderungen & Lösungen

7.1.1 Schulalltag & Schulabschluss

Neben den regulären Leistungsfeststellungen im fünften Jahrgang, war nun auch die Diplomarbeit fertigzustellen. Darüber hinaus war auch das zweijährige Projekt, welches im Vorjahr begonnen wurde zu finalisieren. Außerdem ist wenige Wochen nach dem Abgabetermin der Diplomarbeit, die Standardisierte Reife- und Diplomprüfung, zu welcher nur angetreten werden darf, wenn der Jahrgang positiv abgeschlossen wurde.

7.1.2 Einhaltung des Zeitplans

All diese Dinge führten folglich zu Schwierigkeiten beim Einhalten des Zeitplans. Teilweise kam es auch vor, dass einfach mehr Zeit für Aufgaben benötigt wurde, da man sich noch einarbeiten musste oder einfach falsch kalkuliert wurde.

7.1.3 Lösung

In der ursprünglichen Zeitplanung war der Workload linear auf die Wochen verteilt. Somit konnte man bei Rückstand die Aufgaben einfach nach hinten verschieben, ohne das sich zu große Warteschlangen bilden. Darüber hinaus war in den Wochen vor der Diplomarbeitenabgabe ohnehin nichts mehr eingeplant, wodurch ein Zeitpuffer entstand. Außerdem brachten die Semester- und Osterferien zusätzlich Zeit ein, wodurch Rückstände aufgeholt werden konnten.

7.2 Persönliches Resümee

7.2.1 Erkol Beytullah

Bei dieser Diplomarbeit lernte ich meine Stärken und Schwächen besser kennen, sei es von der technischen oder sozialen Seite. In jedem dieser Bereiche sammelte ich Erfahrung für mein zukünftiges Leben und hoffe damit bevorstehende Arbeiten und Projekte erfolgreicher durchzuführen.

Im Zuge des Projektes vertiefte ich mich im Bereich der Webentwicklung und erweiterte mein Wissen dadurch. Meine Kenntnisse in den Servertechnologien waren zu Beginn sehr gering, durch die eigenständige Konfiguration des Projektserver, baute ich mir ein Grundwissen auf.

Es tauchten natürlich im Laufe des Projektes technische Probleme auf, die zur Verzögerung des Zeitplanes führten. Speziell bei der Webanwendung führte ein falscher Lösungsansatz dazu, dass jede Änderung der Webseite viel Zeit beanspruchte. Die Verwendung von AJAX-Abfragen, die ein partielles Neuerstellen ermöglichte, löste das Problem.

Was mir besonders an der Diplomarbeit gefallen hat, war das selbständige Arbeiten und selbst der Chef meiner Zeiteinteilung zu sein. Grundsätzlich bin ich mit meinen Leistungen an diesem Projekt zufrieden, natürlich kann man es verbessern und noch effizienter lösen. Jedoch wäre da ein größeres Projektteam nötig, denn das Thema Ernährung ist ein großes Thema.

Zu guter Letzt bedanke ich mich bei meinem Diplomarbeit Partner Oliver Voss für seine konstante Projektbeteiligung und bei unserer Betreuungslehrerin Frau Mag. Danner für Ihre souveräne Betreuung der Diplomarbeit.

7.2.2 Voß Oliver

In meiner Zeit an der HTL Perg war ich stets mit vielen Aufgaben, sowie größeren und kleineren Projekten konfrontiert. Die Diplomarbeit zählt auf jeden Fall zu den größeren Projekten und steht auch an der Spitze dieser Kategorie. Während ich bei allen anderen Projekten in größeren Teams gearbeitet habe, waren wir bei der Diplomarbeit zu zweit. Darüber hinaus war ich aufgrund der Idee für EatFit auch der Hauptverantwortliche bzw. Projektleiter. Außerdem gab von Beginn an ein klares Abgabedatum, welches einzuhalten war, denn die Diplomarbeit musste als Bestandteil der Reife- und Diplomprüfung fertig werden.

Speziell diese zwei Gegebenheiten gaben der Diplomarbeit einen strengen Beigeschmack und machten sie für mich zu einem prägenden und lehrreichen Teilabschnitt meiner HTL-Zeit. Ich habe wieder einmal mehr gelernt wie wichtig es ist jeden Tag etwas für meine Ziele zu tun, denn man wird ständig irgendwo auf ein Problem stoßen, egal was man sich vornimmt. Diese Probleme bzw. Fehler kosten teilweise viel Zeit, doch führen sie einen auf den richtigen Weg und optimieren die persönliche Vorgehensweise bei Aufgabenstellungen.

Während ich zu Beginn noch dachte, die Programmierung einer Webanwendung und Android-Applikation seien 08/15-Aufgaben, welche man einfach linear abarbeiten kann, weiß ich nun, welche Tücken einen erwarten können, speziell wenn man sich in manche Themen erst einarbeiten muss.

Zum Schluss möchte ich mich nochmal ausdrücklich bei meinem Teamkollegen Beytullah Erkol für sein Engagement bedanken. Spezieller Dank auch an Frau Mag. Danner für Ihre Unterstützung.

8 Verzeichnisse

8.1 Abbildungsverzeichnis

Abbildung 1: Projektteam	15
Abbildung 2: MyFitnessPal Angabe Portionsgröße	18
Abbildung 3: EatFit Angabe Portionsgröße	18
Abbildung 4: HTML 5 Logo	27
Abbildung 5: CSS3 Logo	27
Abbildung 6: php Logo.....	27
Abbildung 7: MySQL Logo	28
Abbildung 8: JavaScript Logo.....	28
Abbildung 9: jQuery Logo.....	29
Abbildung 10: Windows Server Logo	29
Abbildung 11: Android Logo.....	29
Abbildung 12: phpMyAdmin Logo.....	30
Abbildung 13: Eclipse Logo.....	30
Abbildung 14: Firebug Logo.....	30
Abbildung 15: Google Drive Logo	30
Abbildung 16: VMware Logo	31
Abbildung 17: Android Studio Logo.....	31
Abbildung 18: FileZilla Logo.....	32
Abbildung 19: GUI Planung	33
Abbildung 20: GUI umgesetzt.....	33
Abbildung 21: App Planung	42
Abbildung 22: Webserver PHP Extensions	50
Abbildung 23: Webserver Default Document	51
Abbildung 24: EatFit FTP	51
Abbildung 25: EatFit Komponentenübersicht	53
Abbildung 26: Beispiel EatFit Zielsetzung.....	54
Abbildung 27: Beispiel EatFit Mahlzeit planen.....	55
Abbildung 28: Beispiel EatFit Tagesbilanz	55
Abbildung 29: Beispiel EatFit Wochenübersicht	56
Abbildung 30: Beispiel EatFit Einkaufsliste.....	56
Abbildung 31: Login GUI.....	57
Abbildung 32: Wochenübersicht GUI.....	57
Abbildung 33: Meine Lebensmittel - Vorrat GUI.....	59
Abbildung 35: Meine Lebensmittel - Verwaltung: Produkte verwalten	60
Abbildung 34: Meine Lebensmittel - Verwaltung Menüpunkte.....	60
Abbildung 36: Profil Passwort ändern.....	61
Abbildung 37: App Login GUI	63
Abbildung 38: App Einkaufsliste GUI.....	64
Abbildung 39: Projektstrukturplan.....	74
Abbildung 40: EatFit Meilensteine	75
Abbildung 41: tatsächliche Arbeitsaufteilung in Prozent.....	79

8.2 Tabellenverzeichnis

Tabelle 1: Personendaten Projektteam.....	15
Tabelle 2: Kriterien EatFit vs. Konkurrenz	17
Tabelle 3: Vergleich: Angabe einer Portionsgröße.....	18
Tabelle 4: Vergleich: Anzahl verfügbarer Nährstoffe	18
Tabelle 5: Vergleich: Zugriff eigener Lebensmitteleinträge	18
Tabelle 6: Funktionsvergleich EatFit vs. Konkurrenz.....	18
Tabelle 7: Makronährstoffe Kaloriendichte	21
Tabelle 8: Nährstoffübersicht.....	22
Tabelle 9: Beispiel Zielwerte.....	23
Tabelle 10: Beispiel Nährwerte pro 100 g.....	23
Tabelle 11: Beispiel Zielwerte & Differenz	24
Tabelle 12: Tabelle Benutzer	46
Tabelle 13: Tabelle Benutzerziel.....	46
Tabelle 14: Tabelle Parametertyp	46
Tabelle 15: Tabelle Parameter	46
Tabelle 16: Tabelle Produkt.....	47
Tabelle 17: Tabelle Vorrat	47
Tabelle 18: Tabelle Mahlzeit	47
Tabelle 19: Tabelle Einkaufsliste	47
Tabelle 20: Tabelle Parameter_has_Benutzerziel.....	48
Tabelle 21: Tabelle Parameter_has_Produkt.....	48
Tabelle 22: Tabelle Mahlzeit_has_Produkt	48
Tabelle 23: Tabelle Einkaufsliste_has_Produkt.....	49
Tabelle 24: Beispiel Zielwerte.....	54
Tabelle 25: Funktionen Benutzerauthentifizierung.....	57
Tabelle 26: Funktionen Wochenübersicht	57
Tabelle 27: Funktionen Tagesbilanz	58
Tabelle 28: Funktionen Mahlzeit planen.....	58
Tabelle 29: Funktionen Meine Lebensmittel - Vorrat	59
Tabelle 30: Funktionen Meine Lebensmittel - Verwaltung.....	60
Tabelle 31: Funktionen Meine Lebensmittel - Einkaufsliste	61
Tabelle 32: Funktionen Profil	61
Tabelle 33: Funktionen Zielsetzung.....	62
Tabelle 34: Funktionen App Login.....	63
Tabelle 35: Funktionen App Einkaufsliste	64
Tabelle 36: geplante Arbeitsaufteilung	79
Tabelle 37: Leistungsumfang Technologien	80
Tabelle 38: Leistungsumfang Applikationen	80
Tabelle 39: Leistungsumfang Geschäftsprozesse.....	82

8.3 Literaturverzeichnis

- **Ernährungstechnische Zahlenwerte**
 - http://www.tabelle.info/kalorien_9.html

- **Technologien Theorie**
 - MySQL 5.1 Referenzhandbuch, Kapitel 20

 - de.wikipedia.org/wiki/Document_Object_Model

 - Hannes Pfeiffer: Ajax: Modernes Webscripting. Data Becker, August 2006

 - Walter S. Mossberg: Google Stores, Syncs, Edits in the Cloud, The Wall Street Journal. 25. April 2012

 - Bertram Wöhrmann, Carsten Schäfer, Günter Baumgart, Urs Stephan Alder, Marcel Brunner: VMware vSphere 6: Das umfassende Handbuch. Rheinwerk-Verlag, Reihe Computing

9 Anhang

9.1 Projektdokumentation

9.1.1 Projektstrukturplan

EatFit besteht aus den folgenden vier Komponenten:

- Webanwendung
- Applikation
- Server
- Datenbank

In der Hierarchie auf der folgenden Seite wird speziell auf die Komponenten Webanwendung und Applikation eingegangen. Pro Komponente werden die grafischen Oberflächen und deren Funktionen aufgelistet. Dabei werden nur die wichtigsten Funktionen genannt.

In der ersten Ebene sieht man das Softwareprodukt, in der zweiten Ebene die entsprechenden Komponenten, die dritte zeigt die grafischen Oberflächen und die letzte deren wichtigsten Funktionen.

Eine Auflistung aller Funktionen ist unter Punkt 6.3.1 und 10.3 zu finden.

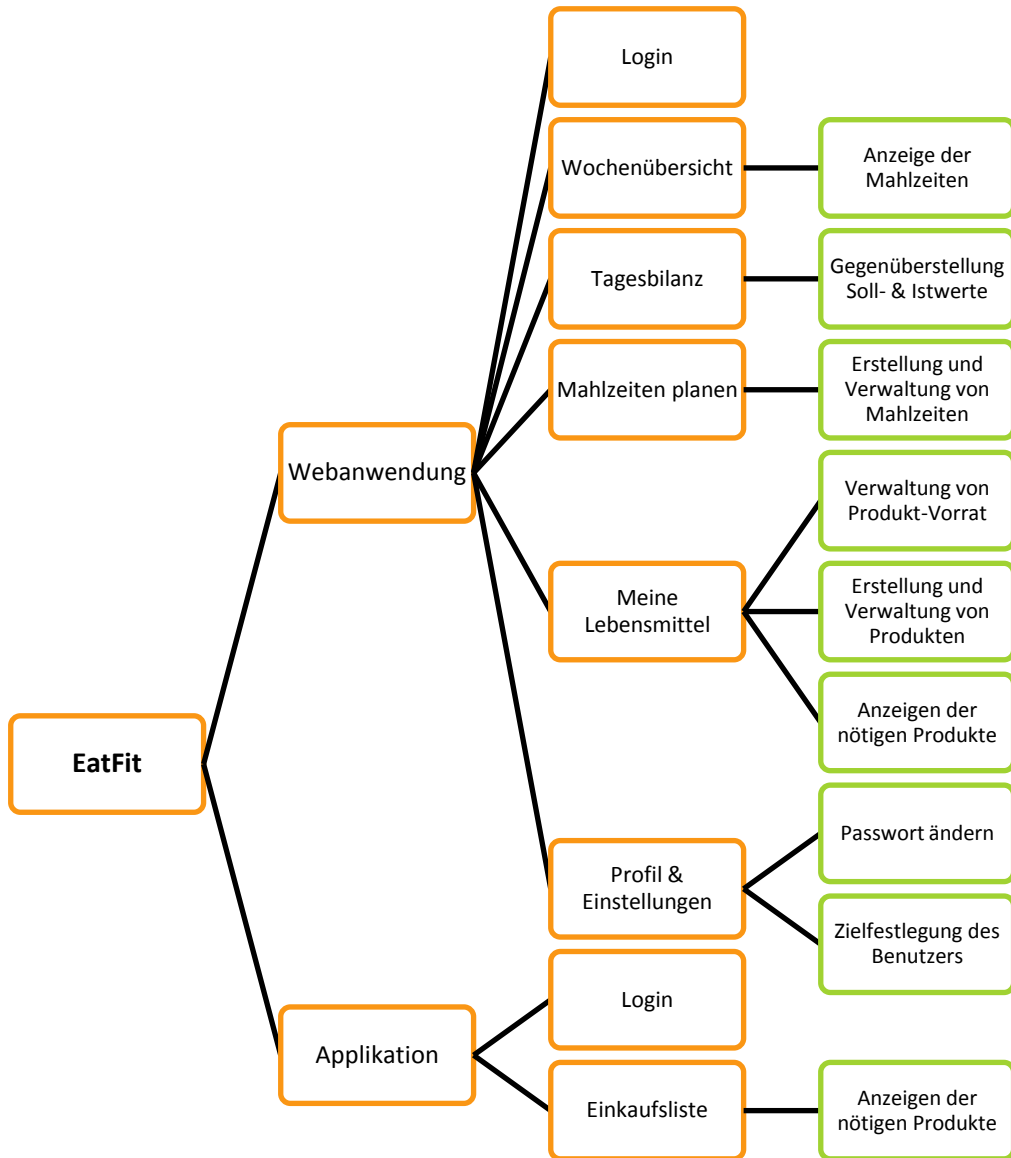


Abbildung 39: Projektstrukturplan

9.1.2 Meilensteine

EatFit Meilensteine	16.06.2015: Start Diplomarbeit
	02.07.2015: Server: Arbeitsumgebung eingerichtet
	27.09.2015: Webanwendung GUI-Planung fertigstellen
	21.10.2015: Datenmodell fertigstellen
	17.03.2016: Fertigstellung Webanwendung
	20.03.2016: Fertigstellung Android-Applikation
	08.04.2016: Abgabe Diplomarbeit

Abbildung 40: EatFit Meilensteine

9.1.3 Zeitplan

Woche 1 (25.01.2016 – 31.01.2016)

- Profil & Einstellungen (Profil) Funktionalitäten
 - Passwort ändern
 - Benutzerprofil deaktivieren
 - Nachfrage ob der Benutzer, wirklich sein Profil endgültig löschen möchte (Dialogfenster)
- Meine Lebensmittel (Verwaltung) Funktionalitäten
 - Produktsuche dem Benutzer ermöglichen
 - Benutzer gibt „Beeren“ ein → Produktvorschläge die Zeichenkette „Beeren“ enthalten
 - Erdbeeren
 - Heidelbeeren
 - ...
 - Die Vorschläge werden in einer Liste angezeigt

Woche 2 (01.02.2016 – 07.02.2016)

- Meine Lebensmittel (Verwaltung) Funktionalitäten
 - Produkt kann vom Benutzer gelöscht werden
 - Voraussetzung → eigenes Produkt
 - Eigene Produkte können angelegt werden
 - Eingabe der Produkt Nährstoffe
 - Auswahl eines Produktes
 - Anzeige der Nährwerte des Produktes
 - Falls eigenes Produkt → Änderung der Produkt-Nährwerte ermöglichen
 - Korrekte Eingabe überprüfen
- Meine Lebensmittel (Vorrat) Funktionalitäten
 - Produkte zu meinem Vorrat hinzufügen → Eingabe der Menge usw.
 - Für manuelles einfügen von Produkten benötigt
 - Speicherung des manuell erstellten Produktliste
- Profil & Einstellungen (Zielsetzung) Funktionalitäten
 - Benutzer-Zielsetzung
 - Eingabe der Parameter in der Datenbank (Nährwert-Buch vorhanden)
 - Anzeige der Parameter in Gruppierungen (Makronährstoffe, Vitamine, ...) in Tabellenform
 - Eingabe und Speicherung der Benutzerziele
 - Prüfung ob erlaubte Werte eingegeben wurden (Negative Zahlen, ...)

Woche 3 (08.02.2016 – 14.02.2016)

- Mahlzeit planen Funktionalitäten
 - Wechseln zwischen Wochentagen ermöglichen, zuerst Auswahl eines Datums, sonst keine Woche definiert
 - Hinzufügen des ausgewählten Produkt zu einer Mahlzeit
 - Mehrere Produkte können zu einer Mahlzeit hinzugefügt werden
 - Speichern der Mahlzeitname (Frühstück, Snack, ...)
 - Automatisches aktualisieren der Mahlzeit-Produkttable beim Einfügen eines Produktes
 - Speicherung der Mahlzeit
 - Löschen der Mahlzeit
 - Erstellen einer neuen Mahlzeit
- App Funktionalitäten
 - Verbindung zum Server
 - Erstellung der App GUI (Codierung)

Woche 4 (15.02.2016 – 21.02.2016)

- Wochenübersicht Funktionalitäten
 - Anzeige der Mahlzeiten zum richtigen Wochentag
 - Drag & Drop oder Pfeilnavigation (nach oben bzw. unten) verschieben von Mahlzeiten bei einem Wochentag
 - Nicht zwischen verschiedenen Wochentagen möglich
 - Verlinkung auf Mahlzeit löschen/verändern/bearbeiten (Mahlzeit planen)
- App Funktionalitäten
 - Benutzer Login ermöglichen
- Mahlzeit planen Funktionalitäten
 - Automatische Aktualisierung der Tagesbilanz Tabelle Soll/Ist
 - Nach einfügen eines Produktes
 - Bestand Prüfung mit dem aktuellen Vorrat
 - Falls Produkt nicht vorhanden und der Benutzer die Mahlzeit trotzdem speichert → Einfügen in die Einkaufsliste
 - Navigation zwischen Mahlzeiten eines Wochentages ermöglichen

Woche 5 (22.02.2016 – 28.02.2016)

- Mahlzeit planen Funktionalitäten
 - Kopieren einer Mahlzeit
 - Falls der Benutzer die Mahlzeit öfters verwenden will
- Tagesbilanz Funktionalitäten
 - Anzeige der Nährstoffwerte in Gruppierungen
 - Vergleich mit Benutzerziel und Tatsächlichem Wert

Woche 6 (29.02.2016 – 06.03.2016)

- Meine Lebensmittel (Einkaufsliste) Funktionalitäten
 - Auswahl eines Datum → Wahl der richtigen Einkaufsliste bzw. richtige Woche
 - Anzeige Einkaufsliste der ausgewählten Woche

Woche 7 (07.03.2016 – 13.03.2016)

- App Funktionalitäten
 - Anzeige und Bestätigung der Einkaufsliste
 - Ausloggen des Benutzers
- Webanwendung & App
 - Bugfixing
 - Testphase durch Projektmitglieder bzw. Projektbetreuerin
 - Verbesserungen

Woche 8 (14.03.2016 – 20.03.2016)

- Diplomarbeit Buch
 - Prüfung auf Rechtschreibfehler
 - Richtige Formatierung (Überschriften usw.)
 - Achtung beim Zitieren usw.

Woche 9 (21-03-2016 – 27.03.2016)

- Diplomarbeit Buch
 - Abgabe bei Buchdruck
 - Übertragung auf Digitale Medien (CD)

Woche 10 (28.03.2016 – 01.04.2016)

- Zeitpuffer

9.1.4 Arbeitsaufteilung

Ursprünglich war folgende Arbeitsaufteilung der Subthemen vorgesehen:

Erkol Beytullah	Voß Oliver
Serverimplementierung & Datenbank	Ernährungstechnischer Hintergrund
Visualisierung (GUI-Planung)	
Datenmodellierung	
Webanwendung umsetzen	Applikation umsetzen
Diplomschrift verfassen	

Tabelle 36: geplante Arbeitsaufteilung

Das folgende Diagramm zeigt in Prozent wie sehr das jeweilige Teammitglied zur Fertigstellung bzw. Umsetzung des Teilbereichs dann tatsächlich beigetragen hat.

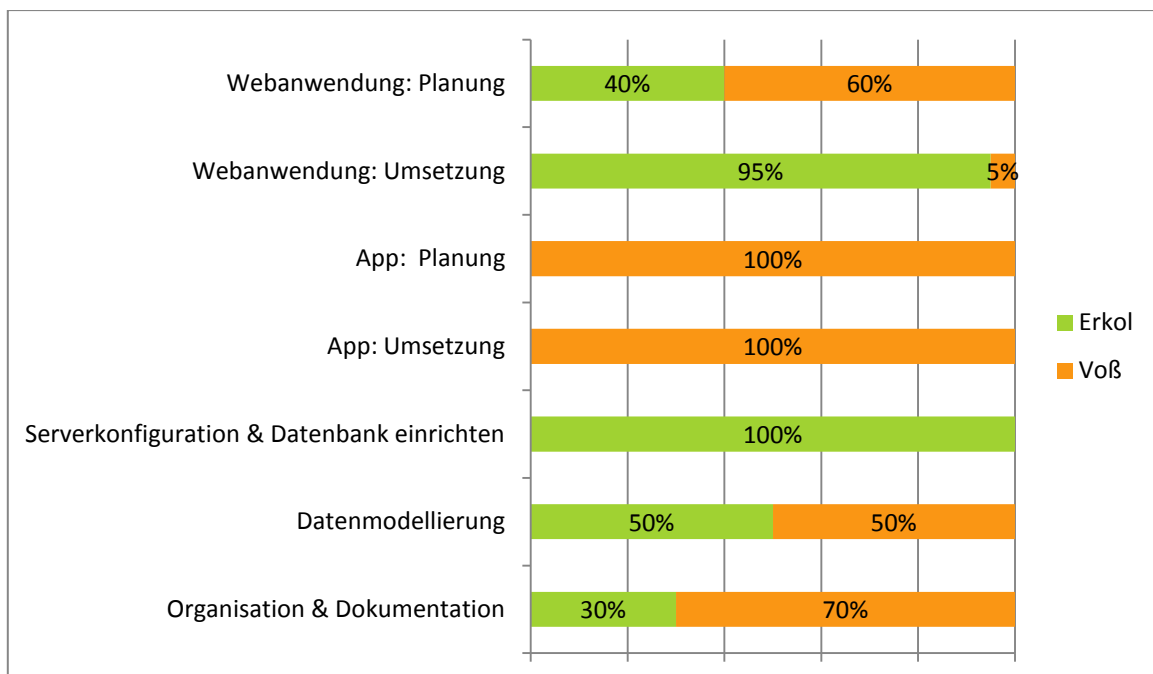


Abbildung 41: tatsächliche Arbeitsaufteilung in Prozent

9.1.5 Kommunikation

9.1.5.1 Teamintern

Innerhalb des Projektteams erfolgte die Kommunikation primär über WhatsApp oder persönlich in der Schule. Regelmäßig wurden auch Telefonate über Skype geführt.

9.1.5.2 Projektteam & Projektbetreuung

Die Kommunikation zwischen Projektteam und Projektbetreuung erfolgte per E-Mail oder persönlich im Gespräch. Projektbesprechungen wurden außerhalb der Schulzeit im Schulgebäude geführt.

10 Leistungsumfang

10.1 Leistungsumfang hinsichtlich der Technologien

In Scope	Out Of Scope
Webanwendung	
HTML5, CSS JavaScript PHP SQL	andere Auszeichnungssprachen, Stylesheet-Sprachen, Skriptsprachen, Datenbanksprachen
Applikation	
Android	andere mobile Betriebssysteme (z.B. iOS, Windows, BlackBerry OS)
Server	
Windows Server 2012 R2	andere Servertechnologien
Datenbank	
MySQL (Administration: phpMyAdmin)	andere Datenbanken

Tabelle 37: Leistungsumfang Technologien

10.2 Leistungsumfang hinsichtlich der Applikationen

In Scope	Out Of Scope
Webanwendung	
<ul style="list-style-type: none"> • Benutzerauthentifizierung • Login & Logout • GUI - CRUD • Wochenübersicht • Tagesbilanz • Mahlzeit planen • Lebensmittel - Vorrat • Lebensmittel - Verwaltung • Lebensmittel - Einkaufsliste • Profil (Passwort ändern) • Zielsetzung • Wochenplan • Einkaufsliste • Schnittstelle zur Applikation 	Schnittstelle zu anderer Anwendung, außer dazugehöriger Applikation Lebensmittel: Haltbarkeitsparameter Lebensmittel/Einkaufsliste: Preis, also keine Budgetverwaltung
Applikation	
<ul style="list-style-type: none"> • Benutzerauthentifizierung • Login & Logout • GUI - CRUD • Einkaufsliste • Schnittstelle zur Webanwendung 	Schnittstelle zu anderer Anwendung, außer dazugehöriger Webanwendung CRUD-Funktionen der Webanwendung, außer Einkaufsliste

Tabelle 38: Leistungsumfang Applikationen

10.3 Leistungsumfang hinsichtlich der Geschäftsprozesse

In Scope	Out Of Scope
Webanwendung	
<ul style="list-style-type: none"> • Benutzerauthentifizierung <ul style="list-style-type: none"> ○ Login ○ Logout 	standardmäßige Benutzerregistrierung (incl. Bestätigungsemail etc.)
<ul style="list-style-type: none"> • Wochenübersicht <ul style="list-style-type: none"> ○ aktuelle Mahlzeiten anzeigen 	
<ul style="list-style-type: none"> • Tagesbilanz <ul style="list-style-type: none"> ○ aktuelle Werte anzeigen 	
<ul style="list-style-type: none"> • Mahlzeit planen <ul style="list-style-type: none"> ○ Mahlzeit bearbeiten ○ Produkt suchen ○ Produkt zur Mahlzeit hinzufügen ○ Produktmenge ändern ○ Produkt entfernen ○ Mahlzeit erstellen ○ Mahlzeit (um-)benennen ○ Mahlzeit löschen ○ Mahlzeit speichern ○ zwischen Mahlzeiten wechseln ○ aktuelle Tagesbilanz anzeigen 	automatische Mahlzeitvorschläge von EatFit
<ul style="list-style-type: none"> • Meine Lebensmittel - Vorrat <ul style="list-style-type: none"> ○ Produktsuche ○ Produkt zu Vorrat hinzufügen ○ Produktmenge ändern ○ Produkt entfernen ○ Vorratsänderungen speichern 	
<ul style="list-style-type: none"> • Meine Lebensmittel - Verwaltung <ul style="list-style-type: none"> ○ Produkt suchen ○ Produkt erstellen incl. (um-)benennen ○ Produkt ändern ○ Produkt löschen 	CRUD eigener Parameter
<ul style="list-style-type: none"> • Meine Lebensmittel - Einkaufsliste <ul style="list-style-type: none"> • anzeigen • eingekaufte Menge eingeben • eingekaufte Menge bestätigen 	DELETE: Produkte aus der Liste entfernen CREATE: Produkte zur Liste hinzufügen
<ul style="list-style-type: none"> • Profil <ul style="list-style-type: none"> ○ Passwort ändern ○ Profil deaktivieren 	
<ul style="list-style-type: none"> • Zielsetzung <ul style="list-style-type: none"> ○ Zielsetzung ändern 	vorgeschlagene Zielwerte von EatFit, errechnet durch Eingabe bestimmter Parameter

Applikation	
<ul style="list-style-type: none">• Benutzerauthentifizierung<ul style="list-style-type: none">○ Login○ Logout	standardmäßige Benutzerregistrierung (incl. Bestätigungsemail etc.)
<ul style="list-style-type: none">• wöchentliche Einkaufsliste<ul style="list-style-type: none">○ anzeigen / synchronisieren○ Datum wählen○ eingekaufte Menge eingeben○ eingekaufte Menge bestätigen	DELETE: Produkte aus der Liste entfernen CREATE: Produkte zur Liste hinzufügen

Tabelle 39: Leistungsumfang Geschäftsprozesse