



# Studienarbeit

food4life - der Kalorientracker

Erstellt von:

Sven Kuczera

Robert-Frese-Straße 31

58332 Schwelm

Prüfer:

Prof. Dr. Seifert

Eingereicht am:

13.4.2018

# Inhaltsverzeichnis

<b>1</b>	<b>Das Team</b>	<b>2</b>
1.1	Das food4life Team . . . . .	2
<b>2</b>	<b>Ziel des Projekts</b>	<b>3</b>
<b>3</b>	<b>Benutzerhandbuch</b>	<b>4</b>
<b>4</b>	<b>Projektplanung</b>	<b>7</b>
4.1	Pflichtenheft . . . . .	7
4.2	Projektablaufplan . . . . .	8
4.3	Planung der Software . . . . .	8
4.3.1	Planung der Mockups . . . . .	8
4.3.2	Planung der Datenstrukturen und Schnittstellen . . . . .	9
4.3.3	Planung der Activities und Layouts . . . . .	9
4.3.4	Planung der Navigation und des Datenaustausches zwischen den Activities . . . . .	9
4.4	Geplante Aufgabenverteilung im Team . . . . .	9
<b>5</b>	<b>Beschreibung des Projektverlaufs</b>	<b>10</b>
5.1	Tatsächliche Aufgabenverteilung im Team . . . . .	10
5.2	Teammeetingprotokolle . . . . .	12
5.3	Projekttagebücher aller Teammitglieder . . . . .	14
5.4	Beschreibung von Problemen . . . . .	14
<b>6</b>	<b>Dokumentation der Software</b>	<b>16</b>
6.1	Dokumentation der Paketstruktur des Android-Projektes . . . . .	16
6.2	Dokumentation der Activities . . . . .	16
6.2.1	Activity 1 . . . . .	16
6.2.2	Activity 2 . . . . .	17
6.2.3	Activity 3 . . . . .	19
6.2.4	Activity 4 . . . . .	20
6.2.5	Activity 5 . . . . .	22
6.2.6	Activity 6 . . . . .	24
6.2.7	Activity 7 . . . . .	25
6.2.8	Activity 8 . . . . .	26

6.2.9	Activity 9 . . . . .	28
6.2.10	Activity 10 . . . . .	30
6.2.11	Activity 11 . . . . .	30
6.2.12	Activity 12 . . . . .	31
6.2.13	Activity 13 . . . . .	32
6.2.14	Activity 14 . . . . .	34
6.2.15	Activity 15 . . . . .	36
6.3	Dokumentation der Navigation und des Datenaustauschs zwischen Activities . . . . .	38
6.4	Dokumentation der Activity-übergreifenden, persistenten Datenhaltung	38
6.5	Dokumentation der Activity-übergreifenden Klassen . . . . .	38
<b>7</b>	<b>Fazits aller Teammitglieder</b>	<b>39</b>
7.1	Jannik Maes . . . . .	39
7.2	Sven Kuczera . . . . .	39
7.3	Henryk Schaffrath . . . . .	40
7.4	Benedikt Burczek . . . . .	41
<b>8</b>	<b>Anhang</b>	<b>42</b>
	<b>Anhang</b>	<b>42</b>
8.1	Anhang mit allen Quelltexten . . . . .	42
8.2	Anhang mit einem Verzeichnis der verwendeten Tools und Hilfsprogramme	42
	<b>Quellenverzeichnis</b>	<b>43</b>
	<b>Ehrenwörtliche Erklärung</b>	<b>44</b>

# 1 Das Team

## 1.1 Das food4life Team



(von links) Sven Kuczera, Jannik Maes, Henryk Schaffrath, Benedikt Burczek

## 2 Ziel des Projekts

Das Ziel des Projektes *food4life* ist es ein neuartiges Kalorientagebuch als Android Applikation zu entwickeln. Die App hilft dem Benutzer die konsumierten Lebensmittel und deren enthaltende Kalorien in einem Tagebuch zu erfassen. Außerdem wird dem Benutzer ein auf ihn persönlich zugeschnittenes Tagespensum angezeigt. Mit der einfacheren und übersichtlichen Oberfläche und dem äußerst angenehmen Layout und dem intuitiven Interaktionsverhalten bietet die App von *food4Life* eine überraschend andere Erfahrung bei der Verwendung eines „Calorie Trackers“. Zur Vereinfachung der Benutzung ist es außerdem möglich verschiedene Lebensmittel zu einem Menü hinzuzufügen und diese dann mit einem Klick in das Tagebuch einzutragen. Die App bietet zudem die Möglichkeit der statistischen Auswertung des durchschnittlichen Kalorienverbrauchs der letzten 7, 14 und 30 Tage.

## 3 Benutzerhandbuch

### Installation der App

Unter Android kann die App über ein an einen, mit Android Studio ausgestatteten, PC/Mac angeschlossenes Gerät (mind. API 15) installiert werden. Dazu muss das Projekt gestartet werden und bei dem Gerät das USB-Debugging aktiviert sein. Mit dem oben gelegenen „Play“ Button kann die App nun auf dem Gerät installiert werden.



Die App wird nach der Installation automatisch auf dem Gerät gestartet.

### Startseite

Die Startseite ist übersichtlich aufgebaut und in drei große Bereiche eingeteilt. Im ersten Teil (von oben) befindet sich die Übersicht. Hier wird neben dem aktuellen Datum in der Mitte, links die eingenommenen Kalorien angezeigt. Diese werden aus dem aktuellen Tag zusammengerechnet angezeigt. Auf der rechten Seite befindet sich die für diesen Tag verbleibende Kalorienzahl, welcher ebenfalls berechnet wird. Zur optisch angenehmeren Darstellung und zur Erleichterung der Benutzung der App wird ebenfalls Fortschrittsbalken angezeigt welcher mit weiterer Kalorienzunahme voranschreitet.

Im zweiten und größten Feld spielt sich der Hauptteil der Aktivität ab, hier sind die vier verschiedenen Mahlzeiten zu sehen, „Frühstück“, „Mittagessen“, „Abendessen“ und „Snacks“. Mit den roten Plus Button lässt sich einfach ein Lebensmittel/Menü zur jeweiligen Mahlzeit hinzufügen. Es öffnet sich ein neues Fenster Hinzufügen. Das Datum ist hier ebenfalls wieder zu finden. In den folgenden beiden Spinnern kann man außerdem neu auswählen zu welcher Mahlzeit man das Lebensmittel/Menü hinzufügen möchte, um einen Fehler schneller zu korrigieren und das Lebensmittel/Menü welches in einem Auswahlmenü zur Verfügung steht. Anschließend muss nur noch die verzehrte Menge eingetragen werden. Die Anzahl an Kalorien wird unten automatisch angezeigt. Ist alles fertig eingetragen klickt man auf den Button „Hinzufügen“. Man gelangt nun automatisch zurück zu Startseite wo sich das Produkt nun in der passenden Ansicht der Mahlzeit befindet. Der Fortschrittsbalken, sowie beide Kalorienzahlen haben sich in der Kopfleiste ebenfalls angepasst.

Der letzte untere Teil besteht aus vier Buttons welche einfach zu erkennen und zu handhaben sind. Der erste Button öffnet die Kalenderansicht. Der zweite und dritte Button öffnet die Fenster Lebensmittel und Menü um diese jeweils hinzuzufügen, zu verändern

oder zu entfernen. Mit dem letzten Button gelangt man zu der Profilansicht.

### **Kalenderansicht**

Der Kalender startet automatisch mit dem aktuellen Datum und ist in zwei Oberflächen eingeteilt. Oben befindet sich eine Kalenderansicht, durch welche durch einfache bekannte Tipp/Wischbewegungen navigiert werden kann. In der unteren Hälfte befindet sich die bereits aus der Startseite bekannte Mahlzeiten Ansicht. Diese ist dem App Design zugrunde gleich der Startseite aufgebaut. So findet man sich sofort in der gesamten App zurecht.

Mit der Auswahl eines Datums wird sofort die Mahlzeiten Ansicht angepasst und mit dem ausgewählten Datum angepasst. Man sieht nun die verzehrten Lebensmittel/Menüs von diesem Tag. Auf diese Möglichkeit lassen sich auch nachträglich Lebensmittel für einen bestimmten Tag eintragen, falls beispielsweise ein Nahrungsmittel vergessen wurde. Alternativ lassen sich auch eingetragene Lebensmittel entfernen oder anpassen, sowohl in der Menge als auch in der Position. Dadurch das sich die Ansicht Oberfläche dynamisch an die Menge der enthaltenen Lebensmittel anpasst ist ein angenehmes Scrollen in der App möglich. Der rote Button führt auch hier wieder zum Öffnen des Fensters Hinzufügen, welches bereits von der Startseite bekannt ist.

### **Lebensmittel**

Dieses Fenster ähnelt dem der Menüs und ist einfach und übersichtlich gestaltet. Die App kommt nach einer frischen Installation mit fünf Basis Lebensmitteln. Banane, Milch, Pizza, Reis und Schokolade. Diese sind in unterschiedlichen Mengen und Einheiten angegeben. Banane in Stück, Milch in l und Reis in Gramm beispielsweise. Die Ansicht der einzelnen Lebensmittel wächst mit der Anzahl an Lebensmitteln. Tippt man ein Lebensmittel an, öffnet man das Fenster Bearbeiten/Löschen hier kann man die Einträge des Lebensmittels verändern oder das Lebensmittel ganz löschen. Dies geschieht ganz einfach mit den jeweiligen Buttons „Löschen“ oder nach einer Änderung „Speichern“. Man kommt nun wie gewohnt automatisch zurück zum vorherigen Fenster.

Auf dem unteren Button „Lebensmittel hinzufügen“ lassen sich nun neue Lebensmittel hinzufügen. Die Felder die auszufüllen sind, sind Name des Lebensmittels, eine kurze Beschreibung, sowie eine Anzahl und die jeweilige Einheit pro Einnahme. Abschließend muss man nur noch die Kalorienanzahl für die jeweilige Menge des Lebensmittels eintragen und kann nun mit dem Button „Hinzufügen“ das Nahrungsmittel zu der Datenbank und zur späteren Auswahl hinzufügen. In der Lebensmittelliste befindet sich nun das

neue Produkt.

## **Menü**

Wie schon gesagt besteht hier eine Ähnlichkeit zu den Lebensmitteln. Oben sind die bereits eingetragenen Menüs zu sehen und mit dem Button Hinzufügen kann man ein neues Menü hinzufügen. Drückt man auf den Button erscheint das Fenster Menü hinzufügen. Hier kann man zuerst einen Namen des Menüs eintragen, sowie eine kurze Beschreibung dazu addieren. In dem Feld Menübestandteile lässt sich mit dem bekannten roten Plus Button ein Lebensmittel zu diesem Menü hinzufügen. Hat man nun alle notwendigen Lebensmittel eingetragen kann man ganz simpel mit dem Button „Hinzufügen“ das Menü eintragen. Die Menüs stehen wie die Lebensmittel zum Eintragen in das Tagebuch im Dropdown Menü zur Verfügung.

Klickt man auf ein bestehendes Menü kann man dieses in dem folgenden Fenster problemlos bearbeiten. Nahrungsmittel können nun ergänzt, angepasst oder gelöscht werden um das Menü auf einen persönlichen Benutzer anzupassen.

## **Mein Profil**

Mit dem letzten Button der Startseite lässt sich das Benutzerprofil bearbeiten. Hier kann man seinen Namen eintragen und weitere benutzerspezifische Daten wie Größe und Gewicht. Für das Geburtsdatum ist ebenfalls ein Feld vorgesehen. Ein wichtiger Punkt ist das Kalorienlimit, welches in dem letzten Feld eingetragen werden kann. Die hier eingetragene tägliche Höchstzahl ist wichtig für die in der Startseite angewandten Zahlen und die Fortschrittsleiste, da diese sich hierauf berufen. Die Änderungen sind mit dem Button „Aktualisieren“ zu beenden, da dieser die eingetragenen Daten abspeichert und anpasst.

In diese Fenster lassen sich außerdem noch die Einheiten anpassen, mit dem Button „Einheiten Übersicht“. Klickt man auf den Button öffnet sich die Einheiten Übersicht, hier sieht man alle zur Verfügung stehenden Einheiten, kann diese anpassen und mit dem Button „Einheit Hinzufügen“ noch neue Einheiten in das Portfolio eintragen. Diese lassen sich selbstverständlich auch bearbeiten und löschen.

Mit einem weiteren Button „Statistische Auswertungen“ öffnet sich das Fenster der statistischen Auswertungen. Hier wird eine Übersicht über die durchschnittlich zu sich genommenen Kalorien der letzten 7, 14 oder 30 Tage gegeben.



## 4 Projektplanung

### 4.1 Pflichtenheft

#### Allgemeines

Dieses Pflichtenheft beschreibt die Anforderungen für die Erstellung einer mobilen Applikation zur Verwaltung der Kalorienzunahme und die Bereitstellung von statistischen Übersichten von einem gewählten Zeitraum.

#### Ziele der Aufgabenstellung

Food4Life möchte mit der Applikation eine Kalorientracker-App in Form eines Tagebuchs ermöglichen. Verschiedene Übersichten über Monate, Tage etc. bieten eine simple und angenehme Oberfläche, zur Verwaltung und Verwendung der einzelnen Lebensmittel und Menüs.

#### Zielgruppe

Die Zielgruppe der Applikation sind Menschen, die ihre Kalorien verwalten wollen.

#### Systemvoraussetzungen

Für die Benutzung der App ist ein Smartphone oder Tablet mit einer Android Version 4.0 oder höher erforderlich. Ein Internetzugang ist nicht erforderlich.

#### Beschreibung der Anforderungen

Die Benutzer der App sollen folgende Funktionen haben:

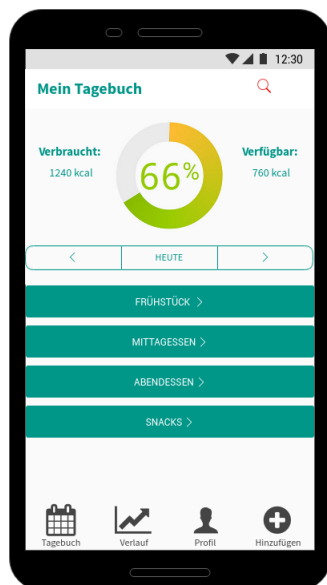
- Kalorientageshöchstwert festlegen und verändern
- Anzeige der Summe der konsumierten Kalorien an einem Tag und des Kalorientageshöchstwert
- Anzeige der durchschnittlichen konsumierten Kalorien der letzten 7, 14 und 30 Tage
- Anzeige aller Einträge des Tagebuches zeitlich aufsteigend sortiert und tageweise untergliedert
- Beliebige Navigation durch den Kalender
- Navigation zum aktuellen Tag im Kalender
- Hinzufügen, Bearbeiten, Kopieren, Verschieben und Löschen eines Eintrags
- Anlegen, Bearbeiten und Löschen von benutzerdefinierten Menüs

- Hinzufügen, Bearbeiten und Löschen eines Lebensmittels
- Hinzufügen und Löschen einer Einheit

## 4.2 Projektablaufplan

## 4.3 Planung der Software

### 4.3.1 Planung der Mockups



Die Startseite der App soll das Herzstück der App darstellen. Beim Öffnen der App soll dem User übersichtlich vermittelt werden, wie viele Kalorien Er/Sie zu sich genommen hat und wie viele Kalorien vom individuellen Tagesbedarf noch zur Verfügung stehen. Zusätzlich soll der Verbrauch der Kalorien visuell mithilfe einer Fortschrittsanzeige (Progress Bar) dargestellt werden. Des Weiteren wird auf der Startseite immer das tagesaktuelle Datum angezeigt. Mithilfe von vier Buttons, soll es dem User möglich sein, die zu sich genommen Lebensmittel einzutragen und kategorisch nach Tageszeit bzw. Mahlzeit zu gliedern. Über die Startseite soll der User zu den weiteren Seiten innerhalb

App gelangen, wobei hierfür mehrere Buttons am Ende des Bildschirmes zur Verfügung stehen. So gelangt der User beispielsweise per Klick zu seinem persönlichen Tagebuch mit Kalenderübersicht oder kann über den Button "Profil" persönliche Daten zu Name, Gewicht, Größe und Kalorientagesbedarf hinterlegen.

#### **4.3.2 Planung der Datenstrukturen und Schnittstellen**

#### **4.3.3 Planung der Activities und Layouts**

#### **4.3.4 Planung der Navigation und des Datenaustausches zwischen den Activities**

### **4.4 Geplante Aufgabenverteilung im Team**

<b>Aufgabe</b>	<b>Teammitglied</b>
Zusammenführung der Einzelteile	Benedikt Burczek
Gruppenweite Koordination	Jannik Maes
Spezifikation und Datenmodell	Benedikt Burczek
Activities	Alle
Navigation zwischen Activities	Sven Kuczera
Studienarbeit	Henryk Schaffrath

## 5 Beschreibung des Projektverlaufs

### 5.1 Tatsächliche Aufgabenverteilung im Team

Aufgabe	Teammitglied
Zusammenführung der Einzelteile	Benedikt Burczek
Gruppenweite Koordination	Jannik Maes
Spezifikation und Datenmodell	Benedikt Burczek
Activities	Alle
Navigation zwischen Activities	Sven Kuczera
Studienarbeit	Henryk Schaffrath



## 5.2 Teammeetingprotokolle

Datum	Dauer (in min)	Inhalt des Meetings
14. Februar	60	Die Aufgabenstellung wurde besprochen und die vorgegebenen Aufgaben auf die einzelnen Teammitglieder verteilt. Zusätzlich wurden erste Ideen zum Aufbau der App sowie der Startseite aufgenommen und dokumentiert. Diese Informationen dienen für erste Entwürfe bzw. für das Mockup.

Datum	Dauer (in min)	Inhalt des Meetings
20. März	30	Der aktuelle Stand der App wurde besprochen sowie neue Ideen und Anreize aufgenommen und dokumentiert. Das Grundgerüst für die benötigten Activities wurde formuliert bzw. aufgrund der neuen Ideen ergänzt. Die nächsten Wochen, weitere Meetings und das weitere Vorgehen wurden intern besprochen sowie geplant.

Datum	Dauer (in min)	Inhalt des Meetings
23. März	180	Ein erster Entwurf des Datenbankmodells wurde durch Benedikt Burczek vorgestellt und mit den anderen Teammitgliedern besprochen. Zusätzlich wurde an den Layouts der Activities weitergearbeitet und eine Beurteilung durch das Team durchgeführt. Daraufhin wurden die Use-Cases gemeinsam spezifiziert und wenn nötig angepasst.

Datum	Dauer (in min)	Inhalt des Meetings
26. März	360	Eine Besprechung zum aktuellen Entwicklungsstands der App wurde durchgeführt. Das Datenbankmodell wurde fertiggestellt und allen Teammitgliedern ausführlich erklärt, damit alle gemeinsames Verständnis haben. Die Datenbank wurde an die App angebunden, sodass nun auch die Bearbeitung der Java Klassen zu den Layouts beginnen konnte. Wobei hier besonders der Fokus auf benötigte Datenbankzugriffe gerichtet war.

Datum	Dauer (in min)	Inhalt des Meetings
28. März	360	Jedes Teammitglied hat an der App bzw. seiner zugeteilten Activity und Java Klasse weitergearbeitet, wobei untereinander ständig im Team kommuniziert wurde, um alles aufeinander abzustimmen und Unklarheiten zu vermeiden.

Datum	Dauer (in min)	Inhalt des Meetings
03. April	420	Der aktuelle Zwischenstand der App wurde besprochen. Dabei lag der Fokus auf der Implementierung der Bearbeitungsfunktion von gesetzten Lebensmittel Einträgen im Tagebuch sprich der Änderung der Stückzahl oder dem Löschen von Einträgen. Des Weiteren wurde von jedem Teammitglied an den noch ausstehenden Activities gearbeitet bzw. Fehlerkorrekturen an bestehenden Activities vorgenommen.

Datum	Dauer (in min)	Inhalt des Meetings
10. April	120	Der aktuelle Zwischenstand bzw. neu ergänzte Funktionalitäten der App wurden besprochen und Tests durchgeführt. Dabei wurde die teilweise etwas unschöne Navigation per "Back-Button" genauer betrachtet und anschließend durch das Team angesprochen und gefixt.

Datum	Dauer (in min)	Inhalt des Meetings
11. April	180	Die App wurde durch alle Teammitglieder getestet, kleine Bugs wurden dokumentiert und auf Teammitglieder zum Fixen verteilt. Zudem wurden letzte Formalitäten für die Abgabe der Studienarbeit und der App geklärt.

### 5.3 Projekttagebücher aller Teammitglieder

### 5.4 Beschreibung von Problemen

#### Known Bugs

- Bei einem Eintrag im Kalorientagebuch kann die Einheit nicht mehr geändert werden.
- Das Kopieren eines Eintrags ist noch nicht möglich, allerdings können diese verschoben werden.
- Mengenangaben bei Menüs noch nicht möglich



- Beim Löschen eines Lebensmittels bleiben Menüs noch nicht unberührt
- Das Löschen von Entsprechungen und das Erhaltenbleiben einer Mindestangabe ist noch nicht möglich
- Die Änderung einer Entsprechung passt bisher nur im Tagebuch automatisch die Kalorienanzahl an
- Kalorientagebucheinträge mit Menüs können noch nicht vollständig, gemäß den Anforderungen, beim Bearbeiten angezeigt werden. Die Daten sind jedoch ordnungsgemäß in der Datenbank hinterlegt

### Umstände im Projekt

Die größten Umstände im Projektverlauf waren vor allem zum einen der mangelnden Zeit, durch die Belegung von Terminen durch Klausuren und Vorträge/Ausarbeitungen bei anderen Kursen und durch Hardware/Software Probleme gegeben.

Das Projekt hat sich zeitlich genau in die Klausurphase gelegt, wodurch ein stetiges konzentriertes Arbeiten ohne Unterbrechungen und Pause leider nicht möglich war. Das ist der Grund warum viel Zeit für die Planung benötigt wurde. Die verschiedenen Teammitglieder hatten unterschiedliche Projekte und Klausuren, zu diversen Terminen. In dem Projekt *food4life* wurden daher viele Teammeetings gehalten, damit Fragen und Weiteres sofort geklärt werden konnten, damit ein Einhalten des Terminplanes, trotz einer straffen Struktur dennoch möglich ist. Außerdem standen noch diverse andere Termine der FHDW dem Zeitmanagement eher kritisch gegenüber.

Ein weiteres großes Problem waren Hard- und Software Probleme, der Android Studio Emulator hat in den besten Fällen schlecht funktioniert, meistens aber gar nicht. Ein erfolgreiches, ununterbrochenes Entwickeln war daher kaum möglich. Das Zurückgreifen auf alternative physische Android Geräte war beschränkt. Verschiedene Unstimmigkeiten in unterschiedlichen Versionen von Android Studio brachten das Projekt auch häufig zu einem kurzen Stopp für mehrere Teammitglieder.

## 6 Dokumentation der Software

### 6.1 Dokumentation der Paketstruktur des Android-Projektes

### 6.2 Dokumentation der Activities

#### 6.2.1 Activity 1

**6.2.1.1 Aufgabe und Funktion** Die Main-Activity umfasst die Startseite der Applikation. Im Header sichtbar ist ein Fortschrittsbalken, der die konsumierten Kalorien mit dem Tagesbedarf an Kalorien gegenüberstellt. Diese Auskunft ist nicht nur visuell, sondern auch textuell hinterlegt mit exakten Werten. In der Fußleiste der Activity findet sich eine Navigationsleiste wieder. In dieser sind vier Knöpfe zu sehen. Das linke Icon zeigt einen Kalender, der Sie beim Klicken in die Kalender-Übersicht bringt. Der Knopf rechts daneben ist betitelt mit „Lebensmittel“. Bei Klick darauf gelangt man in die Lebensmittelübersicht. Der nächste Knopf mit dem Namen „Menüs“ lässt Sie in die Übersicht der Menüs gelangen. Der letzte Knopf der Navigationsleiste zeigt ein Profilicon. Bei Klick auf das Icon gelangen Sie in Ihre persönliche Profilübersicht, bei der Sie die Daten bezüglich ihres Gewichtes, der Größe und den Tagesbedarf an Kalorien ändern können. In der Mitte befinden sich die einzelnen Tageszeiten „Frühstück“, „Mittagessen“, „Abendessen“ und „Snacks“, bei denen man bei Klick auf das rote Plus eine Mahlzeit zur jeweiligen Tageszeit hinzufügen kann. Hierbei wird man auf ein neues Fenster geleitet in der man das Lebensmittel oder ein Menü, die dazugehörige Menge und die Einheit auswählen kann. Die Kalorien werden ausgerechnet und sind nicht veränderbar.

#### 6.2.1.2 Layout, Screenshot(s)

#### 6.2.1.3 Use-Case

<b>Name</b>	Startseite (Main)
<b>Ziel im Kontext</b>	Übersicht des aktuellen Tages mit Informationen zu Kalorienverbrauch und eingenommenen Lebensmitteln
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User öffnet die Applikation
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User öffnet die App, um Übersicht zu erhalten</li> <li>2. User hat folgende Möglichkeiten: <ol style="list-style-type: none"> <li>a. Tagebucheintrag vorzunehmen</li> <li>b. Navigation zum Tagebuch/ Kalender</li> <li>c. Navigation zur Lebensmittelliste</li> <li>d. Navigation zur Menüliste</li> <li>e. Navigation zum persönlichen Profil</li> </ol> </li> </ol>
<b>Erweiterungen</b>	2a. Klick auf Hinzufügen Button (rotes Plus) ist Trigger für andere Activity (AddMeal)

**6.2.1.4 Datenverwaltung innerhalb der Activity** Anzeige: Die gewünschten Parameter werden aus dem entsprechenden Eintrag in der Kalorientagebuchtafel mit Hilfe der Tageszeit und dem Datum in dem Layout angezeigt.

**6.2.1.5 Datenschnittstellen zu anderen Activity** Beim Hinzufügen eines Tagebucheintrags wird die Tageszeit und das Datum als Parameter mitgegeben. Der Tageshöchstwert wird aus der Datenbank-Profiltafel entnommen.

#### 6.2.1.6 Dokumentation des Quelltextes der Activity

### 6.2.2 Activity 2

**6.2.2.1 Aufgabe und Funktion** Die Activity „Kalender“ ist horizontal in zwei Abschnitte geteilt. In dem oberen Teil befindet sich ein Kalender den man durch einfaches Wischen und Klicken bedienen kann. Bei der Ansicht wird der aktuelle Monat angezeigt. In dem unteren Teil der Activity befindet sich die Übersicht aus der Startseite über die Tageszeiten mit der Möglichkeit Mahlzeiten hinzuzufügen. Hier kann der Benutzer entweder zu dem aktuellen Tag weiter Lebensmittel oder Menüs hinzufügen oder zu einem vergangenen Tag eine Mahlzeit hinzufügen. Außerdem ist er in der Lage auf eine

bestimmte Mahlzeit an einem bestimmten Tag zu klicken und somit in ein neues Fenster navigieren, in dem er die Mahlzeit nach Belieben verändern kann.

#### 6.2.2.2 Layout, Screenshot(s)

#### 6.2.2.3 Use-Case

<b>Name</b>	Tagebuch / Kalender
<b>Ziel im Kontext</b>	Übersicht über vergangene, aktuelle und zukünftige Tagebucheinträge
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User hat Kalender-Button auf der Startseite geklickt
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User öffnet den Kalender, um Übersicht zu erhalten</li> <li>2. User wählt gewünschten Tag aus und hat zwei Möglichkeiten: <ol style="list-style-type: none"> <li>a. Eingetragene Lebensmittel/Menüs im Tagebuch einsehen</li> <li>b. Eingetragene Lebensmittel/Menüs hinzufügen, bearbeiten oder löschen</li> </ol> </li> </ol>
<b>Erweiterungen</b>	<p>2b. Klick auf Hinzufügen Button (rotes Plus) ist Trigger für andere Activity (AddMeal)</p> <p>2b1. Klick auf Eintrag ist Trigger für andere Activity (EditOrDeleteMeal)</p>

**6.2.2.4 Datenverwaltung innerhalb der Activity** Anzeige: Die gewünschten Parameter werden aus dem entsprechenden Eintrag in der Kalorientagebuchtafel mit Hilfe der Tageszeit und dem Datum in dem Layout angezeigt.

**6.2.2.5 Datenschnittstellen zu anderen Activity** Beim Hinzufügen eines Tagebucheintrags wird die spezifische Tageszeit und das Datum mitgegeben.

#### 6.2.2.6 Dokumentation des Quelltextes der Activity

### 6.2.3 Activity 3

**6.2.3.1 Aufgabe und Funktion** Die Activity bietet eine Übersicht über die, vom User angelegten, Lebensmitteln in der Datenbank. Außerdem kann der Nutzer per Klick auf „Lebensmittel hinzufügen“ in ein Fenster weitergeleitet werden, in dem er neue Lebensmittel hinzufügen kann. Des Weiteren ermöglicht der Klick auf ein Lebensmittel die Bearbeitung oder Löschung des gewählten Lebensmittels.

#### 6.2.3.2 Layout, Screenshot(s)

#### 6.2.3.3 Use-Case

<b>Name</b>	Lebensmittelübersicht (FoodList)
<b>Ziel im Kontext</b>	Übersicht aller Lebensmittel, die in der Datenbank gespeichert sind
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Lebensmittel“ – Button auf der Startseite
<b>Essenzielle Schritte</b>	1. User klickt auf „Lebensmittel“ – Button, um eine Übersicht über die Lebensmittel in der Datenbank zu erhalten
<b>Erweiterungen</b>	1a. User hat die Möglichkeit den Button „Lebensmittel hinzufügen“ zu klicken um in eine neue Activity (AddFood) zu gelangen

**6.2.3.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Lebensmittel in das Layout geladen.

**6.2.3.5 Datenschnittstellen zu anderen Activity** Der Hinzufügen-Button leitet lediglich auf die entsprechende Activity weiter.

#### 6.2.3.6 Dokumentation des Quelltextes der Activity

## 6.2.4 Activity 4

**6.2.4.1 Aufgabe und Funktion** In der Activity „Lebensmittel hinzufügen“ kann der Benutzer ein neues Lebensmittel in seiner Applikation hinzufügen. Dieses Lebensmittel kann er im Anschluss als Mahlzeit in seinem Tagebuch verwenden oder in ein Menü seiner Wahl hinzufügen. Folgende Informationen zu dem Lebensmittel müssen ausgefüllt werden um ein Lebensmittel erfolgreich zu der Datenbank hinzufügen zu können:

- Name
- Beschreibung
- Anzahl + Einheit
- Kalorien

Bei erfolgreichen Hinzufügen wird das Lebensmittel nun in der Lebensmittelübersicht angezeigt.

## 6.2.4.2 Layout, Screenshot(s)

## 6.2.4.3 Use-Case

<b>Name</b>	Lebensmittel hinzufügen (AddFood)
<b>Ziel im Kontext</b>	Hinzufügen eines Lebensmittels zur Datenbank bzw. Lebensmittelliste
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt in Lebensmittelliste auf Button "Lebensmittel hinzufügen"
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt Button, um Lebensmittel hinzuzufügen</li> <li>2. User gibt Name des Lebensmittels ein</li> <li>3. User gibt Beschreibung für Lebensmittel ein</li> <li>4. User gibt Anzahl/Menge für das Lebensmittel ein</li> <li>5. User wählt Einheit für das Lebensmittel aus</li> <li>6. User gibt Kalorienzahl entsprechend der Menge des Lebensmittels ein</li> <li>7. User bestätigt Eingaben mit Hinzufügen-Button</li> <li>8. Datenbank verarbeitet Eingabe und fügt Lebensmittel der Datenbank hinzu</li> </ol>
<b>Erweiterungen</b>	<p>8a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</p> <p>8a1. Datenbank verarbeitet Eintrag nicht und fügt diesen nicht der Datenbank hinzu</p> <p>8a2. Datenbank benötigt Eingaben in allen Feldern</p>

**6.2.4.4 Datenverwaltung innerhalb der Activity** Anzeige: Die Einheiten werden aus der Tabelle der Einheiten geladen und im Spinner aufgelistet. Hinzufügen: Ein neuer Eintrag wird in der Tabelle der Lebensmittel hinzugefügt, indem die Parameter vom Layout weitergegeben werden.

**6.2.4.5 Datenschnittstellen zu anderen Activity** Schnittstelle zur Einheiten und Entsprechungen Activities vorhanden, um Parameter zu laden und anzuzeigen.

**6.2.4.6 Dokumentation des Quelltextes der Activity**

### **6.2.5 Activity 5**

**6.2.5.1 Aufgabe und Funktion** In der EditOrDeleteFood-Activity hat der User die Möglichkeit, seine Einträge aus der Lebensmittelliste zu bearbeiten oder komplett zu löschen. So können alle zuvor eingetragenen Parameter wie Name, Beschreibung, Anzahl/Menge geändert werden. Dabei bleibt die zuvor eingetragene Einheit jedoch erhalten und kann nicht editiert werden. Zusätzlich kann der User anschließend die Kalorienzahl entsprechend der neuen Parameter eintragen. Zum Speichern des geänderten Lebensmittels steht hierfür der Speichern-Button zur Verfügung. Soll das Lebensmittel jedoch dauerhaft aus der Lebensmittelliste gelöscht werden, so genügt ein Klick auf den Löschen-Button.

### **6.2.5.2 Layout, Screenshot(s)**

### **6.2.5.3 Use-Case**



<b>Name</b>	Bearbeiten / Löschen von Lebensmitteln in der Lebensmittelliste (EditOrDeleteFood)
<b>Ziel im Kontext</b>	Lebensmittel aus der Lebensmittelliste bearbeiten oder löschen
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf Lebensmittel in der Lebensmittelliste
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt Lebensmittel aus der Liste an, um es zu bearbeiten oder zu löschen</li> <li>2. User kann Namen bearbeiten</li> <li>3. User kann Beschreibung bearbeiten</li> <li>4. User kann Menge / Anzahl bearbeiten</li> <li>5. User kann Kalorienwert anpassen</li> <li>6.1 User klickt auf Speichern-Button, um Änderungen zu übernehmen</li> <li>6.2 User klickt auf Löschen-Button, um Lebensmittel dauerhaft zu entfernen</li> <li>7. Datenbank verarbeitet Auswahl und fügt aktualisiertes Lebensmittel der Datenbank hinzu oder löscht es dauerhaft aus der Datenbank</li> </ol>
<b>Erweiterungen</b>	<p>7a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</p> <p>7a1. Datenbank verarbeitet Auswahl/Eingabe nicht und fügt diese nicht der Datenbank hinzu</p> <p>7a2. Datenbank benötigt Eingaben/Auswahl in allen Feldern</p>

**6.2.5.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Lebensmittel und der Entsprechungen in das Layout geladen. Bearbeiten: Die Parameter werden angepasst, indem der aufgerufene Eintrag unverändert in der Datenbank mit einem sogenannten „IS-ACTIVE“ Flag auf 0 beziehungsweise „false“ gesetzt wird. Dadurch werden bereits eingetragene Lebensmittel im Kalorientagebuch nicht verändert. Zusätzlich dazu wird ein komplett neuer Eintrag erstellt, der mit den geänderten Parametern bestückt ist. In der Lebensmittelliste wird nun der neue Eintrag angezeigt und der alte Eintrag ausgeblendet. Löschen: Das Lebensmittel bleibt passiv in der Datenbank gespeichert (Flag „IS-ACTIVE“ wird in der Tabelle auf 0 bzw. "false" gesetzt), um alte Einträge mit dem Lebensmittel nicht ebenfalls aus dem Tagebuch zu löschen. Das gelöschte Lebensmittel wird jedoch nicht

mehr in der Lebensmittelliste angezeigt.

**6.2.5.5 Datenschnittstellen zu anderen Activity** Schnittstelle zur Lebensmittelliste (Activity Food List)

**6.2.5.6 Dokumentation des Quelltextes der Activity**

## 6.2.6 Activity 6

**6.2.6.1 Aufgabe und Funktion** In der Activity „Menü-Übersicht“ werden alle erstellten Menüs in tabellarischer Form angezeigt. Der Nutzer kann auf den Button „Menü hinzufügen“ klicken um ein neues von ihm erstelltes Menü zu der Datenbank hinzuzufügen. Außerdem kann der User auf ein Menü klicken um es dann zu bearbeiten oder zu löschen.

**6.2.6.2 Layout, Screenshot(s)**

**6.2.6.3 Use-Case**

<b>Name</b>	Menüübersicht (MenuList)
<b>Ziel im Kontext</b>	Übersicht aller Menüs, die in der Datenbank gespeichert sind
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Menü“ – Button auf der Startseite
<b>Essenzielle Schritte</b>	1. User klickt auf „Menü“ – Button, um eine Übersicht über die Menüs in der Datenbank zu erhalten
<b>Erweiterungen</b>	1a. User hat die Möglichkeit den Button „Menü hinzufügen“ zu klicken um in eine neue Activity (AddMenu) zu gelangen

**6.2.6.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Menüs in das Layout geladen.

**6.2.6.5 Datenschnittstellen zu anderen Activity** Der Menü Hinzufügen Button leitet lediglich zur entsprechenden Activity weiter.

#### 6.2.6.6 Dokumentation des Quelltextes der Activity

### 6.2.7 Activity 7

**6.2.7.1 Aufgabe und Funktion** In der Activity „Menü hinzufügen“ kann mein Menü hinzufügen. Ein Menü besteht aus mindestens einem Lebensmittel. Das Menü muss einen Namen und eine Beschreibung haben um dies zu der Datenbank hinzufügen zu können. Die Lebensmittel können mit einer Checkbox ähnlichen ListView ausgewählt werden. Mit Hilfe des Buttons „Hinzufügen“ wird das Menü zu der Datenbank hinzugefügt.

#### 6.2.7.2 Layout, Screenshot(s)

#### 6.2.7.3 Use-Case

<b>Name</b>	Menü hinzufügen (AddMenu)
<b>Ziel im Kontext</b>	Hinzufügen eines Menüs zu der Datenbank
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Menü hinzufügen“ in der Menüliste
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt auf „Menü hinzufügen“, um ein Menü in der Datenbank hinzuzufügen</li> <li>2. User betitelt das Menü</li> <li>3. User nennt eine Beschreibung für das Menü</li> <li>4. User kann aus allen Lebensmittel einige mithilfe der Checkbox auswählen</li> <li>5. User bestätigt die Eingaben mit Hinzufügen-Button</li> <li>6. Datenbank verarbeitet Eingabe und fügt Menü der Datenbank hinzu</li> </ol>
<b>Erweiterungen</b>	<p>6a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</p> <p>6a1. Datenbank verarbeitet Eintrag nicht und fügt diesen nicht der Datenbank hinzu</p> <p>6a2. Datenbank benötigt Eingaben in allen Feldern</p>

**6.2.7.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Lebensmittel in das Layout geladen. Hinzufügen: Die eingegebenen Parameter werden in einem neuen Eintrag der Menütabelle der Datenbank gespeichert. Die korrespondierenden Lebensmittel eines Menüs werden in der Zuweisungstabelle von Menü-IDs und Lebensmittel-IDs gespeichert.

**6.2.7.5 Datenschnittstellen zu anderen Activity** Die Lebensmittel-Übersicht bedient sich der Lebensmittel Activity.

#### **6.2.7.6 Dokumentation des Quelltextes der Activity**

### **6.2.8 Activity 8**

**6.2.8.1 Aufgabe und Funktion** In der EditOrDeleteMenu-Activity hat der User die Möglichkeit seine zuvor eingetragenen Menüs aus der Menü-Liste zu bearbeiten oder zu löschen. So können neben dem Namen und der Beschreibung des Menüs auch die Lebensmittel angepasst werden. Hierfür steht dem User eine Checkbox zur Verfügung, über welche er die Lebensmittel einfach aus- oder abwählen kann. Mithilfe des Speichern-Buttons werden alle vorgenommenen Änderungen übernommen und in der Menü-Liste gespeichert. Soll das Menü jedoch dauerhaft aus der Menü-Liste entfernt werden, so genügt ein Klick auf den Löschen-Button.

#### **6.2.8.2 Layout, Screenshot(s)**

#### **6.2.8.3 Use-Case**

<b>Name</b>	Bearbeiten / Löschen von Menüs in der Menüliste ( <u>EditOrDeleteMenu</u> )
<b>Ziel im Kontext</b>	Menü aus der Menüliste bearbeiten oder löschen
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf Menü in der Menüliste
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt Menü aus der Liste an, um es zu bearbeiten oder zu löschen</li> <li>2. User kann Namen bearbeiten</li> <li>3. User kann Beschreibung bearbeiten</li> <li>4. User kann Menübestandteile aus- oder abwählen</li> <li>5.1 User klickt auf Speichern-Button, um Änderungen zu übernehmen</li> <li>5.2 User klickt auf Löschen-Button, um Menü dauerhaft zu entfernen</li> <li>6. Datenbank verarbeitet Auswahl und fügt aktualisiertes Menü der Datenbank hinzu oder löscht es dauerhaft aus der Datenbank</li> </ol>
<b>Erweiterungen</b>	<ol style="list-style-type: none"> <li>6a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</li> <li>6a1. Datenbank verarbeitet Auswahl/Eingabe nicht und fügt diese nicht der Datenbank hinzu</li> <li>6a2. Datenbank benötigt Eingaben/Auswahl in allen Feldern, um es zu aktualisieren</li> </ol>

**6.2.8.4 Datenverwaltung innerhalb der Activity** Anzeige: Die Parameter werden aus dem entsprechenden Eintrag in der Tabelle der Menüs und der Menüs-Lebensmittel-Zuordnung geladen und im Layout angezeigt. Bearbeiten: Der aufgerufene Eintrag wird mit einem Flag auf inaktiv gesetzt und ein neuer Eintrag wird mit den entsprechenden Parametern erstellt. Die Tabelle Menüs und die Tabelle der Zuordnungen kriegen jeweils mindestens einen neuen Eintrag mit den entsprechenden IDS. Löschen: Das Menü bleibt passiv in der Datenbank gespeichert (Flag „IS-ACTIVE“ wird in der Tabelle auf 0 bzw. "false" gesetzt), um alte Einträge mit dem Menü nicht ebenfalls aus dem Tagebuch zu löschen. Das gelöschte Menü wird jedoch nicht mehr in der Menüliste angezeigt.

**6.2.8.5 Datenschnittstellen zu anderen Activity** Schnittstelle zur Menüliste (MenuList-Activity)

**6.2.8.6 Dokumentation des Quelltextes der Activity**

## **6.2.9 Activity 9**

**6.2.9.1 Aufgabe und Funktion** Das Profil des Benutzers wird von dem Benutzer selber ausgefüllt. Der Name, die Größe, das Gewicht, das Geburtsdatum und der Kalorien-tageshöchstwert werden hier festgelegt. Mit dem Button „Aktualisieren“ werden die „alten“ Daten mit den aktuellen Daten überschrieben. Der Knopf „Einheiten-Übersicht“ navigiert den User zu der Einheiten-Übersicht. Mithilfe des Button „Statistische Auswertung“ wird der Nutzer zu einem Fenster mit einer Übersicht seiner durchschnittlich aufgenommenen Kalorien in einem bestimmten Zeitraum geleitet.

**6.2.9.2 Layout, Screenshot(s)**

**6.2.9.3 Use-Case**

<b>Name</b>	Profil (Profile)
<b>Ziel im Kontext</b>	Übersicht über das persönliche Profil/ Aktualisierung der Daten möglich
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Profil“ – Button auf der Startseite
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt „Profil“ – Button auf der Startseite</li> <li>2. User gibt seinen Namen ein</li> <li>3. User gibt seine Größe an</li> <li>4. User gibt sein aktuelles Gewicht an</li> <li>5. User trägt sein Geburtsdatum ein</li> <li>6. User legt seinen Tageskalorienhöchstwert fest</li> <li>7. User bestätigt seine Eingaben mit dem Klick auf „Aktualisieren“</li> <li>8. Datenbank verarbeitet Eingabe und aktualisiert den Tageshöchstkalorienwert, der in der Main eine wichtige Rolle spielt</li> </ol>
<b>Erweiterungen</b>	<ol style="list-style-type: none"> <li>1a. Möglichkeit zur Einheitenliste und zur Statistischen Auswertung zu gelangen per Button Klick</li> <li>6a. Datenbank stellt fest, dass Tageshöchstkalorienwert nicht ausgefüllt ist</li> <li>6a1. Datenbank verarbeitet Eintrag nicht und fügt diesen nicht der Datenbank hinzu</li> <li>6a2. Datenbank benötigt Eingaben in allen Feldern</li> </ol>

**6.2.9.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle des Profils in das Layout geladen. Aktualisieren: Die Parameter werden aktualisiert, indem der Eintrag in der Profiltabelle der Datenbank gemäß den geänderten Werten angepasst wird.

**6.2.9.5 Datenschnittstellen zu anderen Activity** Die Main Activity greift auf den Tageshöchstwert des im Profil festgelegten Wertes zu.

**6.2.9.6 Dokumentation des Quelltextes der Activity**

### 6.2.10 Activity 10

**6.2.10.1 Aufgabe und Funktion** In der Statistics-Activity hat der User die Möglichkeit, seine durchschnittlichen aufgenommenen Kalorien der letzten 7 Tage, 14 Tage und 30 Tage einzusehen.

### 6.2.10.2 Layout, Screenshot(s)

### 6.2.10.3 Use-Case

<b>Name</b>	Statistische Auswertung (Statistics)
<b>Ziel im Kontext</b>	Übersicht über die durchschnittlich eingenommenen Kalorien in einem bestimmten Zeitraum
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Statistische Auswertung“ auf dem Profil
<b>Essenzielle Schritte</b>	1. User klickt auf „Statistische Auswertung“, um sich seine durchschnittlich aufgenommenen Kalorien in den letzten 7, 14 und 30 Tagen angezeigt zu bekommen
<b>Erweiterungen</b>	

**6.2.10.4 Datenverwaltung innerhalb der Activity** Anzeige: Die Parameter werden aus dem entsprechendem Eintrag in der Kalorientagebuchtable geladen, jeweils spezifisch berechnet und anschließend im Layout angezeigt.

**6.2.10.5 Datenschnittstellen zu anderen Activity** Schnittstelle zum Tagebuch (Calendar-Activity)

### 6.2.10.6 Dokumentation des Quelltextes der Activity



### 6.2.11 Activity 11

**6.2.11.1 Aufgabe und Funktion** In dieser Activity sieht der Benutzer eine Übersicht über die verfügbaren Einheiten. Diese kann er mit dem Klick auf „Einheiten hinzufügen“ erweitern. Eine weitere Möglichkeit für den User liegt in dem Klicken auf eine Einheit, woraus sich ein neues Fenster zu Bearbeiten oder Löschen der Einheit.

#### 6.2.11.2 Layout, Screenshot(s)

#### 6.2.11.3 Use-Case

<b>Name</b>	Einheitenübersicht (UnitList)
<b>Ziel im Kontext</b>	Übersicht aller Einheiten, die in der Datenbank gespeichert sind
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Einheitenübersicht“ – Button auf dem Profil
<b>Essenzielle Schritte</b>	1. User klickt auf „Einheitenübersicht“ – Button, um eine Übersicht über die Einheiten in der Datenbank zu erhalten
<b>Erweiterungen</b>	1a. User hat die Möglichkeit den Button „Einheit hinzufügen“ zu klicken um in eine neue Activity (AddUnit) zu gelangen

**6.2.11.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Einheiten in das Layout geladen.

**6.2.11.5 Datenschnittstellen zu anderen Activity** Der Einheiten Hinzufügen Button leitet lediglich zur entsprechenden Activity weiter.

#### 6.2.11.6 Dokumentation des Quelltextes der Activity

### 6.2.12 Activity 12

**6.2.12.1 Aufgabe und Funktion** In dieser Activity hat der User die Möglichkeit eine neue Einheit zu der Datenbank hinzuzufügen. Für diese Aktion wird ein Name benötigt der die Einheit beschreibt. Mit dem Klick auf „Hinzufügen“ wird die Einheit der Datenbank hinzugefügt.

#### 6.2.12.2 Layout, Screenshot(s)

#### 6.2.12.3 Use-Case

<b>Name</b>	Einheit hinzufügen (AddUnit)
<b>Ziel im Kontext</b>	Hinzufügen einer Einheit zu der Datenbank
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf „Einheit hinzufügen“ in der Einheitenübersicht
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User Klickt auf „Einheit hinzufügen“, um eine Einheit in der Datenbank hinzuzufügen</li> <li>2. User gibt Namen der Einheit an</li> <li>3. User bestätigt seine Eingabe durch den Klick auf den „Hinzufügen“ – Button</li> <li>4. Datenbank verarbeitet Eingabe und fügt die Einheit der Datenbank hinzu</li> </ol>
<b>Erweiterungen</b>	<p>6a. Datenbank stellt fest, dass das Feld nicht ausgefüllt ist</p> <p>6a1. Datenbank verarbeitet Eintrag nicht und fügt diesen nicht der Datenbank hinzu</p> <p>6a2. Datenbank benötigt Eingabe in dem Feld</p>

**6.2.12.4 Datenverwaltung innerhalb der Activity** Hinzufügen: Die eingegebenen Parameter werden in einem neuen Eintrag der Einheitentabelle hinzugefügt.

**6.2.12.5 Datenschnittstellen zu anderen Activity** Die hinzugefügten Einheiten werden in der Lebensmittel Activity verwendet.

#### 6.2.12.6 Dokumentation des Quelltextes der Activity

### 6.2.13 Activity 13

**6.2.13.1 Aufgabe und Funktion** In der EditOrDeleteUnit-Activity hat der User die Möglichkeit den Namen einer ausgewählten Einheit aus der Einheitenliste zu bearbeiten oder zu löschen. Über den Speichern-Button werden die Änderungen übernommen und in der Einheiten-Liste gespeichert. Soll die Einheit jedoch dauerhaft aus der Liste entfernt werden, so genügt ein Klick auf den Löschen-Button.

### 6.2.13.2 Layout, Screenshot(s)

### 6.2.13.3 Use-Case

<b>Name</b>	Bearbeiten / Löschen von Einheiten in der Einheitenliste ( <u>EditOrDeleteUnit</u> )
<b>Ziel im Kontext</b>	Einheit aus der Einheitenliste bearbeiten oder löschen
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf Einheit in der Einheitenliste
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt Einheit aus der Liste an, um diese zu bearbeiten</li> <li>2. User kann Namen bearbeiten</li> <li>3.1 User klickt auf Speichern-Button, um Änderungen zu übernehmen</li> <li>3.2 User klickt auf Löschen-Button, um die Einheit dauerhaft zu entfernen.</li> <li>4. Datenbank verarbeitet Änderungen und fügt aktualisierte Einheit der Datenbank hinzu oder löscht diese dauerhaft aus der Datenbank.</li> </ol>
<b>Erweiterungen</b>	<ol style="list-style-type: none"> <li>4a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</li> <li>4a1. Datenbank verarbeitet Auswahl/Eingabe nicht und fügt diese nicht der Datenbank hinzu</li> <li>4a2. Datenbank benötigt Eingaben/Auswahl in allen Feldern, um diese aktualisieren</li> </ol>

**6.2.13.4 Datenverwaltung innerhalb der Activity** Anzeige: Die Parameter werden aus dem entsprechenden Eintrag in der Tabelle der Einheiten geladen und im Layout angezeigt. Bearbeiten: Die Änderungen werden übernommen und in der Datenbank gespeichert. Löschen: Die Einheit bleibt passiv in der Datenbank gespeichert (Flag „IS-ACTIVE“ wird in der Tabelle auf 0 beziehungsweise "false" gesetzt), um alte Einträge mit der Einheit nicht ebenfalls aus dem Tagebuch zu löschen.

**6.2.13.5 Datenschnittstellen zu anderen Activity** Schnittstelle zur Einheiten-Liste (Unit List-Activity)

**6.2.13.6 Dokumentation des Quelltextes der Activity**

## **6.2.14 Activity 14**

**6.2.14.1 Aufgabe und Funktion** In die Activity „Tagebucheintrag hinzufügen“ kommt der Benutzer durch das Klicken des roten Plus auf der Startseite oder im Kalender. Hier kann der Nutzer einen Eintrag hinzufügen zu dem aktuellen Tag, wenn er aus der Startseite kommt. Wenn der Nutzer in dem Kalender auf das Plus klickt, wird der ausgewählte Tag in die Activity übernommen. Der nächste Spinner, also eine Auswahl, stellt den Benutzer vor die Frage ob er die gewählte Tageszeit der Mahlzeit ändern möchte. Die nächsten beiden Auswahleinheiten beziehen sich auf die Lebensmittel und Menüs. Hier kann der Benutzer per Spinner die gewünschte Mahlzeit auswählen. Die hinterlegte Einheit mit den jeweiligen Kalorien wird von der Applikation ausgefüllt und ist nicht mehr zu verändern. Die Menge wird auch von der App ausgefüllt, kann jedoch von dem Benutzer angepasst werden an die Mahlzeit, wie viel er gegessen hat. Die Kalorien passen sich dann je nach Menge der Mahlzeit an und werden angezeigt. Mithilfe des Buttons „Hinzufügen“ wird ein Tagebucheintrag in der Datenbank gespeichert.

**6.2.14.2 Layout, Screenshot(s)**

### 6.2.14.3 Use-Case

<b>Name</b>	Eintrag im Tagebuch vornehmen (AddMeal)
<b>Ziel im Kontext</b>	User trägt Lebensmittel oder Menü in seinem Tagebuch ein
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt Roten Plus-Button auf Startseite oder im Kalender
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt Button, um Tagebucheintrag vorzunehmen</li> <li>2. User wählt über Drop-Down-Menü die Tageszeit aus (Frühstück, Mittagessen, Abendessen, Snack)</li> <li>3. User wählt über Drop-Down-Menü ein Lebensmittel oder Menü aus</li> <li>4. User gibt Anzahl/Menge ein</li> <li>5. User bestätigt Auswahl mit Hinzufügen Button</li> <li>6. Datenbank verarbeitet Auswahl und fügt diese der Datenbank bzw. dem Tagebuch hinzu</li> </ol>
<b>Erweiterungen</b>	<p>6a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</p> <p>6a1. Datenbank verarbeitet Auswahl/Eingabe nicht und fügt diesen nicht der Datenbank hinzu</p> <p>6a2. Datenbank benötigt Eingaben/Auswahl in allen Feldern</p> <p>6b. Datenbank stellt fest, dass kein Lebensmittel oder Menü ausgewählt wurde jedoch aber eine Menge eingetragen wurde</p> <p>6b1. Datenbank verarbeitet Auswahl und fügt automatisch das 1. Lebensmittel aus der Lebensmittelliste der Datenbank bzw. dem Tagebuch hinzu</p>

**6.2.14.4 Datenverwaltung innerhalb der Activity** Anzeige: Die jeweiligen Parameter werden aus dem entsprechenden Eintrag der Tabelle der Lebensmittel, Menüs und Entsprechungen in das Layout geladen. Hinzufügen: Die eingegebenen Parameter werden in einem neuen Eintrag der Kalorientagebuchtable gespeichert.

**6.2.14.5 Datenschnittstellen zu anderen Activity** Das Datum und die Tageszeit werden von der vorherigen Activity gelesen und im Layout eingetragen.

#### **6.2.14.6 Dokumentation des Quelltextes der Activity**

### **6.2.15 Activity 15**

**6.2.15.1 Aufgabe und Funktion** In der EditOrDeleteMeal-Activity hat der User die Möglichkeit, seine Einträge aus dem Tagebuch zu bearbeiten oder zu löschen. Soll der ausgewählte Eintrag komplett gelöscht werden, so steht hierfür der Löschen-Button zur Verfügung. Möchte der User jedoch lediglich seinen Eintrag bearbeiten, so können die Parameter Datum, Auswahl der Mahlzeit, Auswahl des Lebensmittel oder Menü sowie die Menge geändert werden. Über Änderungen am Parameter Datum, können die Einträge innerhalb des Tagebuches auf andere Tage im Kalender verschoben werden. Über ein Drop-Down Menü kann die Auswahl der Mahlzeit geändert werden, sodass Einträge beliebig zwischen Frühstück, Mittagessen, Abendessen oder Snacks verschoben werden können. Zudem dienen zwei weitere Drop-Down Menü zur Auswahl des Lebensmittels oder Menüs aus der Lebensmittelliste/Menüliste, sodass hier ein fälschlicherweise eingetragenes Lebensmittel oder Menü geändert werden kann. Des Weiteren sind Änderungen am Parameter Menge möglich, wobei die Kalorienzahl automatisch aufgrund der hinterlegten Werte in Lebensmittelliste errechnet und angezeigt wird. Über den Speichern-Button werden alle vorgenommenen Änderungen übernommen und anschließend im Tagebuch gespeichert.

#### **6.2.15.2 Layout, Screenshot(s)**

#### **6.2.15.3 Use-Case**

<b>Name</b>	Bearbeiten oder Löschen von Einträgen im Tagebuch (EditOrDeleteMeal)
<b>Ziel im Kontext</b>	User bearbeitet oder löscht Eintrag aus seinem Tagebuch
<b>Akteure</b>	User, Datenbank
<b>Trigger</b>	User klickt auf Eintrag in seinem Tagebuch, entweder auf der Startseite oder dem Kalender
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. User klickt auf Eintrag, um Tagebucheintrag zu bearbeiten oder zu löschen</li> <li>2. User kann Datum über Date-Picker ändern</li> <li>3. User kann über Drop-Down-Menü die Tageszeit (Frühstück, Mittagessen, Abendessen, Snack) ändern</li> <li>4. User kann über Drop-Down-Menü ein anderes Lebensmittel oder Menü auswählen</li> <li>5. User kann Anzahl/Menge ändern</li> <li>6.1 User klickt auf Speichern-Button, um Änderungen zu übernehmen</li> <li>6.2 User klickt auf Löschen-Button, um Eintrag dauerhaft zu entfernen</li> <li>7. Datenbank verarbeitet Änderungen und fügt aktualisierten Eintrag der Datenbank hinzu oder löscht diesen dauerhaft aus der Datenbank.</li> </ol>
<b>Erweiterungen</b>	<p>7a. Datenbank stellt fest, dass nicht alle Felder ausgefüllt sind</p> <p>7a1. Datenbank verarbeitet Auswahl/Eingabe nicht und fügt diesen nicht der Datenbank hinzu</p> <p>7a2. Datenbank benötigt Eingaben/Auswahl in allen Feldern</p>

**6.2.15.4 Datenverwaltung innerhalb der Activity** Anzeige: Die Parameter werden aus dem entsprechenden Eintrag in der Kalorientagebuchtafel geladen und im Layout angezeigt. Bearbeiten: Die Parameter werden in dem entsprechenden Eintrag der Kalorientagebuchtafel angepasst. Löschen: Der Eintrag wird aus der Datenbank komplett entfernt.

**6.2.15.5 Datenschnittstellen zu anderen Activity** Schnittstelle zum Tagebuch (Calendar-Activity) und Startseite (Main-Activity)

**6.2.15.6 Dokumentation des Quelltextes der Activity**

**6.3 Dokumentation der Navigation und des Datenaustauschs zwischen Activities**

**6.4 Dokumentation der Activity-übergreifenden, persistenten Datenhaltung**

**6.5 Dokumentation der Activity-übergreifenden Klassen**



## 7 Fazits aller Teammitglieder

### 7.1 Jannik Maes

Zusammenfassend lässt sich sagen, dass in die Entwicklung einer App neben Zeit auch viel Disziplin und Ehrgeiz gesteckt werden muss, damit das Projekt am Ende erfolgreich ist. Zudem sollte man eine klare Vorstellung vom Endergebnis im Kopf haben und sich persönliche Ziele setzen, welche man im Verlauf des Projektes erreichen möchte. Von kleinen Problemen oder Tiefschlägen bei der Programmierung darf man sich nicht unterkriegen lassen sondern muss stets das Ziel und den Projekterfolg für das gesamte Team vor Augen halten. So können kleine Erfolge bei der Programmierung anstacheln und motivieren, beispielsweise wenn die Activity nach einigen frustrierenden Stunden endlich fehlerfrei funktioniert und den persönlichen sowie gewünschten Vorstellungen entspricht. Mithilfe solcher Erfolge habe ich für mich persönlich neuen Aufschwung und Freude am Projekt gewonnen, sodass ich meine Zeit gerne in das Projekt sowie den Projekterfolg investiert habe. Die Zusammenarbeit mit meinen Teammitgliedern hat mir sehr viel Freude bereitet, was auch prägend für unseren Projekterfolg als Team war und ist. Der Aspekt Kommunikation im Team war sehr wichtig für uns, um etwaige Unstimmigkeiten oder Problemen während der Programmierung der App zu vermeiden. Daher habe ich gerne zahlreiche Teammeetings einberufen und unser motiviertes Team über die Wochen mit viel Freude koordiniert. Wir haben als Team sehr gut zusammengearbeitet und uns gegenseitig unterstützt, wenn es zu Problemen bei der Programmierung an der ein oder anderen Stelle kam. Auf unsere erste eigene App die *food4life* App sind wir als Team sehr stolz und ich persönlich bin auch sehr stolz und zufrieden mit meiner eigenen Programmierleistung. Durch das Projekt konnte ich für mich persönlich viele schöne und lehrreiche Erfahrungen mitnehmen, welche ich fortan in zukünftigen Projekten einsetzen werde.

### 7.2 Sven Kuczera

Das Projekt *food4life* hat viel Zeit und Mühe gekostet, das gesamte Team hat viele lange Stunden an der App und dem allgemeinen Projekt gearbeitet, trotzdem war es für alle ein erfreuliches Arbeiten und vor allem Ergebnis der App. Persönlich lässt sich sagen das ich die Projektzeit auch wenn sie äußerst anstrengend war gut in Erinnerung

behalten werde, was auch an der extrem guten Teamdynamik liegt. Wir haben uns gegenseitig unterstützt und geholfen so oft es ging. Der Fortschritt in dem Projekt war stetig, was einer guten Planung zu danken war. Die Arbeitsverteilung habe ich als sehr gerecht und fair empfunden. Die Kommunikation innerhalb der Gruppe hat sich sehr positiv auf das allgemeine Projekt ausgeübt, sodass keine Umstände und Unstimmigkeiten aufgetreten sind. Bei dem allgemeinen Design und dem Aufbau der App waren wir einstimmig, was das Projekt beschleunigt hat. Die vielen Teammeetings haben sich als ausschlaggebend für ein besseres Verständnis der App und des Projekts herauskristallisiert. Das knobeln an diversen Hürden und Hindernissen hat sich als hilfreich für den gesamten Entwicklungsverlauf der App herausgestellt. Die Hürden die vor allem der fehlenden Programmiererfahrung mit Java zu Grunde lagen habe ich meistern können, was jedoch sehr Zeitaufwändig war. Gegen Ende der App waren die Probleme allerdings beiseitegelegt, sodass hier keine Schwierigkeiten mehr auftraten. Auch wenn es zwischendurch nicht immer so aussah und auch wenn man zwischendurch kurz vor dem aufgeben war, haben wir am Ende eine schöne und voll funktionale App entwickeln können, was noch einmal klargestellt hat, dass sich die ganze Mühe gelohnt hat. Abschließend kann ich sagen das ich das Team, die Entwicklung der App und das ganze Projekt als extrem positiv einordne.

### **7.3 Henryk Schaffrath**

Das Projekt *food4life* hat viel Zeit und Aufwand gekostet. Dennoch hat sich der Aufwand und das Engagement des Teams gelohnt und haben so eine wirklich schöne und gute Applikation für das „Kalorien zählen“ während des Tages erstellt. Die App hat einen ständigen Zuwachs über die Wochen erfahren. Dies verdanken wir der guten Planung und der professionellen Umsetzung dessen. Die Arbeit in unserem Team hat mir sehr gut gefallen. Die häufigen Teammeetings haben dazu beigetragen, dass wir wenig Unstimmigkeiten innerhalb des Teams hatten. Dennoch habe ich festgestellt, dass ich ein paar Defizite beim Programmieren hatte und mir einige Fähigkeiten entweder durch Internetvideos oder durch meine Teammitglieder angeeignet habe, dies war jedoch sehr zeitaufwändig. Allgemein ist zu sagen, dass mir die Arbeit in meinem Team sehr viel Spaß bereitet hat und ich das Arbeitsklima sehr positiv empfunden habe, weswegen das Projekt zu so einem erfolgreichen Endergebnis geführt hat. Mir wird die Zeit positiv im Kopf bleiben, da wir als Team sehr viel geleistet haben und das Projekt erfolgreich abgeschlossen haben.

## **7.4 Benedikt Burczek**

## **8 Anhang**

### **8.1 Anhang mit allen Quelltexten**

### **8.2 Anhang mit einem Verzeichnis der verwendeten Tools und Hilfsprogramme**

- Excel: für die tabellarische Erstellung
- GitHub: für die Projektplanung/ Versionierung der Applikation
- MockFlow: für die Erstellung des Mockups
- Notepad++: für die einfache Codestrukturierung
- Skype: für die Kommunikation innerhalb des Teams
- MS Project: für die Erstellung des Projektplans

## Quellenverzeichnis

## Ehrenwörtliche Erklärung

Hiermit erklären wir, dass die vorliegende Ausarbeitung selbständig angefertigt worden ist. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut haben wir als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form ganz oder teilweise noch keiner Prüfungsbehörde vorgelegen.

---

Ort, Datum, Unterschrift

---

Ort, Datum, Unterschrift

---

Ort, Datum, Unterschrift

---

Ort, Datum, Unterschrift