



# Anwendungsprogrammierung (WIF)

---

Wintersemester 2020

Studienprojekt

## Spezifikation

**Projekt:** *NoPainNoGain*

**Team:** *Katja Wiesmüller*

*Lucas Schunko*

*Marco Schmidl*



Spezifikation

## **NoPainNoGain**

### **Inhaltsverzeichnis**

<b>1</b>	<b>Beschreibung</b>	<b>4</b>
<b>2</b>	<b>Anwendungsfälle</b>	<b>4</b>
<b>2.1</b>	<b>Profil erstellen</b>	<b>5</b>
2.1.1	Anwendungsfall	5
2.1.2	Daten	5
2.1.2.1	Lesend	5
2.1.2.2	Schreibend	5
2.1.3	Ablaufbeschreibung	5
2.1.4	Ergebnis	5
<b>2.2</b>	<b>Login</b>	<b>5</b>
2.2.1	Anwendungsfall	5
2.2.2	Daten	5
2.2.2.1	Lesend	5
2.2.2.2	Schreibend	5
2.2.3	Ablaufbeschreibung	5
2.2.4	Ergebnis	6
<b>2.3</b>	<b>Profil bearbeiten</b>	<b>6</b>
2.3.1	Anwendungsfall	6
2.3.2	Daten	6
2.3.2.1	Lesend	6
2.3.2.2	Schreibend	6
2.3.3	Ablaufbeschreibung	6
2.3.4	Ergebnis	7
<b>2.4</b>	<b>Ziel Festlegen</b>	<b>7</b>
2.4.1	Anwendungsfall	7
2.4.2	Daten	7
2.4.2.1	Lesend	7
2.4.2.2	Schreibend	7
2.4.3	Ablaufbeschreibung	7
2.4.4	Ergebnis	8
<b>2.5</b>	<b>Kalorienbedarf bestimmen</b>	<b>8</b>
2.5.1	Anwendungsfall	8
2.5.2	Daten	8
2.5.2.1	Lesend	8
2.5.2.2	Schreibend	8
2.5.3	Ablaufbeschreibung	8



Spezifikation

**NoPainNoGain**

2.5.4	Ergebnis	9
<b>2.6</b>	<b>Kalorien zählen</b>	<b>9</b>
2.6.1	Anwendungsfall	9
2.6.2	Daten	9
2.6.2.1	Lesend	9
2.6.2.2	Schreibend	9
2.6.3	Ablaufbeschreibung	9
2.6.4	Ergebnis	10
<b>2.7</b>	<b>Gewichtsangabe</b>	<b>10</b>
2.7.1	Anwendungsfall	10
2.7.2	Daten	10
2.7.2.1	Lesend	10
2.7.2.2	Schreibend	10
2.7.3	Ablaufbeschreibung	10
2.7.4	Ergebnis	10
<b>2.8</b>	<b>Profil anzeigen</b>	<b>10</b>
2.8.1	Anwendungsfall	10
2.8.2	Daten	10
2.8.2.1	Lesend	10
2.8.2.2	Schreibend	10
2.8.3	Ablaufbeschreibung	10
<b>2.9</b>	<b>Report</b>	<b>11</b>
2.9.1	Anwendungsfall	11
2.9.2	Daten	11
2.9.2.1	Lesend	11
2.9.2.2	Schreibend	11
2.9.3	Ablaufbeschreibung	11
2.9.4	Ergebnis	11
<b>3</b>	<b>Masken</b>	<b>12</b>
3.1	Maske „Startbildschirm“	12
3.2	Maske „Sign up“	12
3.3	Maske „Login“	13
3.4	Maske „Edit Profile“	13
3.5	Maske „Help“	14
3.6	Maske „Set Goal“	14
3.7	Maske „Home“	15
3.8	Maske „Kalorien zählen“	15



Spezifikation

**NoPainNoGain**

3.9	Maske „Weight“	16
3.10	Maske „Report“	16
3.11	Maske „Profile“	17
3.12	GUI - Landkarte“	18
4	Fachliches Datenmodell	19
4.1	ER - Modell	19
4.2	UML – Diagramm	19
5	Komponenten	20
5.1	Komponentenstruktur	20
5.2	Interfacebeschreibung und Signaturen	20
5.2.1	IUser	20
5.2.2	IUserAdministration	20
5.2.3	IUserInformation	20
5.2.4	IUserPersistence	21
5.2.5	ICalculation	21
5.2.6	ICalculationPersistence	21
5.2.7	IDailyUse	21
5.2.8	IProductAdministration	21
5.2.9	IProduct	21
5.2.10	IProductPersistence	21
5.2.11	IConsumption	22
5.2.12	IConsumptionPersistence	22
5.2.13	IUserConsumption	22
5.2.14	IReport	22
6	Validierung	22
6.1	Test Login	22
6.2	Test Profil erstellen	22
6.3	Test Profil bearbeiten	22
6.4	Test Datenbank	23
6.5	Test Kalorienbedarf bestimmen und verrechnen	23
7	Aufteilung der Aufgaben	24
7.1	Lucas Schunko	24
7.2	Katja Wiesmüller	24
7.3	Marco Schmidl	24



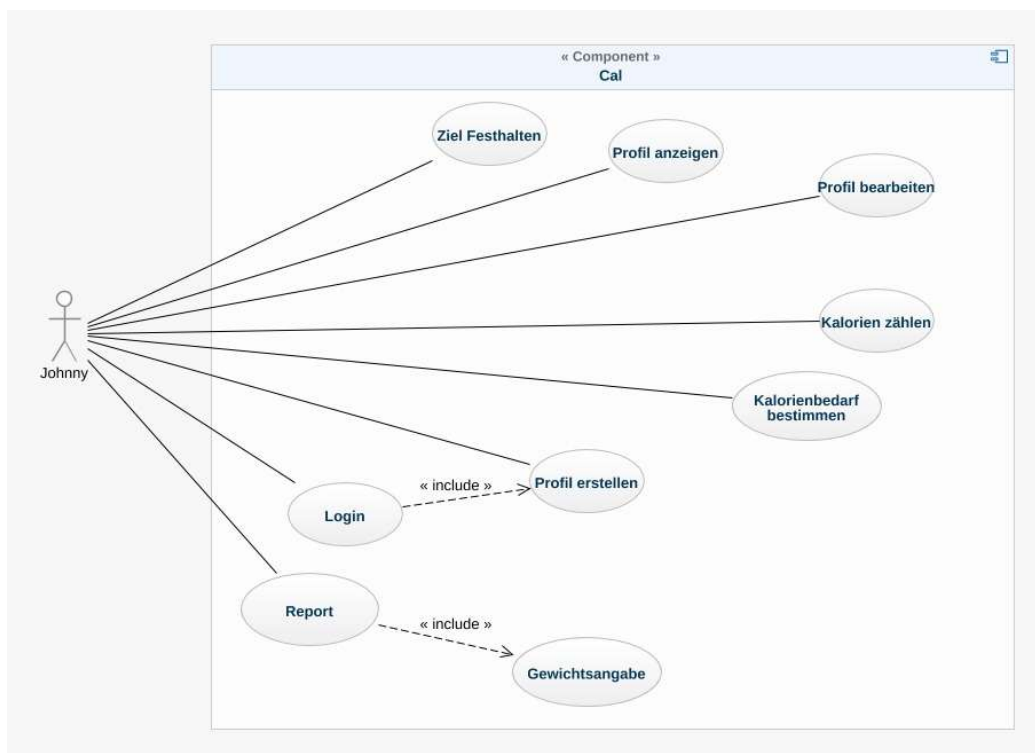
## Spezifikation **NoPainNoGain**

### 1 Beschreibung

Die Desktop Anwendung Cal ist ein digitaler Ernährungsassistent und kann von jeder Person genutzt werden. Die Anwendung soll dem Benutzer dabei helfen ein vorab festgelegtes Gewichtsziel zu erreichen. Dabei ermöglicht die Software es dem User seine täglich aufgenommenen Kalorien mit dem internen Algorithmus abzugleichen und eine aktuelle grafische Kalorienbilanz einzusehen. Um die Datenverarbeitung der täglichen Essgewohnheiten zu vereinfachen, kann der Nutzer Nahrungsmittel speichern und diese wieder abrufen. Zusätzlich können andere Nutzer ebenfalls bereits von anderen Nutzern erstellte Lebensmittel abrufen. Das Zusammenspielen der zwei Komponenten visualisiert damit, ob ein Überschuss oder Defizit von Kalorien zum eigentlichen Bedarf existiert und hilft durch tägliche Anwendung das gewünschte Ziel zu erreichen.

### 2 Anwendungsfälle

In den folgenden Unterpunkten werden alle vorhandenen Anwendungsfälle und die damit verbundenen Abläufe sowie auch Datenabhängigkeiten beschrieben.





Spezifikation

## **NoPainNoGain**

### **2.1 Profil erstellen**

#### **2.1.1 Anwendungsfall**

Der Nutzer will ein Profil erstellen, mit dem er auf die Anwendung zugreifen kann. Sämtliche persönliche Daten, sowie die Kalorienbilanz und Essgewohnheiten werden im Profil gespeichert.

#### **2.1.2 Daten**

##### **2.1.2.1 Lesend**

- E-Mail (Test, ob vorhanden)

##### **2.1.2.2 Schreibend**

- Vorname, Nachname, E-Mail, Passwort

#### **2.1.3 Ablaufbeschreibung**

Der User gelangt auf den Login-Screen. Hier muss er vier Felder: Vorname, Nachname, E-Mail Adresse und Passwort mit seinen Angaben befüllen. Sofern die E-Mail nicht bereits verwendet wurde und alle Angaben den syntaktischen Vorgaben entsprechen, speichert das System diese in der Datenbank ab und ermöglicht dem Nutzer das Abrufen seiner Daten.

#### **2.1.4 Ergebnis**

Der Nutzer hat sein Profil erstellt und kann von nun an alle Funktionen der Anwendung nutzen.

### **2.2 Login**

#### **2.2.1 Anwendungsfall**

Der Nutzer will sich in sein bereits bestehendes Konto einloggen. Dabei muss er zwischen den Funktionen „Sign up“ und „Login“ wählen. Über „Login“ kann der Nutzer sich einloggen. Mit Klick auf „Sign up“ gelangt der Nutzer zum vorherigen Usecase.

#### **2.2.2 Daten**

##### **2.2.2.1 Lesend**

- E-Mail, Passwort

##### **2.2.2.2 Schreibend**

- Keine

#### **2.2.3 Ablaufbeschreibung**

Sofern ein Konto besteht, kann durch die Eingabe der E-Mail Adresse und des definierten Passwortes darauf zugegriffen werden. Das System gleicht die Eingaben mit den Daten der Datenbank ab. Falls der Nutzer einen Fehler bei der Eingabe macht, bspw. durch eine nicht existierende E-Mail-Adresse,



Spezifikation

## **NoPainNoGain**

unzulässige Angaben oder ein falsches Passwort, wird das System ihn darauf hinweisen und weitere Versuche zur Korrektur ermöglichen.

### **2.2.4 Ergebnis**

Der Nutzer hat sich in seinem zuvor erstellten Konto angemeldet. Dies ist notwendig um die Anwendung nutzen zu können.

## **2.3 Profil bearbeiten**

### **2.3.1 Anwendungsfall**

Um sein Profil zu vervollständigen trägt der Nutzer Informationen zu seinem Alter, Gewicht, Geschlecht und seiner Alltagsaktivität ein. Diese Daten werden vom Algorithmus benötigt, um exaktere Ergebnisse zu errechnen und werden ebenfalls in der Datenbank gespeichert. Die Daten sind für eine genaue Berechnung notwendig.

### **2.3.2 Daten**

#### **2.3.2.1 Lesend**

- Keine

#### **2.3.2.2 Schreibend**

- Geschlecht, Größe, Aktuelles Gewicht, Bewegungsfaktor

### **2.3.3 Ablaufbeschreibung**

Der Nutzer wird in der Hauptumgebung nach der Anmeldung dazu aufgefordert, sein Profil um weitere Informationen zu ergänzen. Dabei soll er zuerst auswählen um welches Geschlecht es sich handelt. Zusätzlich wird er gebeten sein Alter und seine Größe im System zu hinterlegen. Die Daten werden auf syntaktische Richtigkeit überprüft. Außerdem werden das Gewicht und das Alter in einem realistischen Rahmen beschränkt. Falls es zu einer Überschreitung kommt, wird eine Fehlermeldung an den User gesendet. Einen weiteren Kern der Berechnung stellt dabei noch der tägliche Bewegungsfaktor da, welcher die Nutzer in 5 Gruppen kategorisiert.

1. Faktor 1,2 (Button 1) = Menschen mit ausschließlich sitzender bzw. liegender Lebensweise. Bettlägerige oder sehr gebrechliche Menschen zählen zu dieser Gruppe.
2. Faktor 1,4 (Button 2) = Menschen mit hauptsächlich sitzender Arbeitsweise und spärlichen sportlichen Freizeitaktivitäten. Alle die viel am Schreibtisch sitzen und keinen Sport betreiben.
3. Faktor 1,7 (Button 3) = Menschen, die sitzend arbeiten aber auch stehenden und gehenden Tätigkeiten nachgehen. Z.B. Studenten.



Spezifikation

## **NoPainNoGain**

4. Faktor 1,9 (Button 4) = Menschen, die täglich sehr viel gehen bzw. stehen. Zum Beispiel Hausfrauen, Kellner, Handwerker oder Verkäufer.
5. Faktor 2,2 (Button 5) = Menschen mit körperlich anstrengenden Tätigkeiten wie Landwirte, Bergleute oder Leistungssportler

Der Nutzer kann diese Angaben jederzeit aktualisieren.

### **2.3.4 Ergebnis**

Der Nutzer hat die angeforderten Daten angegeben. Nun kann das System anhand dieser Angaben das Profil auf den Nutzer einstellen.

## **2.4 Ziel Festlegen**

### **2.4.1 Anwendungsfall**

Der Nutzer will sein Ziel festlegen und bestimmen welches Gewicht er erreichen will. Dabei muss er auf abnehmen („lose“), Gewicht halten („maintain“) oder Gewicht zunehmen („gain“) klicken und ein Zielgewicht eingeben.

### **2.4.2 Daten**

#### **2.4.2.1 Lesend**

- Keine

#### **2.4.2.2 Schreibend**

- Ziel, Zielgewicht

### **2.4.3 Ablaufbeschreibung**

Das System bietet dem User die Möglichkeit sein Ziel zu kategorisieren. Die Möglichkeiten belaufen sich dabei auf:

1. Abnehmen = Dies wird empfohlen, wenn der Nutzer vorhat in einem mittleren Maß Gewicht zu verlieren. Das Defizit wird sich auf 300 KCAL belaufen und stellt damit ein moderates Defizit dar.
2. Gewicht halten = Dies wird empfohlen, wenn das Gewicht gleichbleiben soll und der Fokus daraufgelegt wird, die Ernährung qualitativ zu verbessern. Die Verzehrempfehlung liegt 1:1 beim berechneten Tagesbedarf.
3. Muskeln aufbauen = Dies wird empfohlen, wenn der Nutzer sich das Ziel gesetzt hat, Muskeln aufzubauen und durch einen durchgehenden Überschuss an KCAL die Muskelsynthese zu unterstützen. Hierbei wird mit einem Überschuss von 250 KCAL





Spezifikation

## **NoPainNoGain**

gearbeitet. Dieser Wert soll für eine kontinuierliche Steigerung des Gewichts sorgen und die Steigerung des Körperfettanteils möglichst moderat halten.

Zusätzlich kann der Nutzer ein Zielgewicht festlegen. Dieses Gewicht wird bei den Berechnungen nicht miteinbezogen, sondern dient nur der eigenen Kontrolle. Jedoch hat er die Möglichkeit, dieses nach Belieben wieder zu ändern und an seine derzeitigen Ziele anzupassen. Diese Angabe wird syntaktisch nicht überprüft, da sie keine Relevanz zur Berechnung hat und nur zu Übersicht genutzt wird.

### **2.4.4 Ergebnis**

Sobald der Nutzer sein Ziel festgelegt hat, stehen dem System alle relevanten Daten zur Verfügung, um den Kalorienbedarf zu bestimmen.

## **2.5 Kalorienbedarf bestimmen**

### **2.5.1 Anwendungsfall**

Der Nutzer befindet sich jetzt im Hauptbereich und will wissen wie viel Kalorien er zu sich nehmen muss.

### **2.5.2 Daten**

#### **2.5.2.1 Lesend**

- Kalorienbedarf

#### **2.5.2.2 Schreibend**

- Kalorienbedarf

### **2.5.3 Ablaufbeschreibung**

Das System entnimmt aus den bisher gesammelten Daten: Alter, Größe, Gewicht, Geschlecht, Bewegungsfaktor und Ziel, relevante Information für die Berechnung des gesamten Bedarfs. Aus den ersten vier Datensätzen berechnet es den Grundumsatz. Dies stellt dar, was der Körper in völliger Ruhe verbraucht, um alle körperlichen Funktionen aufrecht zu erhalten.

#### **Grundumsatz berechnen**

Über die folgenden Formeln kann sich Frau und Mann den persönlichen Grundumsatz errechnen.

- Für Frauen =  $655,1 + (9,563 \times \text{Gewicht(kg)}) + (1,850 \times \text{Größe(cm)}) - (4,676 \times \text{Alter})$
- Für Männer =  $66,5 + (13,75 \times \text{Gewicht(kg)}) + (5,003 \times \text{Größe(cm)}) - (6,775 \times \text{Alter})$



Spezifikation

## **NoPainNoGain**

### **Leistungsumsatz berechnen**

Nachdem das System den Grundumsatz berechnet hat, wird mit der nächsten Formel der Leistungsumsatz berechnet. Für Männer & Frauen = Grundumsatz x (Bewegungsfaktor – 1).

**Grundumsatz + Leistungsumsatz = Tagesumsatz.**

Der Tagesumsatz ist der Wert, welcher an den User zurückgegeben wird nachdem er mit dem Ziel verarbeitet wurde und stellt den täglichen Arbeitswert da.

### **2.5.4 Ergebnis**

Dem Nutzer wird der tägliche Kalorienbedarf berechnet und bereitgestellt. Dieser ist ausschlaggebend für die weiteren Anwendungsfälle.

## **2.6 Kalorien zählen**

### **2.6.1 Anwendungsfall**

Der Nutzer will in der Applikation festhalten was er täglich isst. Da er dies dauerhaft tun will, kann er Nahrungsmittel in der Datenbank abspeichern und diese nach Belieben wieder abrufen.

### **2.6.2 Daten**

#### **2.6.2.1 Lesend**

- Nahrungsmittel

#### **2.6.2.2 Schreibend**

- Nahrungsmittel

### **2.6.3 Ablaufbeschreibung**

Das System bietet die Möglichkeit die Inhaltswerte von 100g des Produktes in der Datenbank abzuspeichern und wiederzuverwenden. Dabei nimmt das System den festen Wert und berechnet je nach verspeister Menge die entsprechenden Beträge. Diese gibt es an den Nutzer zurück, um ihm zu veranschaulichen, welche Menge an Nährstoffen er zu sich genommen hat. Daraus wird dann in Kombination mit dem berechneten Wert für den Kalorienbedarf eine Soll/Habenbilanz erstellt.

Der Nutzer kann zwischen zwei Funktionen wählen:

1. Nahrungsmittel neu hinzufügen = Das Produkt wird anhand von den Daten von 100g, welche auf Verpackungen sind gespeichert. Der Nutzer gibt den Eiweißgehalt, Kohlenhydrate, Fettanteil und den gesamte KCAL Wert ein.
2. Nahrungsmittel benutzen = Die bereits gespeicherten Nahrungsmittel werden aufgelistet. Auf diese Produkte kann der User auch über eine Datenbank mit bereits erstellten Nahrungsmitteln zugreifen.



Spezifikation

## **NoPainNoGain**

### **2.6.4 Ergebnis**

Der Nutzer hat seinen Konsum dokumentiert und bekommt seinen restlichen Tagesbedarf zurückgeliefert.

## **2.7 Gewichtsangabe**

### **2.7.1 Anwendungsfall**

Der Nutzer will festhalten wie viel er wiegt, um später seinen Fortschritt einsehen zu können.

### **2.7.2 Daten**

#### **2.7.2.1 Lesend**

- Gewicht

#### **2.7.2.2 Schreibend**

- Gewicht

### **2.7.3 Ablaufbeschreibung**

Das System bietet dem Nutzer die Möglichkeit, sein Gewicht manuell zu aktualisieren. Dabei gibt der Nutzer täglich sein Gewicht in das dafür vorgesehene Feld ein. Zu einem späteren Zeitpunkt kann er seinen Gewichtsverlauf einsehen.

### **2.7.4 Ergebnis**

Der Nutzer hat sein Gewicht aktualisiert. Dieser Wert wird vom System für die Verrechnung des Reports verwendet.

## **2.8 Profil anzeigen**

### **2.8.1 Anwendungsfall**

Der User will sein Profil einsehen.

### **2.8.2 Daten**

#### **2.8.2.1 Lesend**

- Gewicht, Größe, Geschlecht, Alter, Ziel, Zielgewicht

#### **2.8.2.2 Schreibend**

- Keine

### **2.8.3 Ablaufbeschreibung**

In der Funktion Profil anzeigen lassen, zeigt das System dem User seine hinterlegten Daten.



Spezifikation

## ***NoPainNoGain***

## **2.9 Report**

### **2.9.1 Anwendungsfall**

Der Nutzer will wissen wie sich sein Kalorienverbrauch und sein Gewicht entwickelt.

### **2.9.2 Daten**

#### **2.9.2.1 Lesend**

- Kalorienverbrauch, Gewicht

#### **2.9.2.2 Schreibend**

- Keine

### **2.9.3 Ablaufbeschreibung**

Das System zeichnet den Kalorienverbrauch des Nutzers auf und speichert es in der Datenbank. Beim Abrufen der Funktion „Report“, werden die Werte der vergangenen Woche aufgerufen und angezeigt.

### **2.9.4 Ergebnis**

Der Nutzer sieht seinen Kalorienverbrauch und sein Gewicht. Er kann hieraus ableiten, ob er mit seiner Ernährung Fortschritte erzielt hat oder nicht.

Spezifikation  
**NoPainNoGain**

### 3 Masken

#### 3.1 Maske „Startbildschirm“

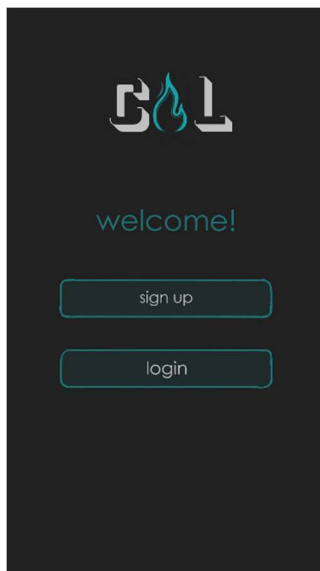


Abb. 1: Startbildschirm

Die hier abgebildete Maske zeigt den Startbildschirm. Dieser wird dem User direkt nach dem Öffnen der Anwendung angezeigt. Er hat hier die Möglichkeit ein Profil zu erstellen oder sich in einem bestehenden Profil anzumelden.

#### 3.2 Maske „Sign up“

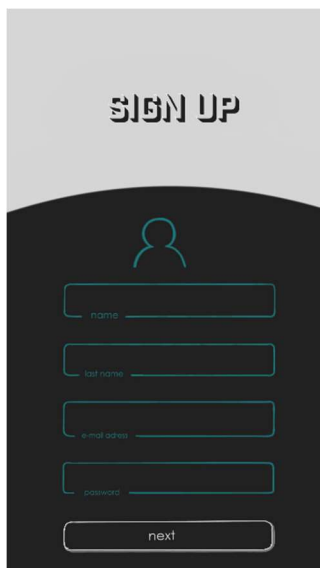


Abb. 2: Sign up

Der Nutzer hat auf den „sign up“ Button geklickt und gelangt damit in die „sign up“ Maske.

Er wird aufgefordert die für die Erstellung eines Profils relevanten Daten in den Freitextfeldern anzugeben (name, last name, e-mail adress, password).

Hierbei handelt es sich um Pflichtfelder. Durch Klick auf den Button „next“ gelangt der Nutzer zu „edit profile“.

## Spezifikation **NoPainNoGain**

### 3.3 Maske „Login“

Abb. 3: Login

Mit dem Klick auf den „login“ Button gelangt der User auf die Anmeldeseite. Hier kann er sich bei einem bestehenden Profil anmelden.

Er wird aufgefordert E-Mail-Adresse und Passwort in den Freitextfeldern anzugeben.

Auch hier handelt es sich um Pflichtfelder. Mit Klick auf den Button „login“ gelangt der Nutzer zu „Home“.

### 3.4 Maske „Edit Profile“

Abb. 4: Edit Profile

Nachdem der User sein Profil angelegt hat, wird er dazu aufgefordert Angaben zu Alter, Gewicht, Größe und Aktivität zu machen.

Die Aktivität kann in fünf Aktivitätsstufen eingeordnet werden. Der Nutzer muss dafür einen der fünf Buttons auswählen. Sollte er dabei Schwierigkeiten haben, kann er über den Hilfe-Button eine Erklärung zu den fünf Aktivitätsstufen abrufen.

Bei den Eingabefeldern und der Aktivität handelt es sich um Pflichtfelder. Mit Klick auf den Button „next“ gelangt der Nutzer zu „set goal“.

## Spezifikation **NoPainNoGain**

### 3.5 Maske „Help“



Abb. 5: Help

Für den Fall, dass der User Schwierigkeiten hat sein Aktivitätslevel einzuschätzen, wird ihm hier eine Definition der einzelnen Stufen bereitgestellt.

Dabei handelt es sich um rein informative Inhalte.

Oben links gelangt der Nutzer durch Klick auf den Button „<<“ zurück zu „edit profile“.

### 3.6 Maske „Set Goal“

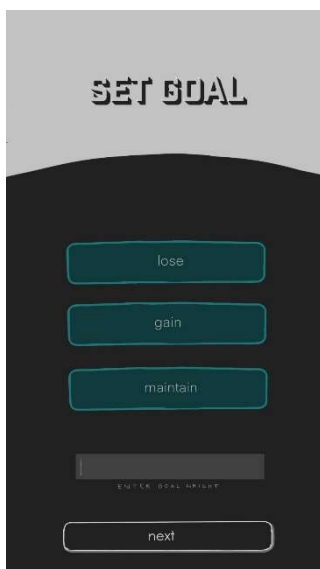


Abb. 6: Set Goal

Im nächsten Schritt soll der User sein Ziel festlegen. Ihm stehen folgende Möglichkeiten zur Auswahl (Buttons):

- „lose“ (Gewicht verlieren)
- „gain“ (Gewicht zunehmen)
- „maintain“ (Gewicht halten)

Er muss einen der drei Buttons wählen.

Im Eingabefeld wird der User aufgefordert sein gewünschtes Zielgewicht einzugeben. Hierbei handelt es sich um ein Pflichtfeld.

Mit Klick auf den Button „next“ gelangt der Nutzer zu „Home“.

## Spezifikation **NoPainNoGain**

### 3.7 Maske „Home“

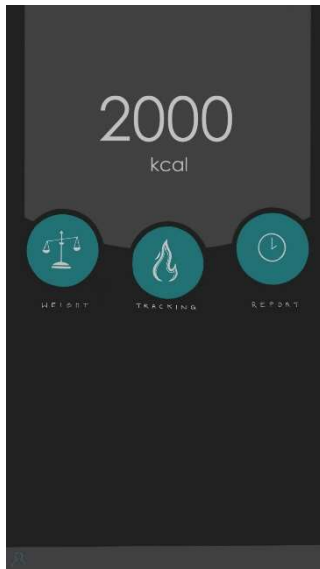


Abb. 7: Home

Nachdem er sein Profil erfolgreich konfiguriert hat wird der User auf den Home-Bildschirm weitergeleitet. Hier wird ihm sein täglicher Kalorienbedarf angezeigt und er gelangt zu allen Funktionen die die Anwendung bietet: (v.l.n.r)

- Weight
- Tracker
- Report
- Profile (unten links)

### 3.8 Maske „Kalorien zählen“

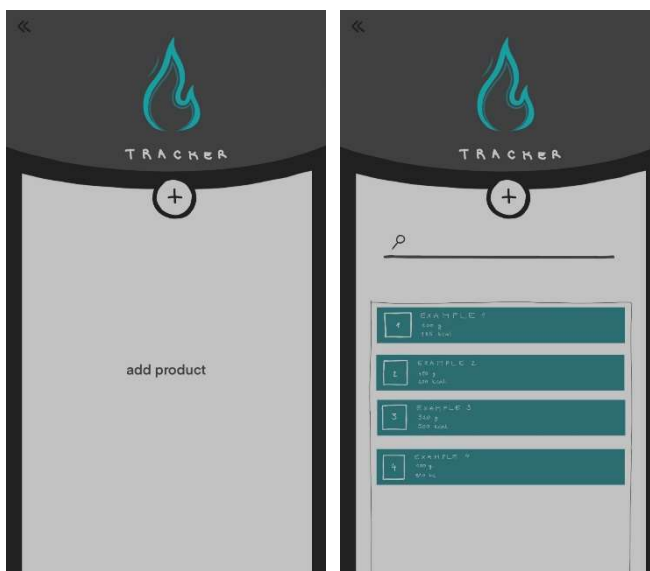


Abb. 8.1: Tracker (empty)

Abb. 8.2: Tracker (filled)

Der User hat sich dazu entschieden die konsumierten Lebensmittel des jeweiligen Tages zu tracken.

Er kann mithilfe des „+“- Buttons Produkte aus der Datenbank auswählen und in der Übersicht dokumentieren.

Abb. 8.1 zeigt die leere Oberfläche. Der User hat demnach noch keine Produkte gespeichert.

Abb. 8.2 zeigt die Oberfläche nachdem der User seine Lebensmittel ausgewählt und gespeichert hat.

Oben links gelangt der Nutzer durch Klick auf den Button „<<“ zurück zu „Home“.



Spezifikation  
**NoPainNoGain**

### 3.9 Maske „Weight“



Abb. 9: Weight

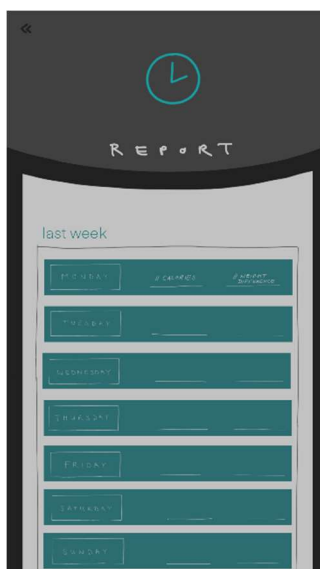
Um seinen Fortschritt vom System berechnen zu lassen, kann der User sein Gewicht täglich angeben.

Dazu wird er in dem Eingabefeld aufgefordert.

Außerdem hat er die Möglichkeit über den „show progress“ – Button den Report, also seinen Fortschritt anzeigen zu lassen.

Oben links gelangt der Nutzer durch Klick auf den Button „<<“ zurück zu „Home“.

### 3.10 Maske „Report“



In der hier abgebildeten Maske wird der Fortschritt des Users angezeigt.

Hierhin gelangt er entweder über den Startbildschirm oder über die Maske „Weight“.

Der Report zeigt das Gewicht und den täglichen Kalorienverbrauch der letzten Woche.

Oben links gelangt der Nutzer durch Klick auf den Button „<<“ zurück zu „Home“.

Spezifikation  
**NoPainNoGain**

### 3.11 Maske „Profile“

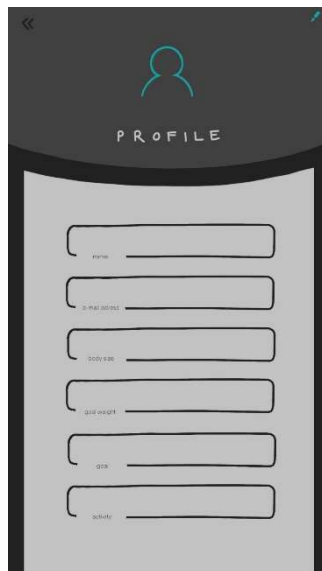


Abb. 11: Profile

Um seine Daten zu überprüfen und gegebenenfalls zu ändern, kann der User sein Profil aufrufen.

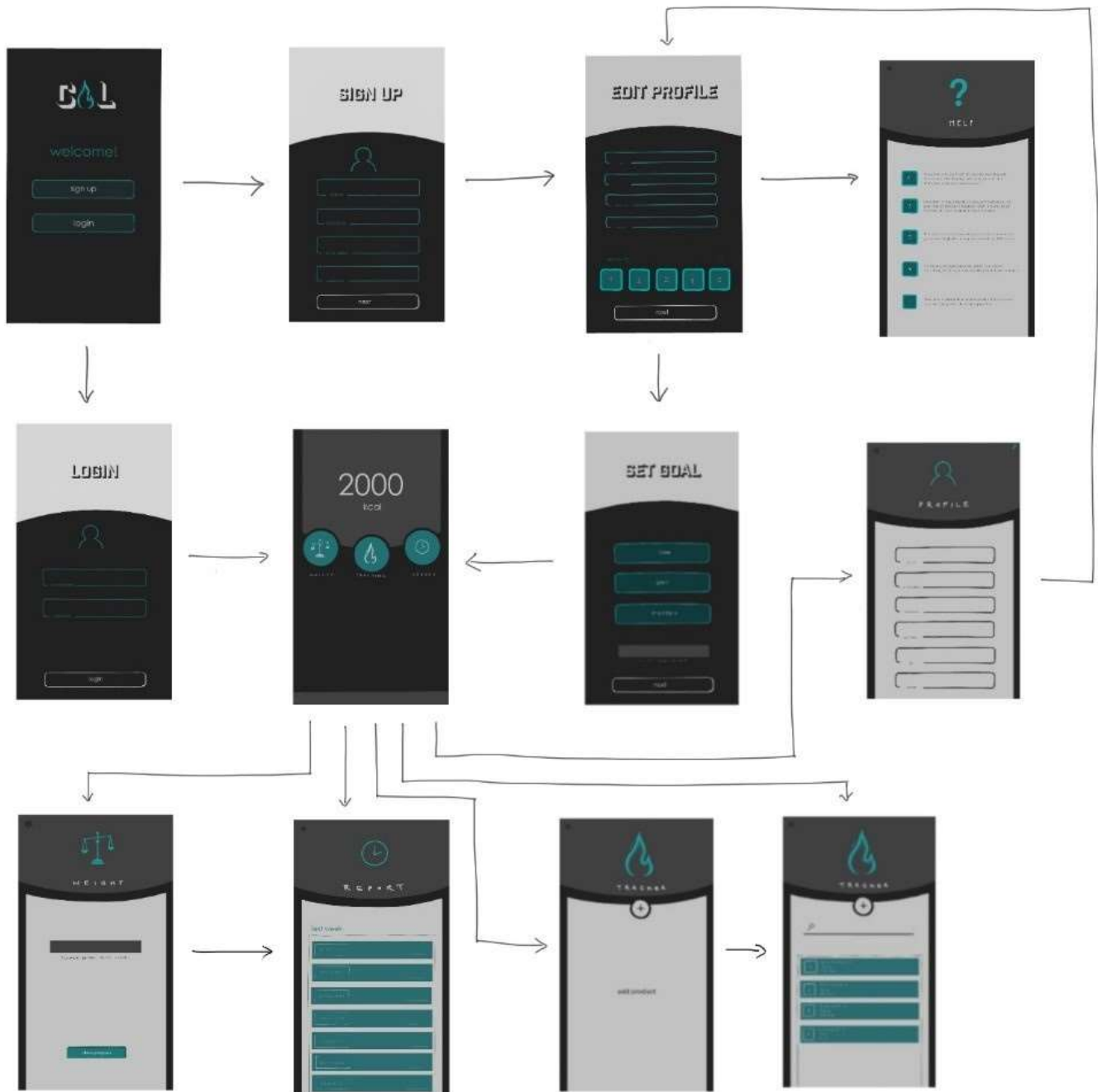
Hier findet er alle von ihm angegebenen Informationen (name, e-mail adress, body size, goal weight, goal, activity)

Oben links gelangt der Nutzer durch Klick auf den Button „<<“ zurück zu „Home“.

Oben rechts gelangt er durch Klick auf den Bleistift zu „edit profile“.

Spezifikation  
**NoPainNoGain**

### 3.12 GUI - Landkarte“

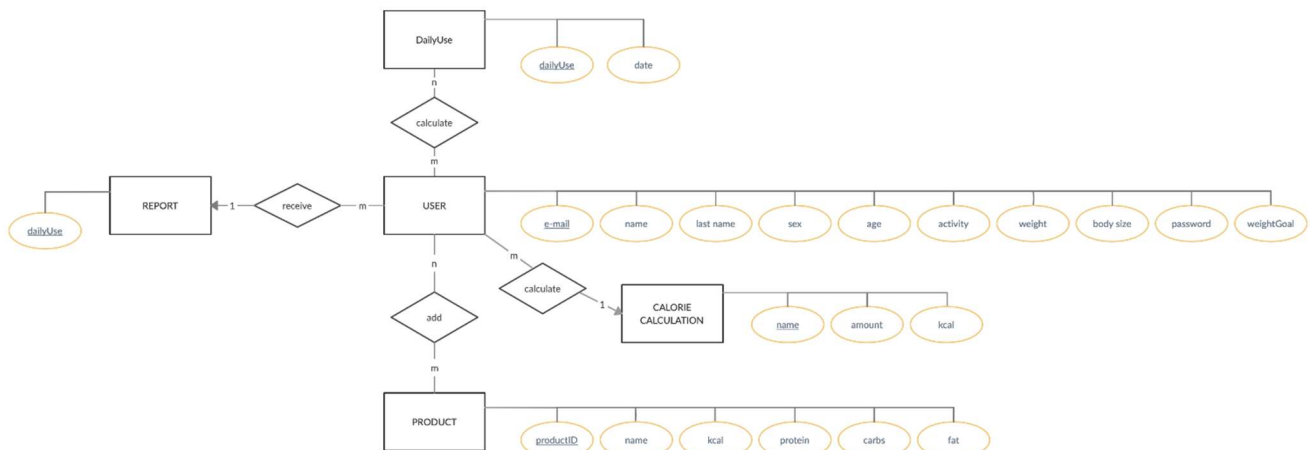




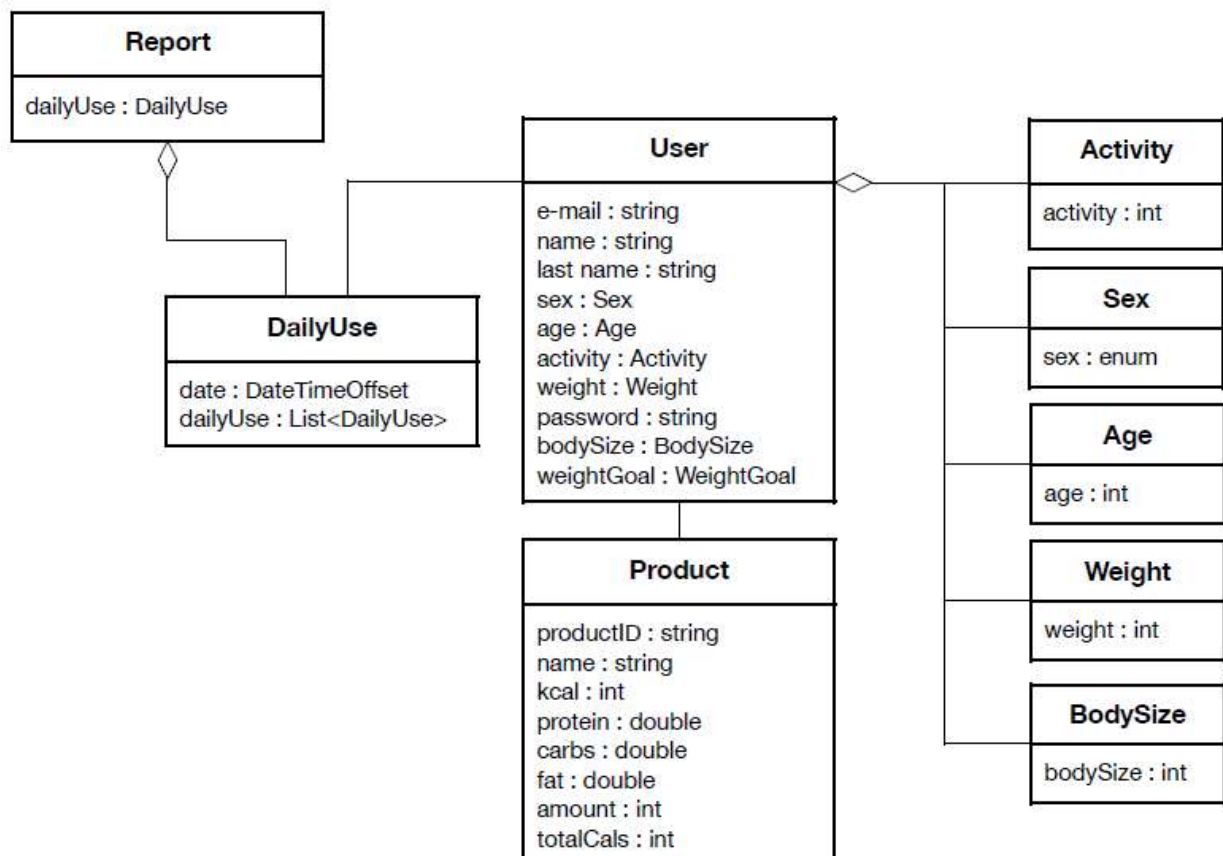
Spezifikation  
**NoPainNoGain**

## 4 Fachliches Datenmodell

### 4.1 ER - Modell



### 4.2 UML - Diagramm

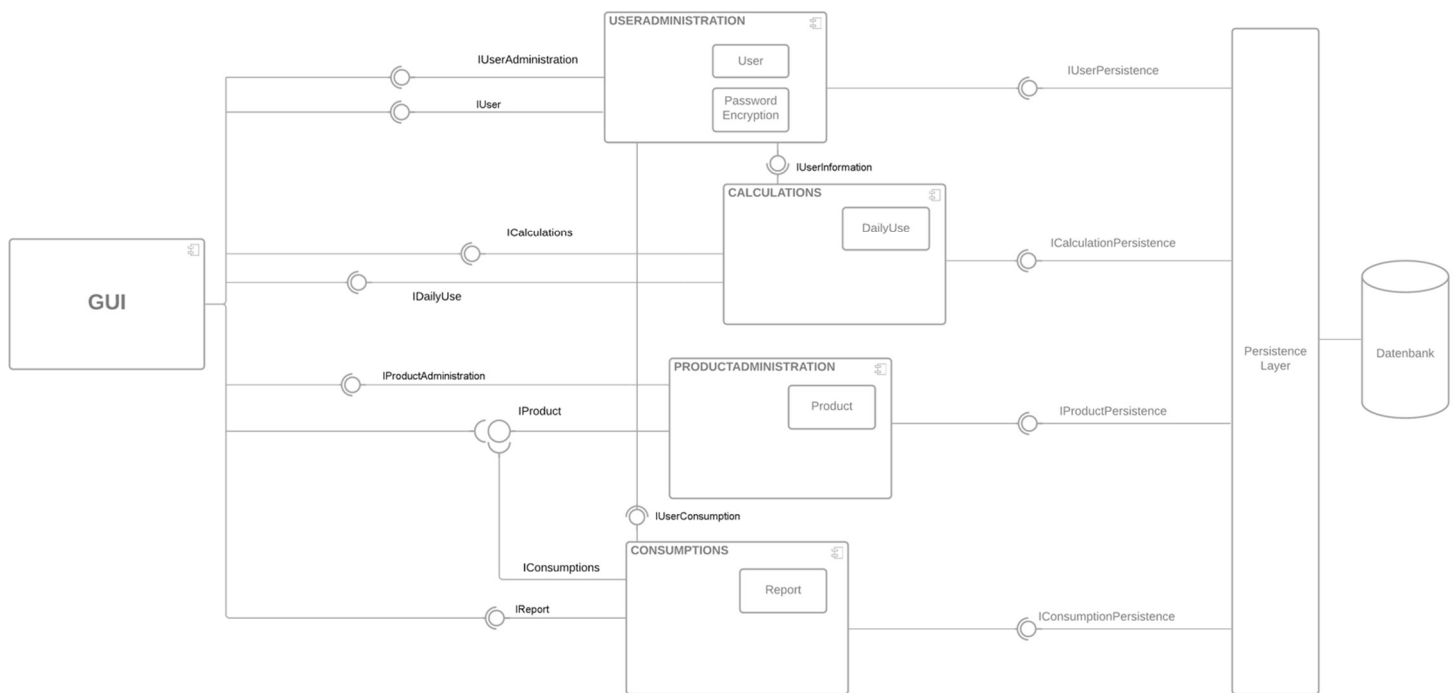




## Spezifikation **NoPainNoGain**

# 5 Komponenten

## 5.1 Komponentenstruktur



## 5.2 Interfacebeschreibung und Signaturen

### 5.2.1 IUser

- getUser(string): User
- setUser(var): User

### 5.2.2 IUserAdministration

- CreateUser(string, string, string, int): User
- UserSettings(User, int, int, int, string, int): User
- GetUser(string): User
- LoginUser(User): bool
- UpdatePassword(string): string

### 5.2.3 IUserInformation

- GetCalcInformation (int, int, int, string, string): int



Spezifikation

## **NoPainNoGain**

### **5.2.4 IUserPersistence**

- CreatorUser(User) : bool
- UpdateUser(string, User): bool
- DeleteUser(string, User): bool

### **5.2.5 ICalculation**

- createDailyCalculation(string, User): int
- calDifference(int): int
- weightCalculation(int, int): int

### **5.2.6 ICalculationPersistence**

- saveDailyCalories (int, User): bool
- reachDailyGoal(bool, User): bool
- saveweightCalucaltion(int, User): bool

### **5.2.7 IDailyUse**

- addProductToTheDay Product): List<Product>
- getDailyProductList(): List<Product>

### **5.2.8 IProductAdministration**

- createProduct(string, string, int, double, double, int, int): Product
- deleteProduct(int): bool
- addProduct(Product): bool

### **5.2.9 IProduct**

- getProductCalories(int): int
- getProduct(int): Product

### **5.2.10 IProductPersistence**

- saveProduct(Product): bool
- saveProductList(int, Product): List<Product>



Spezifikation

## **NoPainNoGain**

### **5.2.11 IConsumption**

- showConsumption(int): List<DailyUse>
- addToConsumption(Product): bool
- createReport(int, int): List<int, int>

### **5.2.12 IConsumptionPersistence**

- saveConsumption(List <DailyUse>): bool
- saveReport(Report): bool

### **5.2.13 IUserConsumption**

- GetReport(string, User): Report
- GetConsumption(string, User): List<Product>

### **5.2.14 IReport**

- createWeightStatistic(int, int): int
- getWeightCalculation(string): int
- getOldDailyUseList(int): List<Product>

## **6 Validierung**

Für das Testen aller Usecases und Entitäten verwenden wir GUI & Unit-Tests.

### **6.1 Test Login**

Beim Login soll getestet werden ob die Abfrage von E-Mail/Passwort korrekt sind und ob die Kommunikation mit der Datenbank funktioniert.

### **6.2 Test Profil erstellen**

Bei der Erstellung des Profils sollen Passwort und E-Mail auf verschiedenen Arten getestet werden. Die E-Mail darf nicht doppelt in der Datenbank abgespeichert werden und eine leere Angabe soll ausgeschlossen werden. Das Passwortfeld darf ebenfalls nicht leer sein und die Zeichen soll durch ‚\*‘ ersetzt werden.

### **6.3 Test Profil bearbeiten**

Dieser Test bezieht sich auf die Logik der Angaben im Usecase Profil bearbeiten. Es soll dafür jeweils realistische Beschränkungen geben um Extremwerte, wie bspw. ein Gewicht von 400 Kilo oder ein Alter



Spezifikation

## **NoPainNoGain**

von 130 Jahren zu unterbinden. Der Nutzer wird gegebenenfalls mit einer Fehlermeldung und einem Hinweis benachrichtigt.

### **6.4 Test Datenbank**

Ebenfalls ist es wichtig zu testen ob die Datenbank eine Verbindung aufweist und einen schreibenden/lesenden Zugriff ermöglicht. Außerdem muss die Übergabe der Daten in der Datenbank auch der im Code hinterlegten Codierung entsprechen. Dies soll durch Anlegen und Bearbeiten von Nahrungsmitteln erfolgen. Der Interaktionstest soll in folgende Schritte unterteilt werden:

1. Produkte erstellt werden können (100g Wert)
2. Produkte und Nutzerdaten in der Datenbank gespeichert werden können
3. Produkte und Nutzerdaten aus der Datenbank abgerufen werden können
4. Produkte und Nutzerdaten in der Datenbank verändert und neu gespeichert werden können
5. Neue Werte abgerufen und Informationen der Summe des Typs KCAL entnommen werden können

### **6.5 Test Kalorienbedarf bestimmen und verrechnen**

Der dritte Ablauf bezieht sich auf die Kalorienrechnung. In diesem Test wird es zwei Nutzer-Objekte geben, um zu ermitteln ob die Rechnung realistische Werte ausgibt.

TestP1 = Mann, 23 Jahre, 182 cm, 84 Kg, Faktor 1.5, Gewicht aufbauen (+300Kcal)

- ⇒  $66,5 + (13,75 \times 84) + (5,003 \times 182) - (6,775 \times 23) = 1976,221$
- ⇒  $1976,221 \times (1.5 - 1) = 988,1105$
- ⇒  $(1976,221 + 988,1105) + 250 = 3214 \text{ Kcal Pro Tag (gerundet)}$

TestP2 = Frau, 43 Jahre, 165 cm, 63 Kg, Faktor 1.8, Gewicht abnehmen(-300Kcal)

- ⇒  $655,1 + (9,563 \times 63) + (1,850 \times 165) - (4,676 \times 43) = 1361,751$
- ⇒  $1361,751 \times (1.8 - 1) = 1089,4008$
- ⇒  $(1089,4008 + 1361,751) - 300 = 2151 \text{ Kcal Pro Tag (gerundet)}$

Nachdem der 1.Test erfolgreich durchlaufen ist, geht es weiter mit der Verrechnung des Tageskonsum. Produkte, welche in der Datenbank gespeichert wurden, sollen durch eine Methode mit dem Tagesbedarf verrechnet werden und einen Saldo an den Nutzer zurückgeben. Dabei wird getestet ob die Methode in der Lage ist den Wert des Tagesbedarfs in korrekter Ausführung zu verändern und diesen an den Nutzer weitergeben und zu speichern.





Spezifikation

**NoPainNoGain**

## 7 Aufteilung der Aufgaben

### 7.1 Lucas Schunko

1. Datenbankanzbindung
2. Fachliche Datenmodelle (ER, UML)
3. Implementierung eines Teils der Usecases inkl. zugehöriger Testtreiber
4. Implementierung der persistenten Komponente

### 7.2 Katja Wiesmüller

1. GUI-Anbindungen mit WPF
2. GUI-Design
3. Implementierung eines Teils der Usecases inkl. Zugehöriger Testtreiber
4. Teil des Anwendungskerns

### 7.3 Marco Schmidl

1. Erstellung der Interfaces
2. Implementierung eines Teils der Usecases inkl. Zugehöriger Testtreiber
3. Teil des Anwendungskerns
4. Komponentenstruktur