

---

## Anforderungsdokument

---

### **Native Android – App für persönliche Fitnessassistenz mit individuellen Plänen und Zielen**



Stand:	02.12.2019
Autor:	Ferenc Horvay
Kurs:	TINF18B5
Zielgruppe:	Fitnessinteressierte

---

## Versionstabelle

<b>Versionsnummer:</b>	<b>Autor:</b>	<b>Änderungsvermerk</b>
1.0	Ferenc Horvay	Initiale Anforderung

## Inhaltsverzeichnis

<b>1. Einleitung .....</b>	<b>1</b>
1.1. Welches Problem soll gelöst werden? .....	1
1.2. Idee des Vorhabens .....	1
1.3. Welcher Nutzen wird von der Lösung erwartet .....	1
1.4. Was ist konkret zu tun? .....	1
<b>2. Anforderungserhebung .....</b>	<b>3</b>
2.1. Stakeholder Identifikation .....	3
2.2. Anforderungsszenarien .....	3
2.3. ELSI-Analyse.....	19
2.3.1. Ethische Aspekte .....	19
2.3.2. Soziale Aspekte .....	19
2.3.3. Rechtliche Aspekte .....	19
<b>3. Anforderungen abstimmen und Spezifizieren .....</b>	<b>20</b>
3.1. Systemkontext .....	20
3.2. Widersprüche / Konflikte.....	20
3.3. Übersicht Anwendungsfälle .....	21
3.4. Textuelle Dokumentation einzelner Anforderungen.....	22
<b>4. Systemdiagramme .....</b>	<b>29</b>
4.1. Systemkontext.....	29
4.2. Dynamische Sichten.....	30
4.3. Statische Sichten.....	35
<b>5. Ausblick .....</b>	<b>36</b>
5.1. Annahme .....	36
5.2. Grenzen.....	36
5.3. Potential .....	36
<b>6. Literaturverzeichnis .....</b>	<b>37</b>

## Abbildungsverzeichnis

Abbildung 1 UML Anwendungsfalldiagramm .....	21
Abbildung 2 Systemkontextdiagramm .....	29
Abbildung 3 UML Aktivitätsdiagramm zur Berechnung des BMIs .....	30
Abbildung 4 UML Aktivitätsdiagramm zur Erstellung eines Trainingsplanes .....	31
Abbildung 5 UML Aktivitätsdiagramm zum Löschen eines Trainingsplans .....	32
Abbildung 6 UML Aktivitätsdiagramm zur Akustischen Funktion Zeitbasierter Übungen .....	33
Abbildung 7 UML Aktivitätsdiagramm zum Speichern einer Übung.....	34
Abbildung 8 UML Klassendiagramm zur Superklasse Trainingsplan.....	35
Abbildung 9 UML Klassendiagramm zu einem Trainingsplan.....	35

## **Tabellenverzeichnis**

Tabelle 1 Glossar .....	2
-------------------------	---

# **1. Einleitung**

## **1.1. Welches Problem soll gelöst werden?**

Mehr und mehr Menschen verlieren aufgrund von ungesunder Ernährung sowie weniger Bewegung ihre Sportlichkeit oder legen an Gewicht zu. Ebenso kann ein Abonnement im Fitness Studio sehr viel Zeit und Geld in Anspruch nehmen. Hierzu soll im Rahmen des Projektes eine App entwickelt werden, welche durch von zu Hause aus ausführbaren Übungen, für welche man weder Geräte noch Gewichte braucht, einen passenden Trainingsplan ermitteln kann.

## **1.2. Idee des Vorhabens**

Die entstandene App soll einen Besuch im Fitnessstudio ablösen, indem es anhand eingegebener Körpermaße und Präferenzen einen passenden Trainingsplan erstellt. Dieser beinhaltet nur Übungen, welche mittels des eigenen Körpergewichtes absolviert werden können und somit auch weniger Zeit in Anspruch nehmen, als ein Fitnessstudiobesuch.

## **1.3. Welcher Nutzen wird von der Lösung erwartet**

Der User soll anhand der angelegten Übungen der App ermutigt werden an den ausgewählten Tagen Sport zu treiben und durch die individuelle Berechnung des Trainingsplans auch nicht überanstrengt zu werden. Somit soll die App den Benutzer nach und nach dabei Unterstützen sein Traumgewicht zu erreichen.

## **1.4. Was ist konkret zu tun?**

Es soll ein Programm entwickelt werden, welches anhand der eingegebenen Daten des Users speziell zugeschnittene Trainingspläne evaluiert.

## Glossar

<b>Begriff:</b>	<b>Erläuterung:</b>
<b>Soll</b>	Ein Soll-Zustand beschreibt einen Zustand den ein Objekt oder ein System annehmen soll aber derzeit nicht besitzt. Es handelt sich hierbei also mehr oder minder um eine Empfehlung eines Normgebers, die also nur ein begrenztes Ermessen einräumt.
<b>Muss</b>	Eine Muss-Vorschrift ist eine Vorschrift, die in jedem Fall eingehalten werden muss, die also kein Ermessen einräumt.
<b>Kann</b>	Eine Kann-Bestimmung ist eine Bestimmung, nach der im Einzelfall verfahren werden kann, aber nicht verfahren werden muss.
<b>Native App</b>	Mit einer nativen App ist eine Anwendung gemeint, welche speziell für das Betriebssystem des jeweiligen Endgerätes konzipiert und entwickelt wurde.
<b>BMI</b>	Der Body-Mass-Index ist ein grober Richtwert, mit der man das Körpergewicht eines Menschen im Verhältnis zu seiner Körpergröße bewerten kann.
<b>Trainingsplan</b>	Hiermit werden die Cardio-, Übungs- oder Muskelaufbaupläne zusammengefasst.
<b>UML</b>	Unified Modeling Language ist eine etablierte, objektorientierte, standardisierte und werkzeugunterstützte Modellersprache für die Visualisierung, Beschreibung, Spezifikation und Dokumentation von Systemen

Tabelle 1 Glossar

## **2. Anforderungserhebung**

### **2.1. Stakeholder Identifikation**

Die Stakeholder für die native Android Fitness App wurden durch ein Interview mit dem Auftraggeber, Literaturrecherchen und Interviews mit einem Fitnessinteressierten, Fitnessstudiomitglieds und Fitnesscoach identifiziert. Da die App für Anfänger sowie für erfahrene Sportler konzipiert wurde, ist keine spezielle Zielgruppe vorhanden.

Aus diesen Erfahrungen heraus wurde nur der Benutzer selbst als Stakeholder identifiziert

### **2.2. Anforderungsszenarien**

Die Folgenden Use-Cases (UC) wurden mithilfe folgender Methoden erhoben:

- Brainstorming
- Interview Fitnessstudiomitglied
- Interview Fitnesscoach
- Interview Fitnessinteressierte

Die Anforderungserhebungsmethode des Brainstormings ausgewählt, da sie den Vorteil hat, dass viele Ideen in kurzer Zeit niedergeschrieben werden konnten.

Die Erhebungsmethode des Interviews wurde ausgewählt, da hierdurch relativ viele Informationen in kurzer Zeit erfragt werden können und hierzu auch nicht-Sachverhalte wie Begründungen, Bewertungen oder Präferenzen besser spürbar sind als bei einer reinen Literaturrecherche.



UC-1	BMI berechnen
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, aus meinen angegebenen Daten einen BMI zu errechnen, damit klar ersichtlich ist, ob ich laut BMI untergewichtig, normalgewichtig oder übergewichtig bin.
Pfad	Benutzer gibt Körpergröße, Gewicht, Alter und Geschlecht ein → Anzeige BMI → Anzeige eines Textes, bezüglich des aktuellen Gesundheitsstatus
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Kein eingegebenes Gewicht → Keine BMI Berechnung (Standardwert)</li> <li>• Keine eingegebene Größe → Keine BMI Berechnung (Standardwert)</li> <li>• Keine eingegebenen Daten → Anzeige: Ohne eingegebene Daten wird ein BMI von 22 angenommen</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer kennt seine Körpergröße, Gewicht, Alter und Geschlecht</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• BMI wird angezeigt und gespeichert</li> </ul>

UC-2	Generierung Cardioplan
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, anhand von mir eingegebener Präferenzen und ausgerechneten BMI einen geeigneten Cardioplan zu erstellen, damit ich während meinen Übungen weder überanstrengt, noch unterfordert werde.
Pfad	Benutzer wählt Präferenzen aus → Erzeugung Cardioplan anhand eingegebener Präferenzen und berechnetem BMI → Anzeige Cardioplan
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>Keine Eingabe von Präferenzen → Cardioplan wird anhand von BMI und Standard Präferenzen erstellt</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eventuell schon Präferenzen zu bestimmten Übungen, wie beispielsweise Joggen</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Cardioplan wird angezeigt und gespeichert</li> </ul>

UC-3	Generierung Muskelaufbauplan
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, anhand von mir eingegebener Präferenzen und ausgerechneten BMI einen geeigneten Plan zum Aufbau von Muskeln zu generieren, damit ich gezielt einzelne Muskelgruppen trainieren kann.
Pfad	Benutzer wählt Präferenzen aus → Erzeugung des Muskelaufbauplans anhand eingegebener Präferenzen und berechnetem BMI → Anzeige Muskelaufbauplan
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>Keine Eingabe von Präferenzen → Muskelaufbauplan wird anhand von berechnetem BMI und Standard Präferenzen erstellt</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eventuell schon Präferenzen zu bestimmten Übungen, wie Beispielsweise Situps</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Cardioplan wird angezeigt und gespeichert</li> </ul>

UC-4	Generierung eines Eigenen Übungsplans
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, mir anhand der in der App hinterlegten Übungen meinen eigenen Übungsplan zu erstellen, damit ich meine eigenen Präferenzen in Richtung Muskelaufbau, sowie Ausdauer entfalten kann.
Pfad	Der Benutzer wählt die Erstellung eines eigenen Übungsplanes aus → Liste aller vorhandenen Übungen werden sortiert angezeigt (von Aufwärmen zu Ausdauer zu Muskelaufbau) → Der Benutzer wählt alle seine Übungen aus → Der Benutzer bestätigt seine Auswahl → Darstellung des neuen Übungsplanes
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer wählt gar keine Übungen aus → Übungsplan wird nicht erstellt</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer hat den Willen seinen eigenen Trainingsplan zu erstellen</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Übungsplan wird angezeigt und gespeichert</li> </ul>

UC-5	Löschen Trainingsplan
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, meinen erstellten Trainingsplan zu löschen, damit ich mir einen neuen Trainingsplan generieren lassen kann.
Pfad	Der Benutzer wählt seinen Trainingsplan aus → Der Benutzer möchte den Plan löschen → Der Benutzer bestätigt den Vorgang
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer bricht die Löschung bei der Bestätigung des Vorgangs ab → Trainingsplan wird nicht gelöscht</li> <li>• Der Benutzer hat noch keinen erstellten Trainingsplan → Keine Löschung möglich</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Es muss mindestens ein Trainingsplan vorhanden sein</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Der Trainingsplan wird von den Plänen entfernt</li> </ul>

UC-6	Generierung Gewichtsdiagramm
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, wöchentlich mein Körpergewicht festzuhalten, damit ich über einen längeren Zeitraum hinweg mein Trainingsziel nachverfolgen kann.
Pfad	Benutzer trägt aktuelles Gewicht ein → Eintragen in Tabelle → Anzeige eines Liniendiagramms
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>Keine Eingabe von Gewicht → Kein erzeugtes Liniendiagramm</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Benutzer besitzt eine Waage</li> <li>Benutzer benutzt immer die gleiche Waage</li> <li>Benutzer besitzt einen festen Wiegezeitpunkt in der Woche</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Liniendiagramm wird erzeugt und angezeigt</li> </ul>

UC-7	Akustische Signale für Zeitabhängige Übungen
Quelle:	Interview: Fitnessscoach
Satzschablone	Als Benutzer sollte das System fähig sein, mir bei zeitabhängigen Übungen ein Akustisches Signal zu Beginn und Ende einer Übung ausgeben, damit ich mich voll auf die Übung und nicht auf die App konzentrieren kann.
Pfad	Der Benutzer ist dabei eine zeitabhängige Übung durchzuführen → Ein Akustisches Signal bestimmt den Anfang der Übung → Ein Akustisches Signal bestimmt das Ende der Übung
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer hat seinen Ton auf stumm → Keine Wiedergabe des Akustischen Signales</li> <li>• Der Benutzer bricht die Übung vorzeitig ab → Manuelles Beenden der Übung</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer hat den Ton auf seinem Handy, auf dem die App ausgeführt wird, an</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Nachdem der Akustische Ton die Übung beendet hat, wird die nächste Übung gestartet</li> </ul>

UC-8	Kalorien Festhalten
Quelle:	Interview: Fitnessstudiomitglied
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, täglich meine aufgenommenen Kalorien/Mahlzeiten zu speichern, damit ich eine bessere Übersicht über meinen Kalorienhaushalt und damit meinen Trainingserfolg haben kann.
Pfad	Benutzer wählt verzehrte Mahlzeit aus → Benutzer gibt eine Menge in Gramm an → Berechnung und Speicherung der Kalorien
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Keine Eingabe Mahlzeiten → Keine Übersicht der Übrigen Kalorien des Tages → Keine Tipps zum schnelleren Abnehmen</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Benutzer weiß in etwa wie viel er gegessen hat</li> <li>• Die Daten (Zucker-, Fett- und Kaloriengehalt) der Mahlzeit sind bekannt</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Die aufgenommenen Kalorien des Tages werden angezeigt</li> <li>• Die „restlichen“ Kalorien des Tages werden angezeigt</li> <li>• Eine Evaluation der Aufgenommenen Fette, Zucker und Kalorien werden angezeigt</li> </ul>



UC-9	App bewerten
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, eine Bewertung für die App zu schreiben, damit ich den Entwicklern Feedback zur App geben kann.
Pfad	Der Benutzer wählt Bewertung aus → Der Benutzer wird auf Google Play Store weitergeleitet → Der Benutzer hinterlässt eine Rezension
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer befindet sich im Google Play Store unter der App → Der Benutzer hinterlässt eine Rezension</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer möchte seine Meinung über die App veröffentlichen</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Die Rezension wird im Google Play Store angezeigt</li> </ul>

UC-10	Tipps bekommen
Quelle:	Brainstorming
Satzschablone	Als Benutzer muss das System fähig sein, mir Tipps zum Sport und zu Mahlzeiten zu geben, da hierdurch der Trainingseffekt maximiert werden kann.
Pfad	Benutzer öffnet die App → App zeigt einen Tipp des Tages an
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer hat den Tipp bereits gesehen → nicht erneut anzeigen</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer öffnet die App</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Die App zeigt einen Tipp an</li> </ul>

UC-11	Erledigte Trainings eintragen
Quelle:	Brainstorming
Satzschablone	Als Benutzer muss das System mir die Möglichkeit bieten, in einem Kalender meine abgeschlossenen Trainingseinheiten einzutragen, damit ich einen Überblick über meine abgeschlossenen Tätigkeiten haben kann.
Pfad	Der Benutzer öffnet die App → Der Benutzer wählt Kalender aus → Der Benutzer wählt Tag aus → Der Benutzer trägt Übungen ein
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Benutzer möchte einen Überblick behalten</li> <li>Der Nutzer weiß wann er welche Übungen gemacht hat</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Der Kalender zeigt abgeschlossene Übungen an</li> </ul>

UC-12	Beispiele zu den Übungen anzeigen
Quelle:	Interview Fitnesscoach
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, ein Beispielbild oder Video zu jeder Übung sehen zu können, damit ich nachvollziehen kann, wie ich die Übung korrekt ausführe.
Pfad	Benutzer öffnet die App → Benutzer wählt Übungen aus → Benutzer wählt eine Übung aus → App zeigt ein Beispielbild
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer möchte genau wissen wie eine Übung funktioniert</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Ein Bild oder Video von der ausgewählten Übung wird angezeigt</li> </ul>

UC-13	Benachrichtigung bei Abwesenheit
Quelle:	Interview Fitnesscoach
Satzschablone	Als Benutzer sollte das System fähig sein mir Benachrichtigungen zu schicken, wenn ich seit längerer Zeit keine Übungen eingetragen habe, damit ich daran erinnert werde Sport zu treiben.
Pfad	Benutzer verwendet die App über längeren Zeitraum nicht → Benutzer erhält eine Benachrichtigung
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>Der Benutzer hat seine für die App Benachrichtigungen deaktiviert</li> <li>Der Benutzer hat alle Benachrichtigungen deaktiviert</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer möchte erinnert werden Sport zu treiben</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Eine anspornende Benachrichtigung wird angezeigt</li> </ul>

UC-14	Daten zurücksetzen
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, alle eingetragenen und generierten Daten zurückzusetzen.
Pfad	Einstellungen → Fortschritt zurücksetzen → Erneut bestätigen
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer hat noch keinen Fortschritt</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer möchte seine Daten nichtmehr in der App haben</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Alle Fortschritte werden zurückgesetzt</li> </ul>

UC-15	Trainingspläne anzeigen
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, alle erstellten Trainingspläne anzuzeigen, um diese verwalten zu können.
Pfad	Der Benutzer öffnet die App → Der Benutzer wählt die Übersichtsseite
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>• Der Benutzer hat noch keinen Trainingsplan erstellt</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>• Der Benutzer hat einen Trainingsplan erstellt</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>• Alle Trainingspläne werden angezeigt</li> </ul>

UC-16	Internetverbindung
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, die App auch ohne Internet starten zu können.
Pfad	Der Benutzer öffnet die App → Der Benutzer wählt die Übersichtsseite
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer hat die App bereits heruntergeladen</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Die App wird geöffnet und startet normal</li> </ul>

UC-17	Android Version
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System mir die Möglichkeit bieten, die App auch auf einem älteren Android Smartphone ausführen zu können, damit ich nicht auf das neuste Smartphone angewiesen bin.
Pfad	Der Benutzer lädt die App herunter → Der Benutzer öffnet die App
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eine Android Smartphone</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Die App wird geöffnet und startet normal</li> </ul>

UC-18	Benutzerfreundlichkeit
Quelle:	Brainstorming
Satzschablone	Als Benutzer sollte das System fähig sein, mich in der Höflichkeitsform anzureden.
Pfad	Der Benutzer lädt die App herunter → Der Benutzer öffnet die App
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eine Android Smartphone</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Alle Anreden in der App werden mit Herrn, Frau oder Sie beginnen</li> </ul>

UC-19	Robustheit
Quelle:	Brainstorming
Satzschablone	Als Benutzer muss das System fähig sein, auch bei falschen oder unvollständigen Eingaben nicht auszufallen.
Pfad	Der Benutzer gibt unvollständige Werte ein → Das System fällt nicht aus
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>/</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eine Android Smartphone</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Die App weist auf eine falsche oder unvollständige Eingabe hin</li> </ul>

UC-20	Look and Feel
Quelle:	Brainstorming
Satzschablone	Als Benutzer soll das System fähig sein, auf allen Android Smartphones oder Tablets die gleiche Ansicht zu haben, damit ich nicht auf ein Gerät beschränkt werde.
Pfad	Der Benutzer lädt die App auf seinem Handy und Tablet → Das System passt sich auf die jeweilige Bildschirmgröße an
Alternative-Pfade/ Ausnahmen	<ul style="list-style-type: none"> <li>Die App wird für das Tablet optimiert</li> </ul>
Vorbedingung	<ul style="list-style-type: none"> <li>Der Benutzer besitzt eine Android Gerät</li> </ul>
Nachbedingung	<ul style="list-style-type: none"> <li>Die App sieht auf beide Geräten gleich gut aus</li> </ul>

---

## **2.3. ELSI-Analyse**

### **2.3.1. Ethische Aspekte**

Offen ist hierbei die eigene Einstellung gegenüber der Frage, welche Bilder von Feminität/Maskulinität in unserer Gesellschaft toleriert und gefördert werden. Dabei ist zu betrachten, dass dem Benutzer ein gesunder Trainingsplan anhand seines berechneten BMIs vorgeschlagen wird, welcher nur einen definierten Körper, sowie eine sportliche Ausdauer als Ziel vorsieht.

### **2.3.2. Soziale Aspekte**

Hinsichtlich der sozialen Aspekte ist zu beachten, dass der Zugang zu Fitness Apps nicht für alle Bürger gewährleistet ist. Dabei sollte im Auge behalten werden, dass durch bestehende soziale Ungleichheiten die körperliche Ertüchtigung nur für einen Teil der Bürger leicht erhältlich und umsetzbar ist.

### **2.3.3. Rechtliche Aspekte**

Da in der App personenbezogene Daten wie der Name, die E-Mail und IMEIs gespeichert werden müssen, gilt demnach die DSGVO. Datenschutzrechtlich ist eine Verarbeitung von personenbezogenen Daten immer nur dann zulässig, wenn der Betroffene ausdrücklich eingewilligt hat.



### 3. Anforderungen abstimmen und Spezifizieren

#### 3.1. Systemkontext

Die native Fitness App wird auf dem lokalen Speicher des Benutzers ausgeführt, auf welcher auch die einzelnen Übungen sowie Bilder/Videos sind. Somit ist die einzige Schnittstelle, welche der User zum System hat die Benutzeroberfläche, da er hierüber steuert, was er sehen möchte.

#### 3.2. Widersprüche / Konflikte

Konflikte	Begründung	Lösung
UC-2, UC3, UC-4	In den Use Cases 2 und 3 wird eine Separation von Cardio- und Muskelaufbauplänen erwünscht, welcher aber durch das Erstellen eines eigenen Übungsplanes (siehe Use Case 4) zunichte gemacht wird.	Als Lösung könnte man hierbei den Cardio-, sowie den Muskelaufbauplan als „generisch erstellte“ Pläne kategorisieren und den Übungsplan aus Use Case 4 als „eigenen Trainingsplan“ kategorisieren. Bei der Erstellung eines Trainingsplans wird also erstmals gefragt, ob der Benutzer ein Trainingsplan generisch erstellen lassen möchte oder ihn selbst bestimmen möchte.

Weiter Widersprüche oder Konflikte zwischen einzelnen Use Cases konnten momentan nicht gefunden werden.

### 3.3. Übersicht Anwendungsfälle

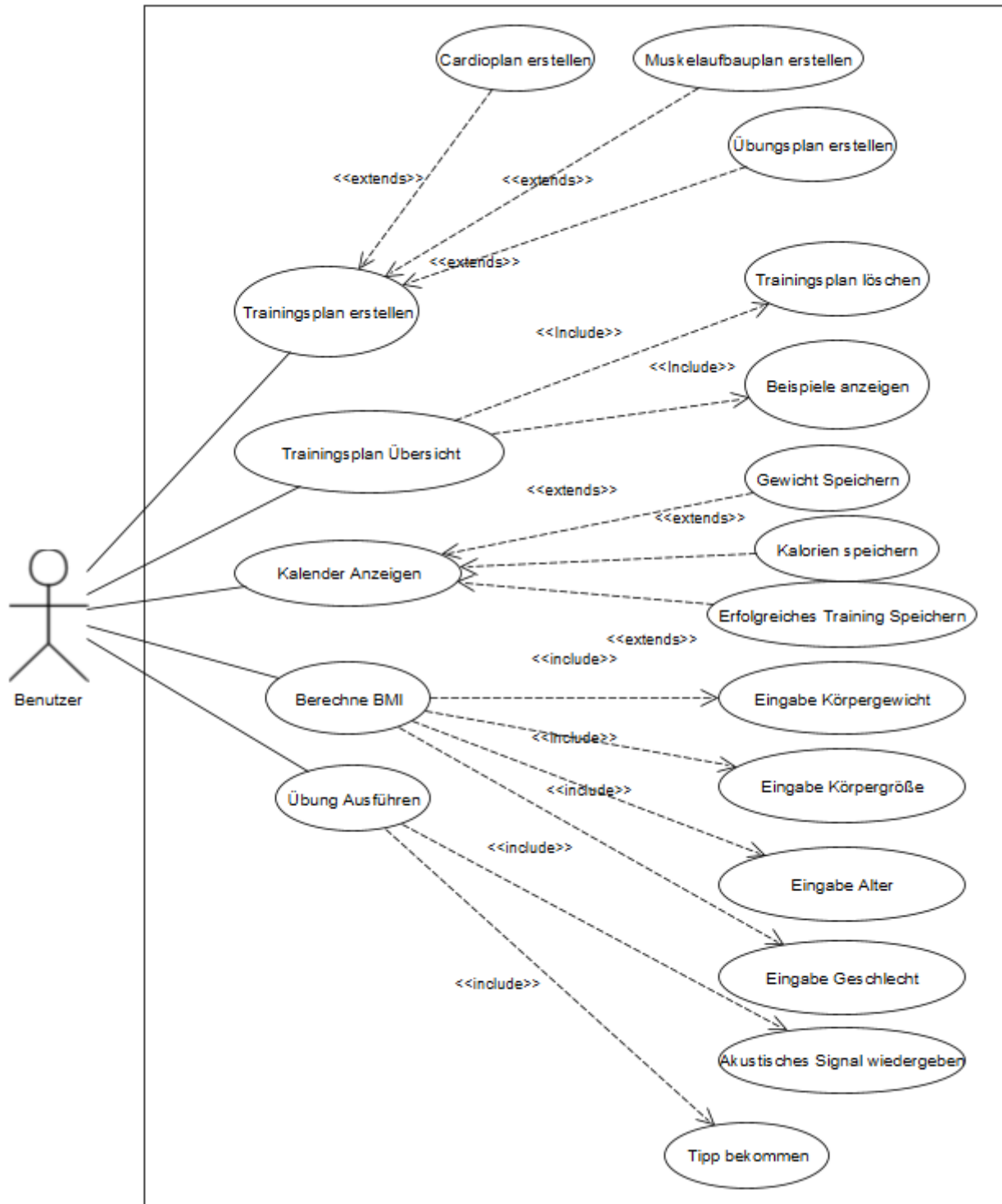


Abbildung 1 UML Anwendungsfalldiagramm

### 3.4. Textuelle Dokumentation einzelner Anforderungen

R1	<b>BMI berechnen</b>	Querbezüge: /
Setzte um: UC-1	Wenn der Benutzer seine Körpergröße, Alter, Geschlecht und Gewicht angegeben hat, muss das System in der Lage sein, aus diesen Daten einen BMI zu berechnen.	
	Dynamisch funktional	

R2	<b>Grafische Einteilung des BMI</b>	Querbezüge: R1
Setzte um: UC-1	Wenn das System den BMI des Benutzers errechnet hat, muss das System in der Lage sein, diesen jederzeit grafisch anzuzeigen.	
	Dynamisch funktional	

R3	<b>Information zur Genauigkeit des BMI Verfahrens</b>	Querbezüge: R1, R2
Setzte um: UC-1	Wenn der Benutzer seinen BMI grafisch sieht, muss das System in der Lage sein, eine Information zur Genauigkeit des BMI Verfahrens hinsichtlich körperlicher Gesundheit darzustellen.	
	Dynamisch funktional	

R4	<b>Festlegung eines Normalgewichtigen BMIs</b>	Querbezüge: R1
Setzte um: UC-1	Wenn der Benutzer keine Daten zu sich angibt, muss das System einen normalgewichtigen BMI von 22 festlegen.	
	Dynamisch funktional	

R5	<b>Information zu geschätzten BMI</b>	Querbezüge: R1, R4
Setzte um: UC-1	Wenn der Benutzer keine Daten zu sich angibt, muss das System eine Information anzeigen, dass von einem Standard BMI ausgegangen wird.	
	Dynamisch funktional	

R6	<b>Generierung Cardio- oder Muskelaufbauplan</b>	Querbezüge: /
Setzte um: UC-2, UC-3	Wenn der Benutzer einen Cardio- oder Muskelaufbauplan erstellen möchte, muss das System in der Lage sein, anhand des BMIs und der Präferenzen des Benutzers zu diesen zu erstellen.	
	Dynamisch funktional	

R7	<b>Information zum Cardio- Muskelaufbauplan</b>	Querbezüge: R6
Setzte um: UC-2, UC-3	Wenn ein Cardio- oder Muskelaufbauplan erstellt wurde, muss das System in der Lage sein einen Informationstext zu diesem Plan anzuzeigen.	
	Dynamisch funktional	

R8	<b>Erstellung eines eigenen Übungsplans</b>	Querbezüge: /
Setzte um: UC-4	Wenn der Benutzer einen eigenen Übungsplan erstellen möchte, muss das System dem Benutzer sortierte Listen von Aufwärm- Ausdauer- und Muskelaufbauübungen darstellen.	
	Dynamisch funktional	

R9	<b>Information zur Ausgewogenheit des Übungsplans</b>	Querbezüge: R8
Setzte um: UC-4	Wenn der Benutzer einen Übungsplan erstellt, muss das System dem Benutzer darauf hinweisen einen ausgewogenen Übungsplan zu erstellen.	
	Dynamisch funktional	

R10	<b>Verwerfen des Übungsplans</b>	Querbezüge: R8
Setzte um: UC-4	Wenn der Benutzer während der Auswahl der Übungen den Vorgang abbricht oder versucht ohne eine Auswahl den Plan fertigzustellen, muss das System eine Information ausgeben und ggf. den Plan verwerfen.	
	Dynamisch funktional	

R11	<b>Anzeigen aller Trainingspläne</b>	Querbezüge: /
Setzte um: UC-15	Das System soll dem Benutzer die Möglichkeit bieten, sich alle erstellten Übungspläne anzeigen zu lassen.	
	Dynamisch funktional	

R12	<b>Löschen eines Trainingsplans</b>	Querbezüge: R8, R6, R11
Setzte um: UC-5, UC-15	Das System soll dem Benutzer die Möglichkeit bieten, einen ausgewählten Trainingsplan wieder zu löschen.	
	Dynamisch funktional	

R13	<b>Generierung Gewichtdiagramm</b>	Querbezüge: /
Setzte um: UC-6	Wenn der Benutzer sein Gewicht eingegeben hat, soll das System in der Lage sein, anhand der eingegebene Daten ein Liniendiagramm zu erstellen.	
	Dynamisch funktional	

R14	<b>Zusammenfassung Abnehmerfolg</b>	Querbezüge: R13
Setzte um: UC-6	Wenn der Benutzer sein Gewichtdiagramm ansieht, soll das System in der Lage sein, anhand der eingegebenen Daten eine Übersicht über die abgenommenen Kilos, und dem zugehörigen Zeitraum darstellen.	
	Dynamisch funktional	

R15	<b>Akustisches Signal für Zeitabhängige Übungen</b>	Querbezüge: /
Setzte um: UC-7	Wenn der Benutzer eine zeitabhängige Übung ausführt, soll das System in der Lage sein, ein akustisches Startsignal und ein Akustisches Stoppsignal auszugeben.	
	Dynamisch funktional	

R16	<b>Vorzeitiges Beenden einer Zeitabhängigen Übung</b>	Querbezüge: R15
Setzte um: UC-7	Wenn der Benutzer eine zeitabhängige Übung ausführt, soll das System dem Nutzer die Möglichkeit bieten die Übung manuell, vor Ablauf der Zeit, zu beenden.	
	Dynamisch funktional	

R17	<b>Kalorien festhalten</b>	Querbezüge: /
Setzte um: UC-8	Wenn der Benutzer seine Kalorienzufuhr festhalten möchte, soll das System ihm die Möglichkeit bieten diese zu speichern.	
	Dynamisch funktional	

R18	<b>Mahlzeiten Überblicken</b>	Querbezüge: R17
Setzte um: UC-8	Wenn der Benutzer seine eingenommenen Mahlzeiten überblicken möchte, soll das System ihm die Möglichkeit bieten, diese über einen monatlichen Zeitraum anzeigen zu lassen.	
	Dynamisch funktional	

R18	<b>Verfärbung für viele Kalorien</b>	Querbezüge: R17
Setzte um: UC-8	Wenn der Benutzer seine eingenommenen Mahlzeiten überblickt, soll das System Tage, an denen viele Kalorien eingenommen wurden farblich markieren.	
	Dynamisch funktional	

R19	<b>App bewerten</b>	Querbezüge: /
Setzte um: UC-9	Wenn der Benutzer die App bewerten möchte, soll das System ihm die Möglichkeit bieten über einen Bewerten Button zum Google Play Store weitergeleitet zu werden.	
	Dynamisch funktional	

R20	<b>Tipps bekommen</b>	Querbezüge: /
Setzte um: UC-10	Wenn der Benutzer die App öffnet, soll das System dem Nutzer einen hilfreichen Tipp zum Sport oder Mahlzeiten geben.	
	Dynamisch funktional	

R21	<b>Tipps bekommen aus Hintergrundprozessen</b>	Querbezüge: 20
Setzte um: UC-10	Wenn der Benutzer die App aus den Hintergrundprozessen öffnet, soll das System keinen Tipp zum Sport oder zu Mahlzeiten geben	
	Dynamisch funktional	

R21	<b>Tipps bekommen deaktivieren</b>	Querbezüge: 20,21
Setzte um: UC-10	Wenn der Benutzer in den Einstellungen der App die täglichen Tipps deaktiviert hat, soll das System dem Nutzer keine Tipps beim Start der App geben.	
	Dynamisch funktional	

R22	<b>Kalender anzeigen</b>	Querbezüge:
Setzte um: UC-11	Das System soll dem Benutzer die Möglichkeit bieten einen Kalender anzuzeigen.	
	Dynamisch funktional	

R23	<b>Erledigte Trainings eintragen</b>	Querbezüge: R22
Setzte um: UC-11	Wenn der Benutzer im Kalender ist, soll das System ihm die Möglichkeit bieten ein erledigtes Training einzutragen.	
	Dynamisch funktional	

R24	<b>Abgeschlossene Trainings im Kalender hervorheben</b>	Querbezüge: R22, R23
Setzte um: UC-11	Wenn der Benutzer im Kalender ist, soll das System eingetragene Trainings farblich hervorheben.	
	Dynamisch funktional	

R25	<b>Beispiele zur Übung anzeigen</b>	Querbezüge: R11
Setzte um: UC-12	Wenn der Benutzer sich eine Übersicht der Übungen innerhalb seines Trainingsplans anschaut, soll das System in der Lage sein ihm hierzu nützliche Beispielbilder oder Beispielvideos zur Übung anzuzeigen.	
	Dynamisch funktional	

R26	<b>Benachrichtigung bei Abwesenheit</b>	Querbezüge: /
Setzte um: UC-13	Das System soll dem Benutzer jede Woche eine Benachrichtigung schicken, um daran erinnert zu werden Sport zu treiben.	
	Dynamisch funktional	

R27	<b>Daten Zurücksetzen</b>	Querbezüge: /
Setzte um: UC-14	Das System soll dem Benutzer die Möglichkeit bieten, alle Daten zurückzusetzen.	
	Nicht funktional	

R28	<b>Warnung beim Zurücksetzen</b>	Querbezüge: 27
Setzte um: UC-14	Wenn der Benutzer all seine Daten zurücksetzen möchte, soll das System dem Nutzer eine Warnung anzeigen, dass dies unwiderruflich ist.	
	Dynamisch funktional	

R29	<b>Benachrichtigung Trainingspläne anzeigen</b>	Querbezüge: /
Setzte um: UC-15	Wenn der Benutzer Trainingspläne erstellt hat, soll das System in der Lage sein alle erstellten Trainingspläne gruppiert anzuzeigen.	
	Dynamisch funktional	

R30	<b>Verfügbarkeit</b>	Querbezüge: /
Setzte um: UC-16	Das System soll in der Lage sein, auch ohne aktive Internetverbindung gestartet und benutzt werden zu können.	
	Nicht funktional	



R31	<b>Android Version</b>	Querbezüge: /
Setzte um: UC-17	Das System soll in der Lage sein, ab einer Android Version von 7.0 problemlos zu funktionieren.	
	Nicht funktional	

R32	<b>Benutzerfreundlichkeit</b>	Querbezüge: /
Setzte um: UC-18	Das System soll Benutzerfreundlich aufgebaut sein und den Benutzer mit der Höflichkeitsform ansprechen.	
	Nicht funktional	

R33	<b>Robustheit</b>	Querbezüge: /
Setzte um: UC-19	Das System soll gegenüber Fehleingaben geschützt sein.	
	Nicht funktional	

R34	<b>Look and Feel</b>	Querbezüge: /
Setzte um: UC-20	Das System soll die Oberfläche auf alle Bildschirmgrößen aktiv anpassen.	
	Nicht funktional	

## 4. Systemdiagramme

### 4.1. Systemkontext

Die App ist so ausgelegt, dass sie auf einem Smartphone läuft und erstmals keine Aktive Internetverbindung während der Benutzung bestehen muss. Alle Daten werden auf dem lokalen Speicher des Endgeräts gespeichert. Neue Übungen, Tipps und Tricks oder generelle Updates können im dann im Nachhinein über Updates bezogen werden.

Schnittstellen zum System stellen hierbei also nur der Touchscreen, sowie die Benutzeroberfläche der App da.

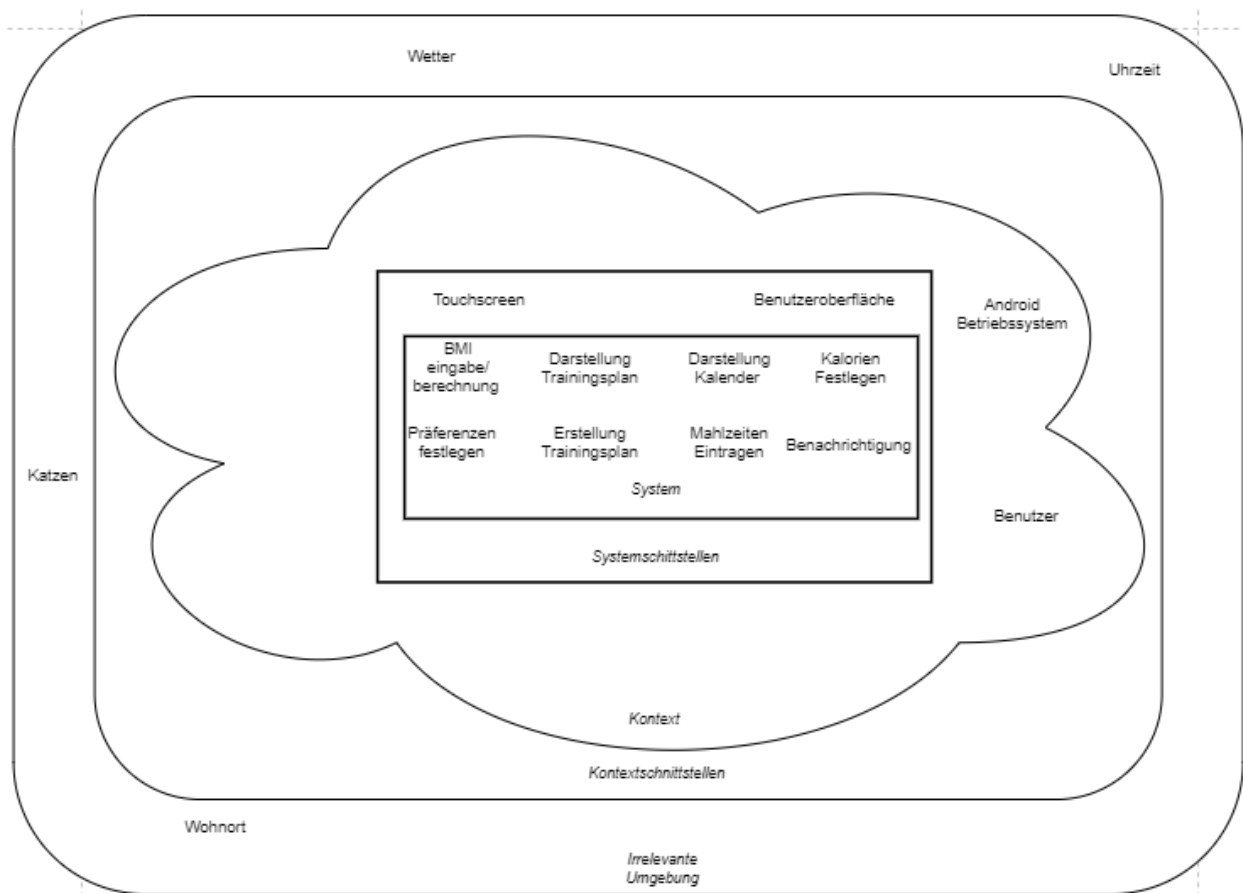


Abbildung 2 Systemkontextdiagramm

## 4.2. Dynamische Sichten

Nachfolgend werden Dynamische Sichten innerhalb verschiedener Vorgänge der App gezeigt, auf die der Benutzer Einfluss hat.

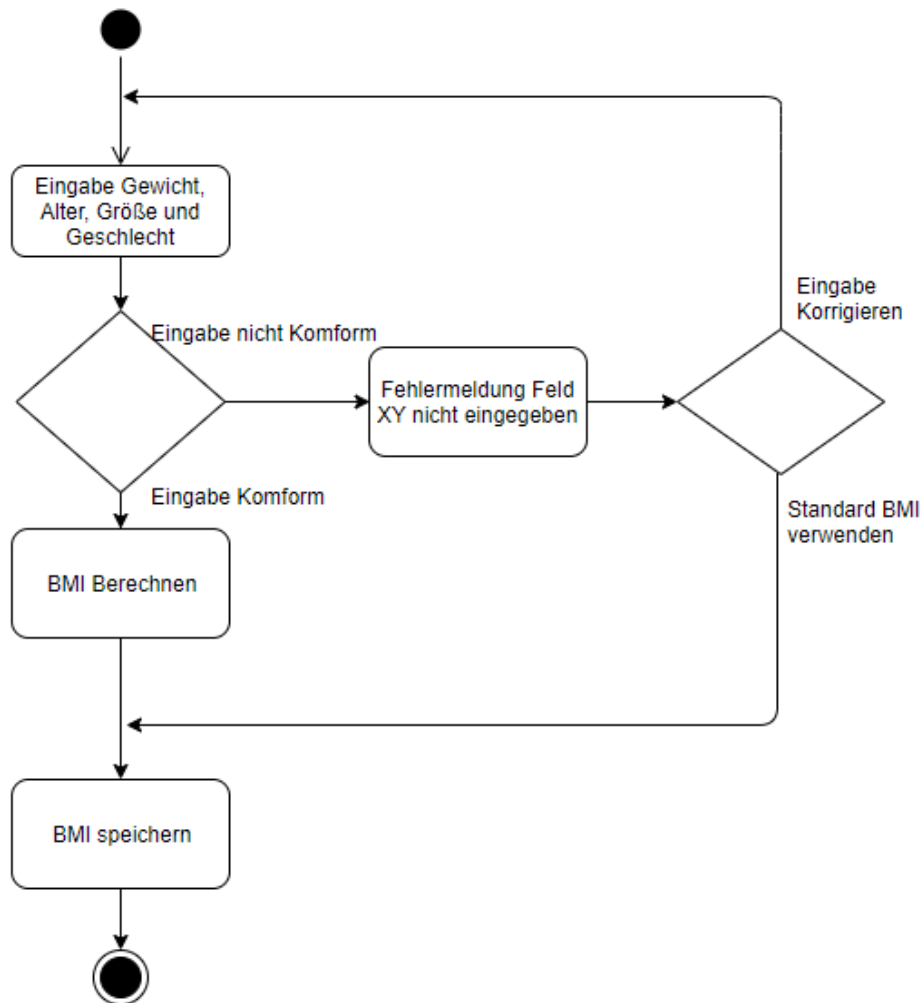


Abbildung 3 UML Aktivitätsdiagramm zur Berechnung des BMIs

Abbildung 3 stellt den Ablauf der BMI Berechnung dar, wie man dem Diagramm entnehmen kann, ist es auch möglich von einem Standard BMI auszugehen, wie er in R4, bzw. UC-1 beschrieben wurde.

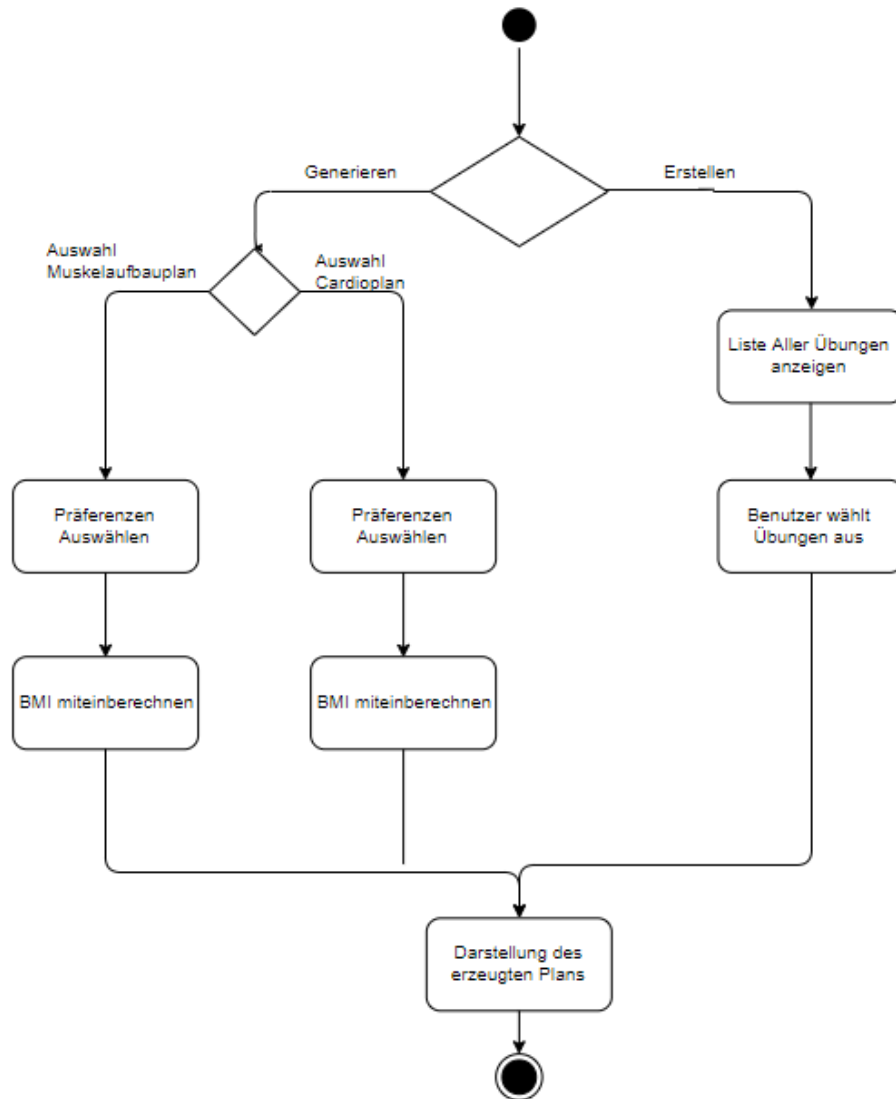


Abbildung 4 UML Aktivitätsdiagramm zur Erstellung eines Trainingsplanes

Abbildung 4 zeigt die Erstellung eines Trainingsplans. Hierbei kann der Benutzer frei entscheiden, ob er diesen selbst bestimmen möchte, oder ob er vom System generiert werden soll (Siehe R6, R8, UC-2, UC-3, UC-4).

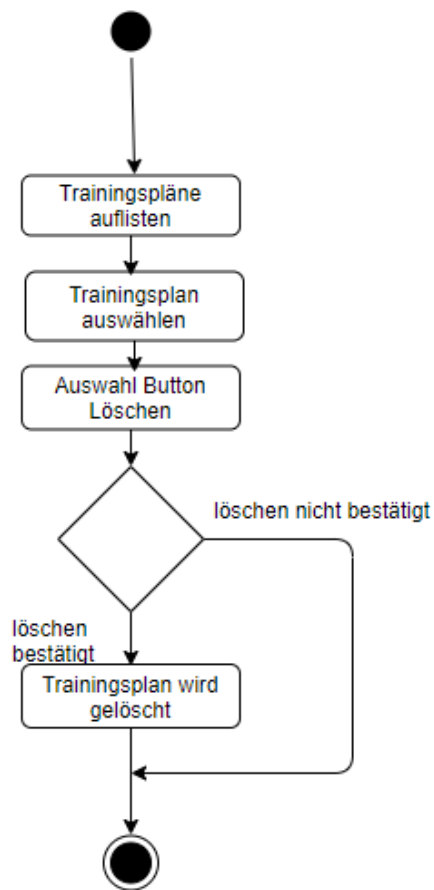


Abbildung 5 UML Aktivitätsdiagramm zum Löschen eines Trainingsplans

Abbildung 5 zeigt den Ablauf, welcher beim Löschen eines Trainingsplans durchgegangen wird (Siehe R12, UC-5, UC-15).

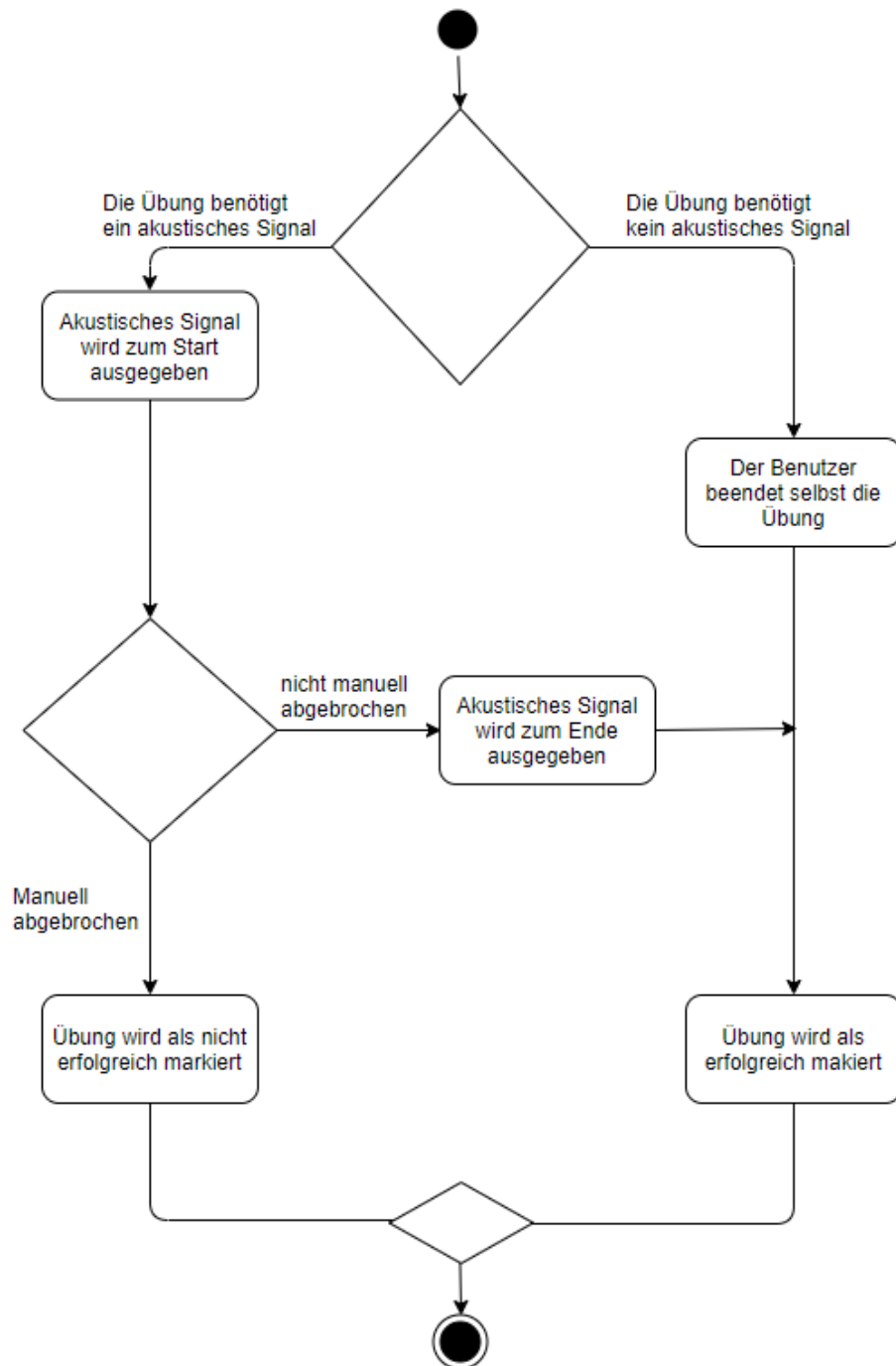


Abbildung 6 UML Aktivitätsdiagramm zur Akustischen Funktion Zeitbasierter Übungen

Abbildung 6 zeigt den Ablauf einer Zeitabhängigen Übung (Siehe R15, UC-7).

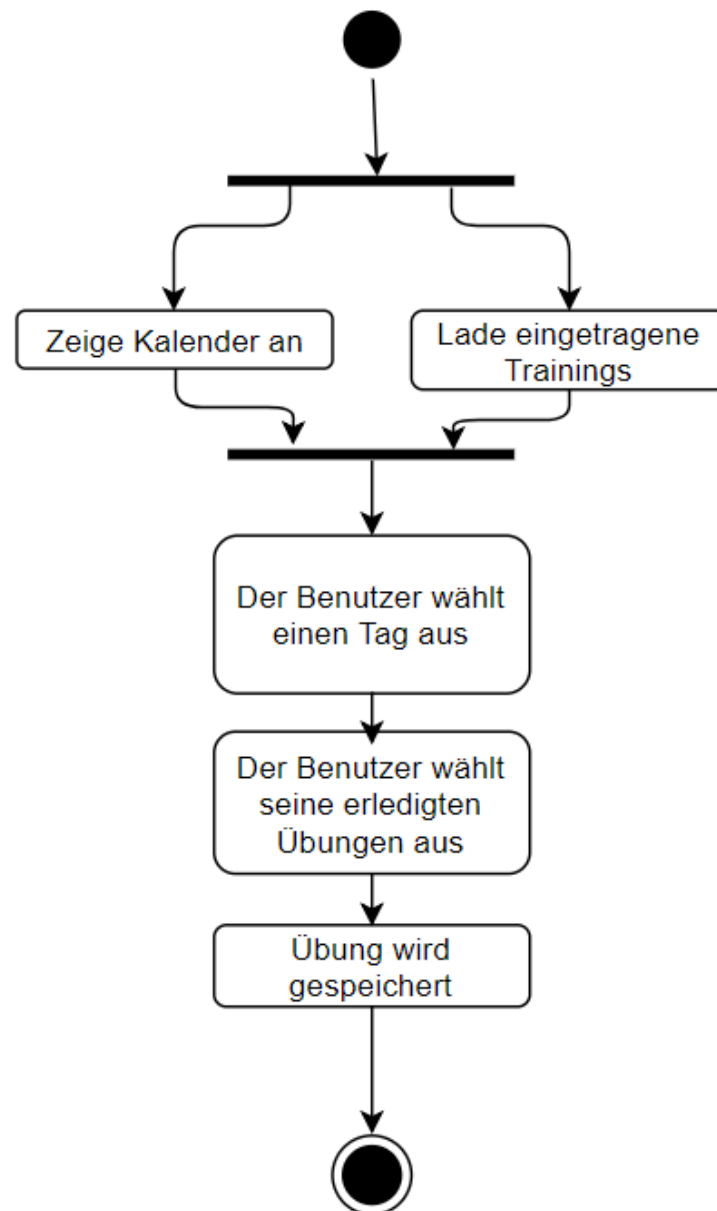


Abbildung 7 UML Aktivitätsdiagramm zum Speichern einer Übung

Abbildung 7 zeigt welche Schritte beim Eintragen einer Übung in den Kalender durchgangen werden (Siehe R23, UC-11)

### 4.3. Statische Sichten

Da die Anwendung mit mehreren Klassen arbeitet, werden diese in den nachfolgenden Klassendiagrammen genauer beschrieben.

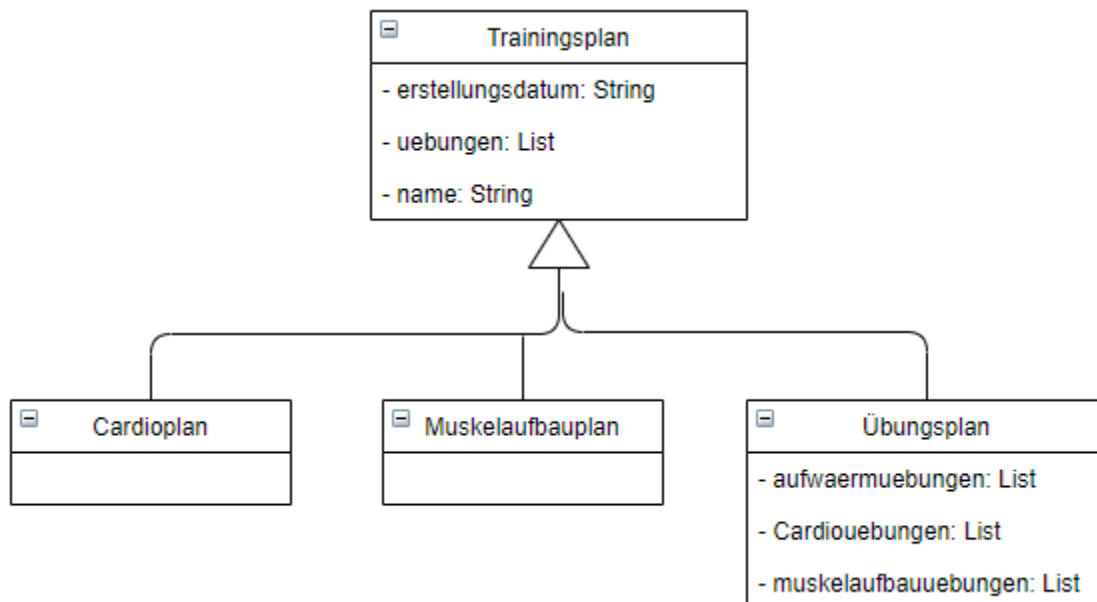


Abbildung 8 UML Klassendiagramm zur Superklasse Trainingsplan

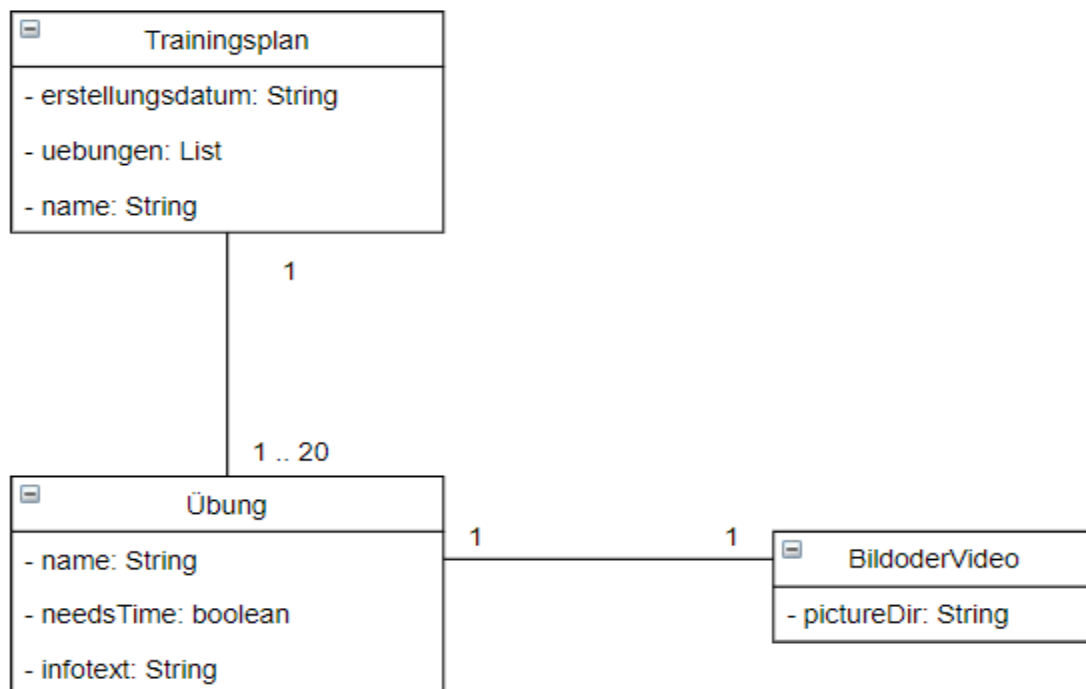


Abbildung 9 UML Klassendiagramm zu einem Trainingsplan



## **5. Ausblick**

### **5.1. Annahme**

Angenommen wird das die App nur von einem Benutzer pro Endgerät verwendet wird.

### **5.2. Grenzen**

Da die App keine wirklichen Kontrollmechanismen besitzt, um zu überprüfen, ob der Benutzer wirklich alle Übungen durchgeführt hat, wird hierbei auf die Ehrlichkeit des Benutzers gesetzt. Da es aber weder dem Benutzer, noch dem Entwickler der App einen Vor- oder Nachteil bringt, wenn der Benutzer seine Daten nicht wahrheitsgemäß angibt, ist dieser Punkt auch in gewisser Weise wieder zu vernachlässigen.

### **5.3. Potential**

Um die Problematik der Begrenzung von einem Benutzer pro Endgerät zu beheben, könnte in nachfolgenden Versionen eine Userverwaltung integriert werden, welche mehrere Profile innerhalb einer App ermöglichen würde.

Dadurch könnte jeder Benutzer seinen eigenen BMI berechnen lassen sowie eigene Trainingspläne erstellen können.

## 6. Literaturverzeichnis

[https://www.uni-salzburg.at/fileadmin/multimedia/Zentrum\\_fuer\\_Ethik\\_und\\_Armutsforschung/RV.AngewandteEthik.2015/Schweiger.RV.Sportethik.pdf](https://www.uni-salzburg.at/fileadmin/multimedia/Zentrum_fuer_Ethik_und_Armutsforschung/RV.AngewandteEthik.2015/Schweiger.RV.Sportethik.pdf)

<https://www.datenschutzexperte.de/blog/datenschutz-im-alltag/wearables-und-fitness-apps-datenschutzrisiko/>

[https://www.uni-muenchen.de/forschung/news/2018/wild\\_mhealth.html](https://www.uni-muenchen.de/forschung/news/2018/wild_mhealth.html)

<https://creately.com/diagram/example/iojq506v2/Fitness%20Tracker%20Use%20Case%20Diagram>

[https://cdn.studydrive.net/d/prod/flashcards/f02d9bcf784dfd46a4f137695771d1b6/LARGE\\_2245342155c5c40c97d4796.41187872.phpHETYSx.JPG](https://cdn.studydrive.net/d/prod/flashcards/f02d9bcf784dfd46a4f137695771d1b6/LARGE_2245342155c5c40c97d4796.41187872.phpHETYSx.JPG)

[http://dbis.eprints.uni-ulm.de/1078/1/BA\\_AliyaAras2014.pdf](http://dbis.eprints.uni-ulm.de/1078/1/BA_AliyaAras2014.pdf)

[https://de.wikipedia.org/wiki/Muss-,\\_Soll-\\_und\\_Kann-Vorschrift](https://de.wikipedia.org/wiki/Muss-,_Soll-_und_Kann-Vorschrift)

<https://i.pinimg.com/originals/d9/dd/32/d9dd32b8dfc8aa19e0ce30d8f43305d2.png>