Projektarbeit

zum Thema

PersonalFit

Kurzfassung

[Eine Zusammenfassung der Arbeit in deutscher Sprache (ist die Arbeit selbst nicht in deutscher Sprache kann dies wegfallen) - Umfang: 1/2 bis 1 Seite]

Abstract

[Eine zusätzliche Zusammenfassung der Arbeit in englischer Sprache - Umfang: 1/2 bis 1 Seite]

Inhaltsverzeichnis

[Abbildungsverzeichnis V](#_Toc490941493)

[Tabellenverzeichnis VI](#_Toc490941494)

[1 Einleitung 1](#_Toc490941495)

[1.1 Projektbeschreibung 1](#_Toc490941496)

[1.2 Projektplanung 1](#_Toc490941497)

[1.2.1 Aufgabenverteilung 1](#_Toc490941498)

[1.2.2 Meilensteine 1](#_Toc490941499)

[1.2.3 Herausforderungen 1](#_Toc490941500)

[2 Analyse bestehender Fitness-Apps 2](#_Toc490941501)

[3 Projektübersicht 3](#_Toc490941502)

[3.1 Zielgruppe 3](#_Toc490941503)

[3.2 Anforderungen 5](#_Toc490941504)

[3.2.1 User-Stories 5](#_Toc490941505)

[3.2.2 Funktionale Anforderungen 5](#_Toc490941506)

[3.2.3 Nicht-funktionale Anforderungen 7](#_Toc490941507)

[3.2.4 Data Dictionary 8](#_Toc490941508)

[3.3 Modelle 14](#_Toc490941509)

[3.3.1 Sitemap 14](#_Toc490941510)

[3.3.2 UML-Klassendiagramm 15](#_Toc490941511)

[3.4 Prototypen 15](#_Toc490941512)

[3.4.1 Papierprototypen 17](#_Toc490941513)

[3.4.2 Wireframes 18](#_Toc490941514)

[3.4.3 Mock-Ups 28](#_Toc490941515)

[3.5 SWOT-Analyse 42](#_Toc490941516)

[4 Zusammenfassung & Fazit 43](#_Toc490941517)

[5 Ausblick 44](#_Toc490941518)

[Literaturverzeichnis 45](#_Toc490941519)

[6 Literatur 47](#_Toc490941520)

Abbildungsverzeichnis

Abbildung 1: Zusammenfassung der Auswirkungen von körperlicher Aktivität auf die Gesundheit 4

Abbildung 2: Sidemap 14

Abbildung 3: Sitemap 15

Abbildung 4: UML-Klassendiagramm 15

Abbildung 5 Aufmachung und Start 18

Abbildung 6: Authenfication 30

Abbildung 7: iOS Feedback 31

Abbildung 8: Color 32

Abbildung 9:Interaktive und Nicht-Interaktive Farben 32

Abbildung 10 Layouts 33

Abbildung 11: IOS Text Styles 34

Abbildung 12 Dynamic Typ 35

Abbildung 13 Scroll Views 35

Abbildung 14 Beispiele System Icons 36

Abbildung 15 Scroll Views 37

Abbildung 16 Beispiel Label 39

Abbildung 17 Beispiel Sliders 39

Abbildung 18 Steppers 40

Abbildung 19 Beispiel Switches 41

Abbildung 20 Text Fields 42

Tabellenverzeichnis

Tabelle 1: Data Dictionary 9

# Einleitung

...

## Projektbeschreibung

...

## Projektplanung

...

### Aufgabenverteilung

...

### Meilensteine

...

### Herausforderungen

...

# Analyse bestehender Fitness-Apps

...

# Projektübersicht

...

## Zielgruppe

Die Studie „Gesundheit in Deutschland aktuell 2012“ von GEDA hat gezeigt, dass in Deutschland 36% der Erwachsenen keinen Sport ausüben. Unabhängig von dem Geschlecht und Bildungsgruppe erhöht sich die Prozentzahl, die nicht Sport machen ab 30 Jahre um das doppelte im Gegensatz zum Alter von 18-29 Jahren (GEDA, S.3). Außerdem wurde bewiesen, dass bereits junge Menschen zwischen 18 und 29 Jahren durchschnittlich 195 Minuten pro Tage auf der Arbeit sitzen. Je höher die Altersgruppe ist, desto mehr verringert sich die Anzahl der Minuten ein wenig. Allerdings steigt bei den höheren Altersgruppen die Anzahl des Fernsehens (Prof. Dr. Ingo Froböse, Dr. Birgit Wallmann-Sperlich, S.28). Sodass daraus gefolgt werden kann, dass die Erwachsenen ab 30 Jahren zu wenig in Bewegung sind. Dazu wächst der Anteil der Inaktivität mit dem Alter stark an (Pahmeier 2008, S. 170). Durch die körperliche Inaktivität erhöht sich der gesundheitliche Risikofaktor. Ein Viertel der Bevölkerungen haben eine chronische Erkrankung, die durch mangelnde Bewegung entstanden ist. Die chronische Erkrankung kann teilweise zu Todesfällen führen. Wenn die Hälfte der inaktiven Menschen körperlichen Aktivitäten nachgehen würden, könnte die Anzahl von ungefähr 6.500 Todesfälle durch Herz-Kreislauf-Problemen pro Jahr in Deutschland verringert werden. Demzufolge hat unregelmäßige Bewegung negative Auswirklungen auf den Körper, da dadurch das Risiko an Krankheiten und Beschwerden erhöht wird. In der Tabelle werden die Hauptauswirkungen geschildert.

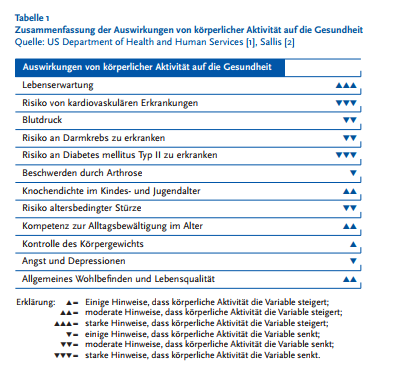


Abbildung : Zusammenfassung der Auswirkungen von körperlicher Aktivität auf die Gesundheit

Bewegung und körperliche Aktivitäten führen zu einer höheren Lebensqualität und fördert die Gesundheit. Auch das physische und mentale Wohlbefinden kann durch mehr Bewegung erhöht werden. Zum Beispiel Depressions-Erkrankten können durch Sport ihr Selbstvertrauen stärken.

Für bisher inaktive Erwachsene und ältere würde ein „Strammes Spazieren“ ausreichen, um das Verletzungsrisiko zu verringern (Robert Koch Institut Statisches Bundesamt, S. 1, 7, 13). Die Ergebnisse der GEDA Studie 2012 „Gesundheit in Deutschland 2012“ hat bewiesen, dass die gesundheitlichen Einschränkungen mit dem Alter wachsen und ab 65 Jahre besonders hoch sind. (GEDA, S.3) Generell wird empfohlen mindestens 30 Minuten täglich Sport zu machen, um sein Gesundheit zu fördern(Robert Koch Institut Statisches Bundesamt, S. 13).

Auf Grund dieser Erkenntnisse wird die Zielgruppe auf die Altersgruppe der 35-65-jährigen unabhängig ihres Geschlechtes und besonders auf Arbeitnehmer, die eine Bürotätigkeiten ausführen und denen wenig Freizeit zur Verfügung steht, eingeschränkt. Weitere Kriterien sind zum einen, dass sie der deutschen Sprache mächtig sind und mobile Applikation benutzen sowie körperlich nicht erheblich eingeschränkt sind. Zum anderen wird vorausgesetzt, dass die Zielgruppe keine Leistungssportler und Fortgeschrittene ausschließt, sondern sich auf die Anfänger bzw. sportlich inaktive Personen richtet.

## Anforderungen

In der Systementwicklung ist es wichtig Anforderungen, Dienste und Bedingungen zu beschreiben, um ein Problem zu lösen oder das Ziel zu erreichen. Anforderungen legen die Eigenschaften für das Verhalten des Systems fest, die dem Benutzer zur Verfügung gestellt werden. Bei der Beschreibung der Anforderung müssen folgende Fragen beantwortet werden „Was tut das System?“ und „Was soll es aufgrund der Aufgabenstellung können?“. Im Folgenden werden Anforderungen mit Hilfe von unterschiedlichen Instrumenten spezifiziert.

### User-Stories

...

### Funktionale Anforderungen

Die Funktionalen Anforderungen sind ein wichtiger Bestand der Spezifikation der Anforderungen, denn sie beschreiben die Funktionen, die vom betrachteten System erwartet werden. Hierbei werden mehre Beschreibungs-Aspekten berücksichtigt. Zum einen die Eingaben und die zugehörigen Einschränkungen als vordefinierte Funktionsabläufe, die das System aus der Sicht des Benutzers bzw. der Systemumgebung beschreibt. Ein weiterer Aspekt ist die Ausgabe. All diese Aspekte weisen einen bestimmten Algorithmus auf.

Die Beschreibung der Funktionalen Anforderungen umfasst die Dienste und Funktionalitäten. Die einfachste Form einer Spezifikation ist die Anwendung der natürlichen Sprache. Die natürliche Sprache ist eine weit verbreitete Technik und somit für jeden verständlich. Die Strukturierung durch Nummerierung- und Gliederungsschemata klärt Details und veranschaulicht die Zusammenhänge. Deshalb ist es wichtig, dass die Anforderungen prägnant, einfach und soweit wie möglich zu untergliedern. Zusätzlich unterstützt dies die Systementwicklung für alle Beteiligten, da alle weiteren Entwürfe auf das dieser Spezifizierung aufbaut. Allerdings können durch starke Vereinfachung und Gliederung es schnell unübersichtlich werden und zu Missverständnissen führen.

Nach diesem Schema wurden die Anforderungen dokumentiert:

1. Benutzereinrichtung
   1. Der Benutzer wird nach seinem Vornamen gefragt.
   2. Der Benutzer gibt Auskunft über sein „Geschlecht“, „Alter“, „Größe“, „Gewicht“ und „Körperumfang“.
   3. Der Benutzer gibt seine individuelle Trainingshäufigkeit an.
   4. Der Benutzer gibt an, ob er eine Trainingserinnerung von dem System erhalten möchte.
   5. Der Benutzer wählt eine Liste von unterstützen Trackinggeräten diejenige aus, die er besitzt.
   6. Alle getätigten Eingaben können zum beliebigen Zeitpunkt in den Einstellungen verändert werden.
2. Fitnesskategorien
   1. Das System stellt die Fitnesskategorien „Fokus“, „Kraft“, „Balance“, („Stressreduktion“) und „Straffen“ dem Benutzer bereit.
   2. Der Benutzer sucht sich mindestens eine Fitnesskategorie aus. Eine maximale Beschränkung ist nicht vorhanden.
3. Levelsystem
   1. Das Levelsystem besteht aus .. untergeordneten Leveln.
   2. Ein Level wird erhöht, sobald eine gegebene Anzahl an Workouts abgeschlossen wurde.
   3. Nach jedem Level werden fortgeschrittene Übungen freigeschaltet.
4. Einstufungstest
   1. Das System bietet dem Benutzer zu Beginn einen Einstufungstest in den Kategorien „Kraft“, „Balance“ und „Straffen“ an, um sein Fitnesslevel zu identifizieren.
   2. Die Dauer des Einstufungstestes wird vorab angezeigt.
   3. Der Benutzer hat die Möglichkeit nach jedem Level den Einstufungstest zu wiederholen.
5. Trainingsplan
   1. Workout
      1. Das System gibt dem Workout eine bestimmte Reihenfolge der Übungen vor.
      2. Ein Workout besteht aus ? verpflichtende Übungen und ? optionalen Übungen
      3. Das System stellt dem Benutzer eine Reihe von optionalen Übungen dar, die er selbst aus einer Liste von vorgegebenen zusammenstellen kann.
      4. Ein Workout ist abgeschlossen, sobald der Benutzer alle verpflichtende Übungen als auch ein Teil der optionalen Übungen beendet hat.
      5. Wird das angefangene Workout nach einem Tag nicht vollständig beendet, bekommt der Benutzer eine Benachrichtigung.
   2. Übungen
      1. Das System stellt dem Benutzer zu der jeweiligen Übung eine Anleitung bereit.
         1. Die Anleitung besteht aus einem Video, einer Beschreibung in Textform und eine skizzierte Veranschaulichung der jeweiligen beanspruchten Muskelpartien.
      2. Die Übungen passen sich dem jeweiligen Level an, in dem der Benutzer sich befindet.
      3. Das System empfiehlt dem Benutzer zu der jeweiligen Übung eine genaue Anzahl an Sätzen, Wiederholungen, Häufigkeit und Dauer der Pause zu machen.
      4. Das System erfasst, wenn möglich, die erledigte Übungen und kennzeichnet sie.
      5. Der Benutzer kann seine abgeschlossenen Übungen manuell makieren.
6. Dashboard
   1. Das System veranschaulicht kurz und prägnant von den ausgewählten Statistiken einen ausgewählten Parameter an.
   2. Jedes ausgewählte Ziel stellt das System einzeln dar.
   3. Das System zeigt zu jedem ausgewählten Ziel auf dem Dashboard den „Namen“ des ausgewählten Ziels, „Stand des Levels“, „Häufigkeit“ an.
7. Statistiken
   1. Das System stellt dem Benutzer eine Auswahl von Statistiken, um seinen Fortschritt in den Bereichen „Kalorienverbrennung“, „Route“, „Pulsmesser“, „Schritte“, „Gewicht“, „Brust-, Hüft-, Oberarm-, Taillen-, Oberschenkelumfang“ einzusehen.
      1. Die „Kalorienverbrennung“, „Route“, „Pulsmesser“ und „Schritte“ werden eigenständig vom System bzw. vom Trackinggerät erfasst und abgespeichert.
      2. Bei „Gewicht“, „Brust-, Hüft-, Oberarm-, Taillen-, Oberschenkelumfang“ hat der Benutzer die Möglichkeit manuell seine Daten einzugeben.
   2. Der Benutzer kann manuell seine angezeigten Diagramme auswählen und hinzufügen.
   3. Die Werte sind in unterschiedlichen Skalierungen im Zeitlauf darzustellen.

### Nicht-funktionale Anforderungen

...

### Data Dictionary

In der Softwareentwicklung sind verständliche, verlässliche, überprüfbare und letztendlich korrekte Daten von höchster Wichtigkeit. Nur wenn die relevanten Daten einer Domäne auf angemessene Weise in digitaler Form abgebildet werden können, ist die korrekte Funktionsweise technischer Informationssysteme möglich.

Da die relevanten Daten stark von der jeweiligen Domäne des Informationssystems abhängen und damit jede Domäne Besonderheiten mit sich bringt, ist der Einsatz eines Hilfsmittels zur strukturierten Erfassung der relevanten Daten ratsam. Ein mögliches Hilfsmittel für die strukturierte Erfassung dieser Daten ist ein sogenanntes Data Dictionary. Darunter versteht man ein Dokument, welches Metadaten, also Daten über Daten enthält. Es liefert Definitionen für alle relevanten Begriffe des Informationssystems und weist damit Ähnlichkeiten mit einem Wörterbuch auf. Im Gegensatz zu einem klassischen Wörterbuch, welches mitunter für ein und denselben Begriff mehrere Definitionen erlaubt, umfasst das Data Dictionary für jeden Begriff genau eine Definition. Mithilfe dieser einheitlichen Definition können Missverständnisse bei der Kommunikation zwischen verschiedenen Stakeholdern vorgebeugt werden.

Für die Konzeptionierung der Fitness-App PersonalFit hat sich nach wenigen Meetings gezeigt, dass die präzise Verwendung bestimmter Begrifflichkeiten für eine erfolgreiche interne sowie externe Kommunikation zwingend nötig ist. Das Data Dictionary, welches in Tabelle 1 abgebildet ist, wurde im Laufe des Projekts stetig erweitert, da auch der für die Konzeptionierung benötigte Wortschatz im Laufe des Projekts gewachsen ist. Das Data Dictionary wurde in Anlehnung an die ISO/IEC 11179-4 Norm erstellt. Zusätzlich zu der Definition, findet sich für den jeweiligen Begriff außerdem ein vorgeschlagener Datentyp (siehe dafür auch Abbildung 3) und eine mögliche, beispielhafte Ausprägung des Begriffs.

Im Folgenden werden die Einträge des in diesem Abschnitt abgebildeten Data Dictionary detailliert beschrieben.

Tabelle : Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Begriff | Datentyp | Beschreibung | Beispiel |
| Workout | Übung[1…n] | Ein Workout besteht aus mindestens 2 verpflichtenden und mindestens einer optionalen Übung. Nach Erfüllen aller nötigen Übungen des Workouts gilt dieses als abgeschlossen. Ein abgeschlossenes Workout trägt zum Fortschritt aller damit verbundenen Ziele bei. Ein Workout ist einem oder mehreren Zielen zugeordnet. | Cardio-Workout |
| Übung | Übung | Eine Übung besteht aus einer Beschreibung in Textform, einer Anleitung in Form eines Videos, einer grafischen Darstellung der beanspruchten Muskelpartien sowie Anzahl der vorgeschlagenen Sätze, Wiederholungen und Pausen. | Liegestütz |
| Ziel | Ziel | Es existieren fünf veschiedene Ziele: "Fokus", "Kraft", "Balance", "Ausgleich" und "Straffen". Der Benutzer wählt bei der Einrichtung der PersonalFit-App mindestens eines der Ziele aus. Ihm stehen im Anschluss Workouts zu allen ausgewählten Zielen zur Verfügung. | Fokus |
| Altersspanne | Altersspanne | Der Benutzer gibt nicht sein konkretes Alter oder seinen Geburtstag an, sondern wählt die seinem Alter entsprechende Altersspanne aus. Dabei gibt es vier verschiedene Ausprägungen: "bis 29", "30-44", "45-64" und "über 65". | 30-44 |
| (Trainings)intensität | Integer | Bei der Einrichtung gibt der Benutzer die gewünschte Trainginsintensität an. Die Intensität reicht von 1 (niedrigste Intensität) bis 5 (höchste Intensität) und wird bei der empfohlenen Anzahl der Wiederholungen und Sätze aller Workouts berücksichtigt. | 3 |
| Level | Integer | Der Benutzer hat für jedes bei der Einrichtung ausgewählte Ziel ein Level, welches seinen Fortschritt für das jeweilige Ziel angibt. Mit dem erfolgreichen abschließen von Workouts steigt das Level aller mit dem Workout verbundenen Ziele. Der Benutzer befindet sich zunächst auf Level 1; eine maximale Begrenzung gibt es nicht. Mit zunehmendem Level werden dem Benutzer zusätzliche verpflichtende sowie optionale Übungen angeboten. | 42 |

**Workout**

Die ersten beiden Begriffe „Workout“ und „Übung“ des Data Dictionary in Tabelle 1 stehen in engem Zusammenhang zueinander, denn ein Workout ist im Grunde genommen eine Menge von Übungen. Ein Workout besteht dabei aus mindestens zwei verpflichtenden Übungen. Jede verpflichtende Übung muss für das erfolgreiche Abschließen des Workouts durchgeführt werden. Darüber hinaus beinhaltet jedes Workout eine Reihe von optionalen Übungen (mindestens zwei), von denen nur ein gewisser Teil absolviert werden muss, damit das Workout als abgeschlossen gilt. Der Anteil der für den Abschluss des Workouts nötigen Übungen kann von Workout zu Workout variieren, um dem Benutzer eine dynamische Erfahrung zu bieten. Für die konkrete Umsetzung des hier präsentierten Konzepts ist für die Erstellung eines neuen Workouts sowohl die Anzahl der Übungen, als auch das Verhältnis von verpflichtenden und optionalen Übungen der Thematik des Workouts entsprechend zu wählen. Ein Workout zur Förderung der Konzentration, welches beispielsweise das Lösen eines Sudoku-Rätsels umfasst, kann dabei unter Umständen aufgrund der dafür erforderlichen Dauer eine niedrigere Anzahl von Übungen beinhalten als ein Cardio-Workout.

Neben der Thematik eines Workouts, welche durch die Bezeichnung (beispielsweise „Cardio-Workout“) wiedergespiegelt wird, verfügt jedes Workout über eines oder mehrere Ziele. Wird das Workout abgeschlossen, so trägt dieses zum Level-Fortschritt in jedem der mit dem Workout verbundenen Ziele bei. Dem Benutzer werden nur solche Workouts zur Auswahl gegeben, welche sich mit seinen bei der Einrichtung der PersonalFit-App gewählten Zielen decken. Wird ein mit dem Workout verbundenes Ziel vom Benutzer nicht verfolgt, so wird ihm jenes Workout auch nicht zur Verfügung gestellt.

**Übung**

Eine Übung ist eine Aufgabe, die der Benutzer zu erfüllen hat. Sie bildet gemeinsam mit anderen Übungen ein Workout und kann dabei sowohl körperliche als auch geistige Leistungen erfordern.

Damit der Benutzer die Übung korrekt umsetzen kann, wird diese sowohl durch eine Beschreibung in Textform, als auch mithilfe eines Videos der beispielhaften Durchführung der Übung erklärt.

Handelt es sich um eine körperliche Übung, werden die hierfür beanspruchten Muskelpartien anhand einer Skizze des menschlichen Körpers visualisiert. Zudem wird der geschätzte Kalorienverbrauch angegeben. Im Falle einer körperlosen Übung entfallen diese Darstellung und Angabe.

Je nach Gestalt der Übung, muss diese lediglich einmal oder mehrfach durchgeführt werden. Eine Übung sollte nicht mehr als zehn Minuten in Anspruch nehmen, sodass ein Workout die Dauer von 30 Minuten nicht übersteigt. Für sehr kurze Übungen, wie beispielsweise der Liegestütz, kann eine mehrfache Durchführung angegeben werden. Aufeinanderfolgende Durchführungen der Übung werden in „Wiederholungen“ angegeben, die mehrfache Durchführung einer Anzahl von Wiederholungen wird in „Sätzen“ angegeben. Falls mehrere Sätze einer Übung vorgesehen sind, so sollte die Dauer der zwischenzeitlichen Pause angegeben werden.

**Ziel**

Bei der Einrichtung der PersonalFit-App wählt der Benutzer aus den fünf Zielen "Fokus", "Kraft", "Balance", "Ausgleich" und "Straffen" all diejenigen aus, die er verfolgen möchte, mindestens jedoch eines. Dem Benutzer werden daraufhin lediglich Workouts zu einem der von ihm ausgewählten Zielen oder einer Kombination von diesen zur Verfügung gestellt.

Das Ziel „Fokus“ umfasst Workouts zur Verbesserung der Konzentrationsfähigkeit. Denkbar sind hierbei etwa Sudoku-Rätsel (vgl. QUELLE). Hat der Benutzer das Ziel „Kraft“ ausgewählt, so stehen ihm Workouts zur Verfügung, die den Muskelaufbau beabsichtigen (vgl. QUELLE). „Balance“-Workouts hingegen umfassen Übungen, welche auf eine Verbesserung der Beweglichkeit und Flexibilität des Benutzers abzielen, wie beispielsweise Pilates-Übungen (vgl. QUELLE). Hinter dem Ziel „Ausgleich“ verbirgt sich die Stressreduktion, welche beispielsweise durch meditative Übungen oder Yoga erreicht werden kann (vgl. QUELLE). Beabsichtigt der Benutzer das „Straffen“-Ziel zu verfolgen, erwarten ihn hierfür Übungen zur Körperfettreduktion (vgl. QUELLE).

**Altersspanne**

Die Altersspanne entspricht dem ungefähren Alter des Benutzers, welches er bei der Einrichtung der PersonalFit-App angibt. Sie wird dafür genutzt, dem Benutzer eine sinnvolle Anzahl von Wiederholungen, Sätzen und Pausen von körperlichen Übungen zu empfehlen. Es ist außerdem denkbar, einige Workouts, welche hohe Ansprüche an die körperliche Fitness des Benutzers stellen, lediglich den Benutzern der jüngeren Altersspannen freizugeben. Die Annahme, dass ältere Benutzer mit einer höheren Wahrscheinlichkeit gesundheitliche Einschränkungen aufweisen sowie die Auswahl der Altersspannen „bis 29“, „30-44“, „45-64“ und „über 65“ basieren auf der Studie Gesundheit in Deutschland aktuell 2009, 2010 und 2012 (QUELLE).

**Trainingsintensität**

Die Trainingsintensität wird ebenfalls bei der Einrichtung der App vom Benutzer eingegeben. Es handelt sich dabei um einen numerischen ganzzahligen Wert zwischen eins und fünf. Er wird für die Berechnung der empfohlenen Anzahl der Wiederholungen, Sätze und Pausen aller Übungen verwendet, also sowohl der körperlichen als auch der geistigen Übungen. Die Trainingsintensität erlaubt es dem Benutzer, die PersonalFit-App seinen persönlichen Vorstellungen anzupassen.

**Level**

Das Level gibt den Fortschritt des Benutzers für ein von ihm verfolgtes Ziel an. Der Benutzer hat für jedes seiner Ziele ein separates Level. Dieses steigt, wenn der Benutzer ein dem Ziel entsprechendes Workout erfolgreich abschließt. Die Anzahl der zum Levelanstieg benötigten abgeschlossenen Workouts nimmt mit steigendem Level zu. So ist es denkbar, dass der Benutzer zum Erreichen von Level 2 lediglich drei Workouts abschließend muss, zum Levelanstieg von Level 14 auf 15 jedoch bereits zehn Workouts nötig sind. Der Benutzer befindet sich anfangs auf Level 1 und kann beliebig viele Levelanstiege erreichen.

Nach jedem Levelanstieg wird der Benutzer darum gebeten, den Einstufungstest zu wiederholen. Durch die Wiederholung der anfänglich durchgeführten Übungen, welche dem Benutzer nach einer Reihe absolvierter Workouts leichter fallen sollten, soll dem Benutzer sein eigener Fortschritt für das jeweilige Ziel veranschaulicht werden.

Das Level-System hat neben der der Verdeutlichung des Fortschritts außerdem noch den Zweck, dass fortgeschrittene Übungen erst ab einem bestimmten Level freigeschaltet werden können. Dies hat den Vorteil, dass der Benutzer anfänglich nicht überfordert wird und außerdem durch neue Herausforderungen und Abwechslung langfristig motiviert wird.

## Modelle

Der Begriff „Modell“ hat in der natürlichen Sprache unterschiedliche Bedeutungen, darunter „Darstellung“, „Vorbild“ und „Abbild“ (vgl. QUELLE Duden). Eine für die Softwareentwicklung anerkannte Definition des Begriffs ist in QUELLE Stachowiak 1973 zu finden und besagt, dass ein Modell immer ein Abbild bzw. Repräsentation der Realität ist. Dabei werden nicht alle Eigenschaften der Realität abgebildet, sondern es wird durch das Prinzip der Abstraktion eine Reduktion des Detailgrades hervorgerufen. Eine Reduktion der abgebildeten Details ermöglicht es dem Modellschaffer wiederum, den Fokus auf die relevanten Aspekte zu richten.

Für die Konzeptionierung der PersonalFit-App wurden zwei Modelle angefertigt. Die Sitemap, welche in Abschnitt 3.3.1 detailliert beschrieben wird, erlaubt die übersichtliche Abbildung aller Ansichten der App sowie die Navigation zwischen ebendiesen. Umgesetzt wurde die Sitemap als UML Block Definition Diagram.

Bei dem zweiten Modell handelt es sich um ein UML-Klassendiagramm, welches alle relevanten Objekte, Daten, Datentypen und Kardinalitäten abbildet. Das Diagramm sowie die dazugehörige Beschreibung sind in Abschnitt 3.3.2 zu finden. Beide Modelle wurden mithilfe von Microsoft Visio umgesetzt.

Zu Beginn der Projektarbeit wurde zusätzlich die Erstellung weiterer Modelle in Erwägung gezogen. Es hat sich jedoch zu keinem Zeitpunkt im Projektverlauf die Verwendung eines weiteren Modells angeboten. Eine Datenbanknotation beispielsweise hat sich im Laufe der Konzeptionierung aufgrund der weitreichenden Überschneidungen mit dem Klassendiagramm als überflüssig herausgestellt. Das Erstellen von Modellen dient schließlich nicht dem Selbstzweck, sondern sollte vielmehr als Hilfsmittel für die vereinfachte Abbildung komplexer Sachverhalte genutzt werden.

### Sitemap

Als „Sitemap“ wird eine Übersicht über alle verfügbaren Seiten beziehungsweise Ansichten bezeichnet. Der Begriff stammt ursprünglich aus dem Bereich des Webdesigns. Dort werden Sitemaps häufig dafür eingesetzt, die eigene Website auf eine für Suchmaschinen optimierte Weise aufzulisten.

In Bezug auf die Konzeptionierung einer mobilen App, kann eine Sitemap jedoch noch weitere Funktionalität bieten. So ist es möglich, neben der Auflistung aller Ansichten, zusätzlich eine Verkettung der Ansichten untereinander darzustellen. Für jede einzelne Ansicht können also alle nachfolgenden Ansichten abgebildet werden. Die Sitemap stellt also außerdem die gesamte mögliche Navigation durch die App wieder.

Darüber hinaus kann für die Navigation von einer Ansicht zur Darauffolgenden angegeben werden, ob hierfür eine Aktion erforderlich ist und wenn ja welche. Beispielsweise kann das Betätigen des „Zurück-Buttons“ die vorherige Ansicht aufrufen.

Für das Projekt PersonalFit hat sich die Sitemap als übersichtliche Darstellung des gesamten App-Umfangs bewährt. Mit ihrer Hilfe konnte im Zuge der Bestimmung des Funktionsumfangs der App sichergestellt werden, dass alle relevanten Aspekte berücksichtigt werden.

Im weiteren Verlauf des Projekts wurde die Sitemap zudem als Grundlage für die Erstellung von Papierprototypen sowie Mock-Ups genutzt. Zwar ist die Sitemap zunächst unabhängig vom finalen Design der App, implizit wurden jedoch frühe Design-Entscheidungen bereits bei der Erstellung der Sitemap getroffen. Bezüglich der Einrichtung durch den Benutzer kam beispielsweise die Frage auf, wie viele Ansichten für die initiale Einrichtung der App nötig sind. Zur Beantwortung dieser Frage war es nötig abzuwägen, wie viel Platz die einzelnen Schritte der Einrichtung beanspruchen würden.

Im Folgenden wird die Sitemap aus Abbildung 2 und damit der strukturelle Aufbau der PersonalFit-App detailliert beschrieben. Das dort abgebildete Block Definition Diagram hat im Wesentlichen zwei Bestandteile: Blöcke und Kanten. Erstere symbolisieren jeweils eine Ansicht, wohingegen die Kanten mögliche Navigationsschritte abbilden. Kanten können außerdem mit der für die Navigation benötigten Aktion beschriftet sein. Zudem gibt es in der Sitemap zwei Kanten ohne Ursprung, welche den Systemstart oder Systemeinstieg repräsentieren.

Einer der möglichen Systemeinstiegspunkte tritt lediglich auf, wenn es sich dabei um den ersten Systemstart handelt. Der Benutzer muss die PersonalFit-App in diesem Fall die zunächst einrichten.

Dafür wird er eingangs mit einer Willkommens-Ansicht begrüßt. Auf der gleichen Seite wird er darum gebeten seinen Vornamen einzugeben, mit dem er fortan persönlich angesprochen wird.

Hat er diese Eingabe vorgenommen und bestätigt, wechselt die App zur nächsten Ansicht, auf der der Benutzer seine Fitnessziele auswählen kann (vgl. Tabelle 1: Data Dictionary).

Durch eine erneute Bestätigung öffnet sich die darauffolgende Seite, auf der die Trainingsintensität sowie die -häufigkeit durch den Benutzer einzutragen ist. Die Definition der Trainingsintensität ist ebenfalls im Data Dictionary zu finden. Die Trainingshäufigkeit wird dem Benutzer einerseits nach der Einrichtung auf dem Dashboard angezeigt und andererseits genutzt um den Benutzer zu benachrichtigen, falls dieser bereits eine längere Zeit kein Workout abgeschlossen hat.

Auf der nächsten Seite gibt der Benutzer an, ob er Benachrichtigungen der PersonalFit-App empfangen möchte. Denkbar sind neben den bereits erwähnten Benachrichtigungen bei längerer Trainingspause außerdem Benachrichtigungen zu unterbrochenen Workouts, die für deren erfolgreichen Abschluss fortgeführt werden müssen.

Die darauffolgende Ansicht listet eine Auswahl von unterstützten Trackinggeräten auf, zu denen der Benutzer angibt, ob er im Besitz eines oder mehrerer dieser Geräte ist. Trackinggeräte können zur automatischen Erfassung abgeschlossener Übungen genutzt werden, indem die verbauten Sensoren wie beispielsweise der Pulsmesser genutzt werden.

Auf der nächsten Seite wird der Benutzer nach einer Reihe von persönlichen Daten gefragt. Sämtliche dieser Angaben sind freiwillig und es besteht hier die Möglichkeit die Eingabe zu überspringen. Die Daten werden außerdem vertraulich behandelt und verbleiben ausschließlich auf dem Gerät, auf dem die App zum Einsatz kommt. Bei den persönlichen Daten handelt es sich um das Geschlecht, das Alter als Altersspanne (siehe Data Dictionary), die Größe, das Gewicht und den Umfang an verschiedenen Körperstellen (Brust, Hüfte, Oberarm, Taille und Oberschenkel). Das Gewicht und die Umfangsmaße können in regelmäßigen zeitlichen Abständen durch die PersonalFit-App aufgezeichnet werden, um dem Benutzer so seinen Fortschritt mithilfe grafischer Darstellungen aufzuzeigen. Persönliche Daten können außerdem vom System genutzt werden, um Workouts im Hinblick auf die Leistungsfähigkeit des Benutzers anzupassen.

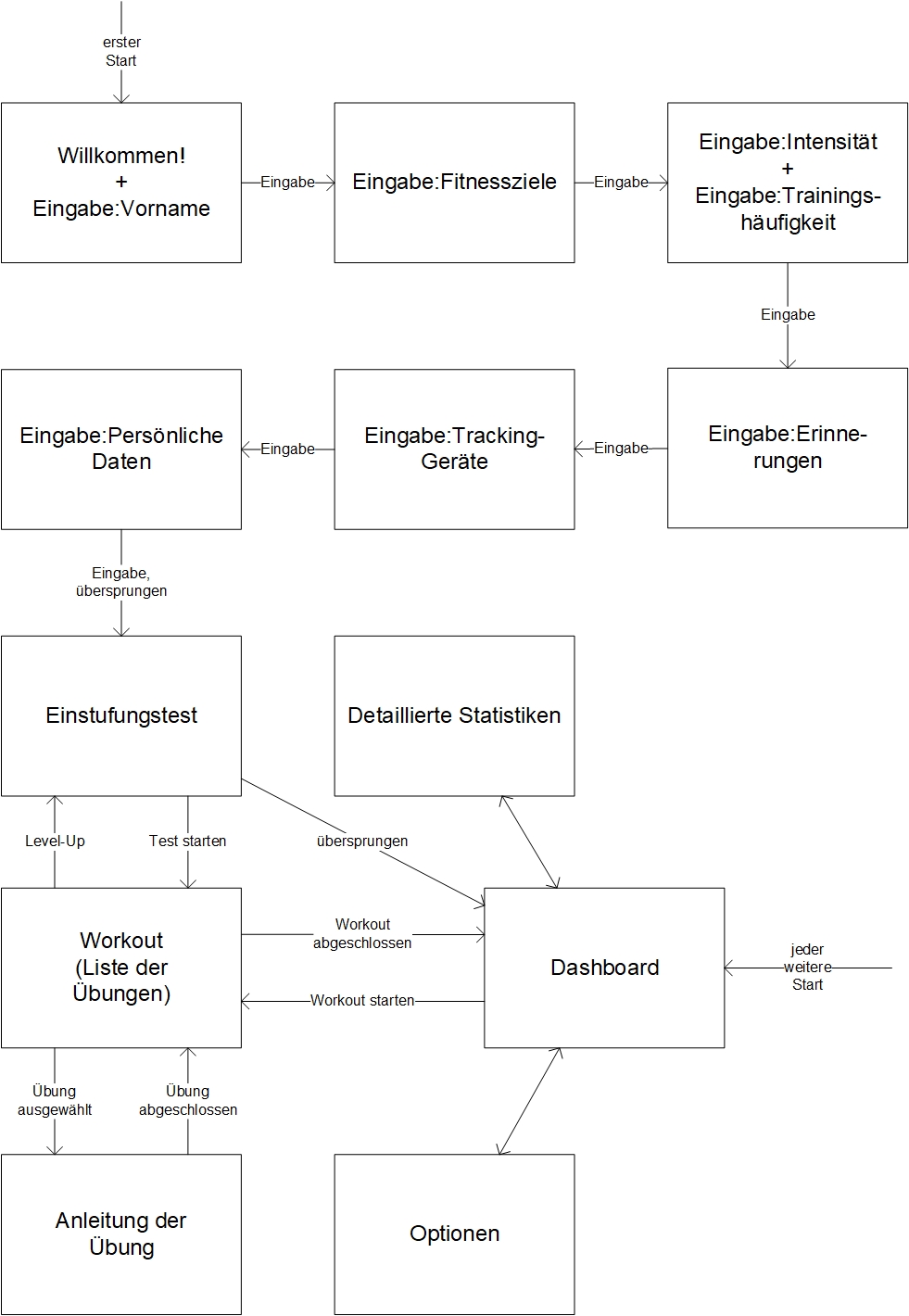


Abbildung : Sitemap

Mit der Eingabe der persönlichen Daten oder dem Überspringen dieser, ist die Einrichtung der App abgeschlossen. Der Benutzer wird anschließend darum gebeten, einen Einstufungstest durchzuführen. Zunächst wird ihm hierfür auf einer Seite erläutert, worum es sich bei dem Einstufungstest handelt. Er hat anschließend die Möglichkeit, den Test zu starten oder den Einstufungstest zu überspringen. Der Einstufungstest ist vergleichbar mit einem Workout, weist jedoch auch einige entscheidende Unterschiede auf: Er dient vordergründig nicht dem Training, sondern vielmehr der Selbsteinschätzung des Benutzers. Der Eignungstest wird nach jedem Levelaufstieg in gleicher Form wiederholt, was es dem Benutzer ermöglicht, den persönlichen Fortschritt wahrzunehmen. Der Eignungstest ist von kürzerer Dauer als ein übliches Workout und wird unmittelbar an ein Workout angeschlossen, falls dieses einen Levelaufstieg ausgelöst hat. Der Eignungstest trägt außerdem nicht zum Levelfortschritt bei.

Für den Fall, dass der Benutzer den Einstufungstest überspringt, gelangt er auf das Dashboard. Diese Ansicht ist zugleich der Einstiegspunkt der App bei jedem Systemstart mit Ausnahme der erstmaligen Ausführung. Das Dashboard ist der Dreh- und Angelpunkt der PersonalFit-App. Diese Seite zeigt dem Benutzer zu jedem seiner ausgewählten Ziele das Level, den jeweiligen Fortschritt bis zum nächsten Levelanstieg sowie die ausgewählte Trainingshäufigkeit an. Zu jedem der Ziele kann der Benutzer direkt vom Dashboard aus ein Workout starten und gelangt damit zur Ansicht „Workout“. Außerdem werden dem Benutzer für jedes seiner Ziele die Gesamtdauer aller hierfür absolvierten Workouts, sowie im Falle von körperlichen Zielen (Kraft, Balance und Straffen) die Summe der verbrauchten Kalorien angezeigt. Mit einem Klick auf diese Statistiken, gelangt der Benutzer zur Ansicht „Detaillierte Statistiken“.

Unter dieser Ansicht werden dem Benutzer sämtliche für ihn relevante Statistiken in grafischer Form präsentiert. Bei den Statistiken handelt es sich um:

* die Anzahl der durchgeführten Workouts
* die Summe der dadurch verbrauchten Kalorien (falls der Benutzer mindestens ein körperliches Ziel ausgewählt hat)
* die täglich zurückgelegten Schritte (falls der Benutzer über ein geeignetes Trackinggerät verfügt)
* das Gewicht des Benutzers
* die vom Benutzer eingegeben Körperumfangsmaße (Brust, Hüfte, Oberarm, Taille und Oberschenkel)

Der Benutzer hat auf dieser Seite außerdem die Möglichkeit, Diagramme hinzuzufügen oder zu entfernen und für vorhandene Statistiken neue Messpunkte einzugeben. Für die Darstellung der abgebildeten Daten sind eine geeignete Form sowie geeignete Maßstäbe zu wählen, welche mit zunehmender Datenmenge dynamisch angepasst werden.

Entschließt sich der Benutzer dafür, ein Workout zu starten, so gelangt er auf die Ansicht „Workout“. Dort befindet sich eine Liste der Übungen, welche in verpflichtende und optionale Übungen unterteilt ist. Der Benutzer wählt die entsprechende Übung aus, um diese zu starten. Die Reihenfolge ist ihm dabei freigestellt. Nach dem erfolgreichen Abschluss jeder Übung, gelangt der Benutzer zurück zu der Ansicht „Workout“ und kann anschließend die nächste Übung auswählen. Bereits abgeschlossene Übungen werden in dieser Übersicht entsprechend markiert. Zudem wird die für den Abschluss des Workouts nötigen optionalen Übungen sowie der bisherige Fortschritt angezeigt.

Die Ansicht „Anleitung der Übung“ erreicht der Benutzer, wenn er eine Übung aus der Liste der Übungen eines Workouts auswählt. Diese Seite beinhaltet alle für die Durchführung der Übung relevanten Informationen: eine Beschreibung in Textform, eine Anleitung in Form eines Videos, die beanspruchten Muskelpartien sowie die vorgeschlagenen Sätze und Wiederholungen (letztere nur für körperliche Übungen, vgl. auch Tabelle 1: Data Dictionary). Falls der Benutzer über ein geeignetes Trackinggerät verfügt und die Übung dies ermöglicht, kann das erfolgreiche Abschließen der Übung automatisch erkannt werden. Für den Fall, dass dies nicht möglich ist, bestätigt der Benutzer manuell, dass er die Übung abgeschlossen hat.

Bei der letzten Ansicht handelt es sich um die „Optionen“. Diese Seite ist ebenfalls vom Dashboard aus zu erreichen und dient dazu, alle bei der initialen Einrichtung der App getroffenen Eingaben zu korrigieren beziehungsweise zu ergänzen. So ist es beispielsweise denkbar, dass der Benutzer im Besitz eines neuen Trackinggerätes ist und dieses in der PersonalFit-App nutzen möchte.

### UML-Klassendiagramm

Bei der Unified Modeling Language, kurz „UML“ genannt, handelt es sich um eine Modellierungssprache, welche seit den 1990er Jahren existiert. Heute ist die von der Object Management Group entwickelte Sprache die gängigste Modellierungssprache im Bereich der Softwareentwicklung (QUELLE).

Mit dem Klassendiagramm bietet die UML eine Möglichkeit, die statische Struktur eines Softwaresystems oder Softwareartefaktes abzubilden. Die wichtigsten Bestandteile eines Klassendiagramms sind die eigentlichen Klassen, deren Eigenschaften (Attribute), deren Verhalten (Operationen) und die Beziehungen der Klassen untereinander.

Als eine Klasse wird in diesem Zusammenhang eine Menge von gleichartigen Objekten definiert. Sie weisen sowohl die gleiche Struktur, als auch das gleiche mögliche Verhalten auf. Dabei wird die Struktur der Objekte durch die Attribute beschrieben, wohingegen das Verhalten mithilfe der Operationen ausgedrückt wird.

Zusätzlich bieten Klassendiagramme eine Reihe von Beziehungen, die mehrere Klassen miteinander in Verbindung setzen. Dadurch können unter anderem die Prinzipien der Vererbung, Komposition und Aggregation abgebildet werden.

Im Rahmen der Projektarbeit wurde das Klassendiagramm aus Abbildung 3 erstellt. Hierbei wurde vom Verhalten der Klassen abstrahiert und der Fokus auf die Eigenschaften der Objekte gelegt. Die Begründung hierfür hängt mit der Beschaffenheit der PersonalFit-App zusammen. Diese weist bis auf wenige Ausnahmen (beispielsweise den Levelfortschritt) einen relativ statischen Charakter auf und dient lediglich der Anzeige dieser statischen Elemente.

Dennoch ist der Nutzen des Klassendiagramms für die Konzeptionierung der PersonalFit-App gegeben, denn für die übersichtliche Darstellung aller statischen Elemente ist das Klassendiagramm gut geeignet. Zunächst gibt es eine Reihe von selbstdefinierten Datentypen. Diese sind in Abbildung 3 als <<Enumeration>> gekennzeichnet. Alle möglichen Ausprägungen dieser Datentypen sind in der jeweiligen Klasse aufgelistet. Das Alter beispielsweise kann die vier Werte „bis 29“, „30-44“, „45-64“ und „über 65“ annehmen. Bei der Enumeration „Muskel“ wurde aus Gründen der Übersichtlichkeit auf die Auflistung aller Muskeln des menschlichen Körpers verzichtet.

Eine der wichtigsten Klassen ist mit dem Benutzer zugleich die umfangreichste Klasse. Sie enthält alle bei der initialen Einrichtung der App eingegeben Daten, wie die bis zu fünf ausgewählten Ziele, den Vornamen, die optionalen persönlichen Daten (vgl. Abschnitt 3.3.1), die Auswahl des Benutzers bezüglich des Empfangs von Benachrichtigungen sowie die vorhandenen Trackinggeräte. Bei der Wahl der Datentypen ist darauf zu achten, dass die Angaben des Vornamens, Geschlechts, Alters und der Größe lediglich einen Wert zulassen. Werden diese nachträglich über die Einstellungen geändert, so wird der frühere Wert überschrieben. Die Eigenschaften bezüglich des Körperumfangs und des Gewichts können jedoch eine Vielzahl von Messwerten aufnehmen, sodass mit deren Hilfe die Veränderungen und Fortschritte des Benutzers im Laufe der Zeit festgehalten werden können. Die Ansicht der detaillierten Statistiken greift auf diese Daten zu, um sie in grafisch aufbereiteter Form darstellen zu können.

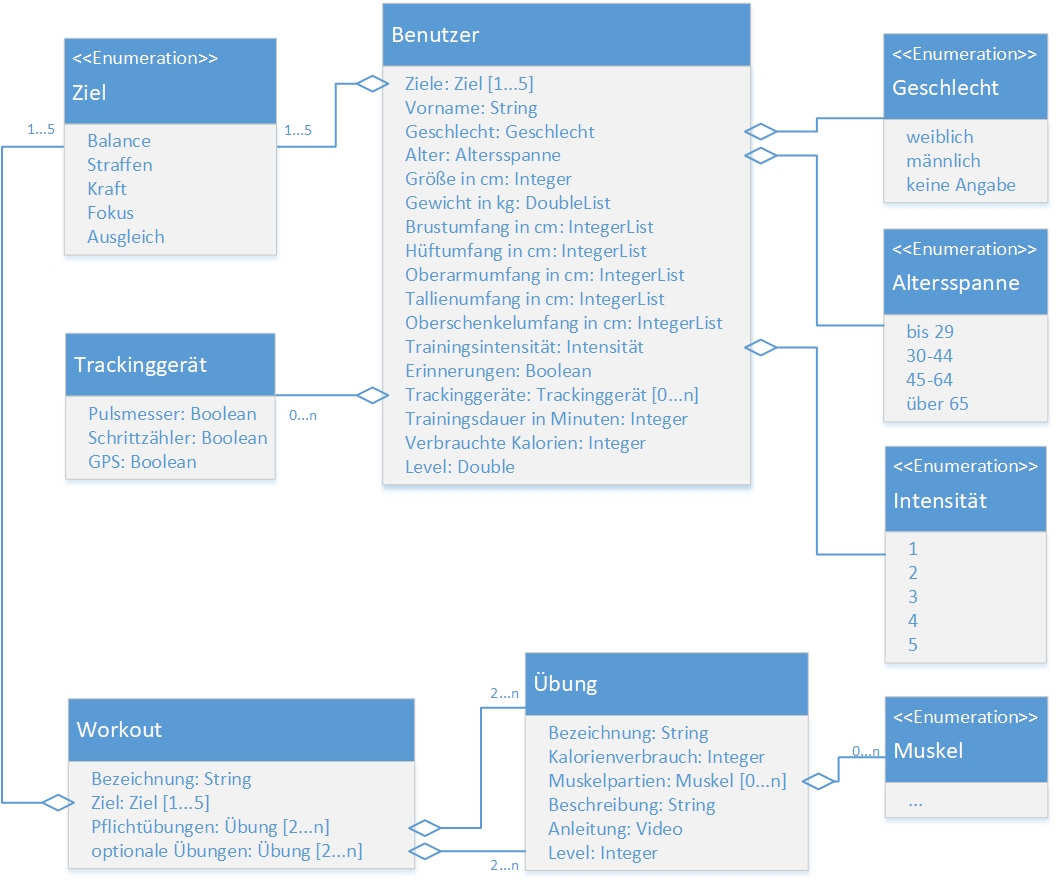


Abbildung : UML-Klassendiagramm

Neben den bei der initialen Einrichtung eingegebenen Daten, verfügt der Benutzer über drei Attribute, welche sich nach einem erfolgreich abgeschlossenen Workout automatisch aktualisieren. Die Summe der Trainingsdauer wird demnach um die Zeit erhöht, die der Benutzer für das Workout benötigt hat. Die verbrauchten Kalorien können aus dem Attribut der jeweiligen Übungen entnommen werden.

Bezüglich der Aktualisierung des Levelfortschrittes nach einem Workout gilt zu beachten, dass dieser mit zunehmendem Level stets niedriger ausfallen soll, sodass die ersten Levelanstiege schneller zu erreichen sind als die weiter fortgeschrittenen. Das Level kann beispielsweise als Gleitkommazahl umgesetzt werden, wobei die Stellen vor dem Komma das aktuelle Level des Benutzers angeben und die Stellen nach dem Komma den derzeitigen Fortschritt zum nächsten Levelanstieg in Prozent.

Die Klasse „Trackinggerät“ enthält Informationen zu jenen im Gerät verbauten Sensoren, welche zum automatisierten Erfassen der Übungen genutzt werden könnten. Es handelt sich dabei um gängige Sensorentypen, welche in vielen modernen „smart Wearables“ verbaut sind. Einige davon, wie der GPS-Sensor oder der (permanente) Pulsmesser, sind jedoch teils erst bei Geräten der höheren Preisklassen zu finden. Ohne den gesamten Übungskatalog bestimmt zu haben, ist es zudem schwer einzuschätzen, ob die drei hier genannten Sensoren für die automatische Erkennung der Übungen geeignet sind.

Die Klasse „Workout“ besteht, wie bereits im Data Dictionary in Tabelle 1 definiert, aus je mindestens zwei verpflichtenden sowie optionalen Übungen. Zudem ist das Workout bis zu fünf Zielen zugeordnet, mindestens jedoch einem Ziel. Es verfügt zusätzlich über eine Bezeichnung, welche den Fokus des Workouts ausdrücken sollte.

Die wesentlichen Bestandteile der Klasse „Übung“ sind ebenfalls bereits im Data Dictionary beschrieben worden. Jede Übung verfügt über eine Bezeichnung, einen geschätzten Kalorienverbrauch, eine Beschreibung in Textform, eine Anleitung in Form eines Videos, eine Liste der beanspruchten Muskelpartien. Außerdem kann für jede Übung angegeben werden, ab welchem Level diese für den jeweiligen Benutzer freigegeben werden soll. So ist es beispielsweise möglich, ein anspruchsvolle Übungen lediglich fortgeschrittenen Benutzern anzubieten. Das Level kann in diesem Fall einen ganzzahligen Datentyp haben, da neue Übungen erst nach einem Levelanstieg, nicht aber zwischen zwei Leveln freigeschaltet werden.

Die Assoziationen im hier abgebildeten Klassendiagramm treten in der Form von Aggregationen auf. Diese Assoziation bildet den Sachverhalt ab, dass eine Klasse ein Teil einer anderen Klasse ist. So ist beispielsweise die Übung Teil eines Workouts. Durch die Angabe der Kardinalitäten kann zusätzlich angegeben werden, wie viele Instanzen der Klasse Teil dieser Assoziation sind. Im Beispiel der eben genannten Aggregation handelt es sich um je mindestens zwei verpflichtende sowie optionale Übungen, die Teil eines Workouts sind.

## Prototypen

Prototyp oder auch Prototyping genannt, wird als eine Software-Entwicklungsstrategie angesehen. Die Strategie kann die Probleme, die während einer Entwicklung auftreten können, lösen. (G. Pomberger, W. Pree, A. Stritzinger, S. 1). Prototypen sind frühe Versionen von einem zu entwickelnden Softwaresystem. Diese erhalten einige oder alle Eigenschaften und vordefinierten Funktionen der letztendlichen Version.

Typische Eigenschaften der Prototypen sind zum einen, dass sie schnell und kostengünstig entwickelt werden und dem Benutzer bzw. auch dem Arbeitgeber einen Einblick von der geplanten Software anhand eines Prototyps mit wesentlichen Bestandteilen des Systems vor der Implementierung gibt. Zudem sind Prototypen flexibel, sodass eine Veränderung oder eine Erweiterung der Anforderungen einfach umzusetzen ist. Prototypen müssen nicht ein vollständiges System modellieren. (G. Pomberger, W. Pree, A. Stritzinger, S. 2)

Es gibt drei Kategorien in der Systematik von Prototypen. Die erste Kategorie ist das „Exploratives Prototyping“. Hier dienen die Prototypen als Kommunikationshilfe zwischen dem Auftraggeber bzw. auch dem Benutzer. Deshalb ist die Zusammenarbeit wichtig, um herauszufinden, ob die Anforderungen berücksichtigt oder vergessen worden sind. So haben die Auftraggeber bzw. der Benutzer die Möglichkeit aktiv Verbesserungs- und Änderungswünsche zu äußern bis alle Anforderungen und Funktionen erfüllt worden sind. (Aichele und Schönberger 2014, S.36). Die zweite Kategorie ist das „Experimentelles Prototyping“. Hierbei handelt es sich um die Erkenntnis von unbekannten Anforderungen an das System, die durch mehrmaligen Versuchen entstehen. Diese Methode ist bereits bei alten praktischen Softwareentwicklungsmethoden bekannt. (Spitta 1989, S. 5)

Es gibt eine Vielzahl an unterschiedlichen Prototyping-Werkezeuge. Vorausgesetzt ist die Möglichkeit eine schnelle und kostengünstige Erstellung des Prototyps zu erstellen. Der Wahl des Werkzeuges hat eine entscheidende Bedeutung auf den Einfluss der Qualität als auch auf den Aufwand der Erstellung und Bearbeitung des Prototyens. (G. Pomberger, W. Pree, A. Stritzinger, S.4)

Einige Autoren aber bevorzugen „echte Prototypen“, d.h. mit Funktionen als auch mit wichtigen Merkmalen ausgestattet. Da dieses zur Grundlage der Software-Entwicklungsprozess dient. Andere Autoren präferieren die „Bottom Up-Vorgehensweise“, d.h. es werden auf die einfachsten und wichtigsten Funktionen abstrahiert und dann vom Benutzer immer wieder getestet und verbessert. Es werden solange weitere Funktionen implementiert bis das System vollständig ist. (G. Pomberger, W. Pree, A. Stritzinger, S. 2)

Prototypen unterstützt zum einen die Kommunikation zwischen dem Kunden und dem Entwickler und schafft zusätzlich eine Grundlage für die Systemspezifikation. Durch die Einfachheit werden wesentliche Eigenschaften bzw. Anforderungen des geplanten Produktes veranschaulicht. Zusätzlich werden die Funktionalitäten deutlich ausdrucksvoller dargestellt als in einer textuellen Beschreibung oder rein statischen Modells. Die Vollständigkeit der bisherigen Spezifikation kann ebenfalls durch Prototypen verbessert werden und die Projektphasen bei einer Softwareentwicklung mit fehlenden Informationen ergänzen. (G. Pomberger, W. Pree, A. Stritzinger, S.3) Es wurde bewiesen, dass durch Prototypen die Wirtschaftlichkeit um rund 45% verbessert, die Qualität der Software erhöht und den Aufwand um 40% verringert. (Barry W. Boehm, Terence E. Gray, and Thomas Seewaldt, S. 293)

Doch das Prototyping bringt auch Nachteile mit sich. Da ein Entwicklungsprozess in der Regel linear und häufig ohne Wiederholungen der einzelnen Phasen verläuft, ist ein bestimmter Ablauf in der Entwicklungsphase festgelegt. Häufig findet das Prototyping zum Ende einer Entwicklungsphase erst statt, sodass auf Grund Kundenwünsche Änderung der vorherigen Phasen sich als schwierig und aufwendig gestalten lässt. Außerdem spielen spätere Phasen eine wichtige Rolle auf die Vorgehensweise. Ebenfalls kann die Überprüfung von Produkten und Komponenten erst sehr spät durchgeführt werden.

Im Folgenden werden die Papierprototypen und die Mock-Ups nahegebracht.

### Papierprototypen

...

### Wireframes

Wireframes (englisch Drahtgerüst) bauen auf den bisherigen erstellten Papierprototypen auf und umfasst das Design der zu entwickelnden App. Frühzeitig erstellte Wireframes helfen dabei die Softwareidee der App mehr zu veranschaulichen. Unabhängig von dem Werkzeug sollte die Erstellung einfach und kurz gehalten werden. Das hat den Vorteil, dass während der Entwicklung Änderungen und Erweiterung des Systems schnell umgesetzt werden können. Auf Grund der schnellen und einfachen Erstellung der Wireframes ist der Zeit - und auch der Kostenaufwand sehr gering. Wireframes haben eine kurze „Lebensdauer“. Trotz der geringen Lebensdauer ist diese Methode besonders einflussreich und wichtig, da bei der Durchführung der Methode die meisten innovativen Ideen entstehen. Durch den niedrigen Aufwand und der niedrigen Kosten der Entwürfe können somit riskante Ideen ausprobiert werden und wieder ohne hohen Kosten weggeworfen werden (QUEELEN EFFECTIV PROTOTZPING)

Im Folgenden werden die erstellten Wireframes für die App „PersonalFit“ näher erläutert. Die Wireframes wurden nach eigenen Vorstellungen und Präferenzen ohne Rücksicht auf die Guidelines von iOS entworfen.

**Aufmachung und Start**

Sobald die App „PersonalFit“ angeklickt worden ist, erscheint die Aufmachung mit dem Logo. Kurz darauf öffnet sich das Startfenster. Der Benutzer wird dann vom System als auch von seinem virtuellen und persönlichen „Personaltrainer“ begrüßt und wird aufgefordert seinen Vornamen in das Textfeld einzutragen. Nachdem der Benutzer seinen Vornamen angegeben hat, kann der Benutzer mit Hilfe des Buttons „Los geht’s“ auf die „Fitnesskategorien“ gelangen.

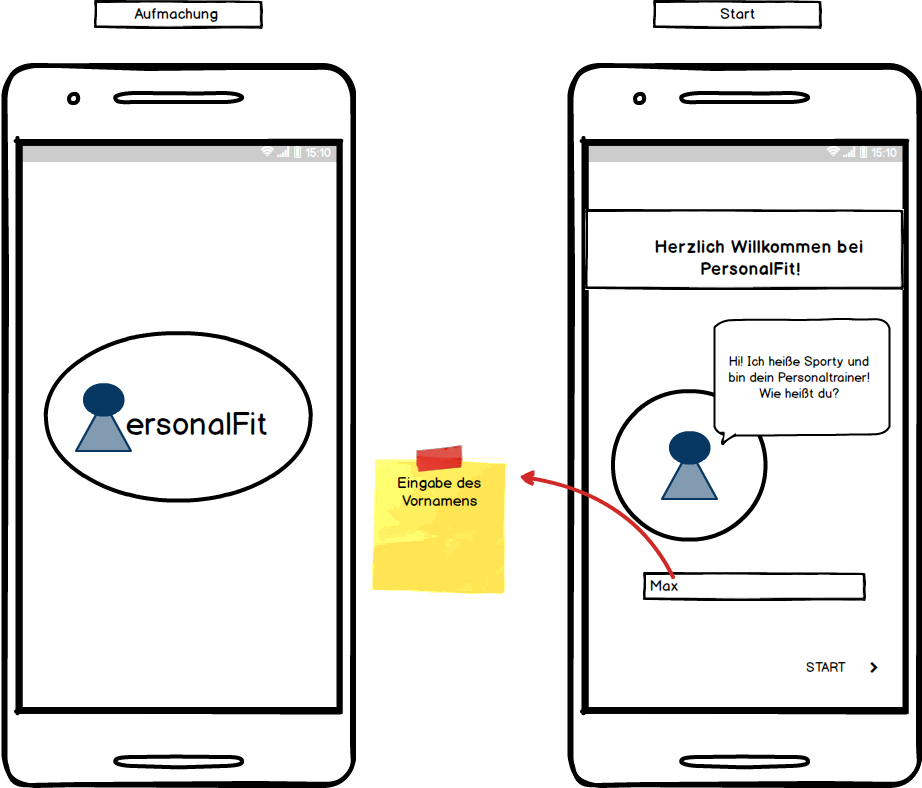
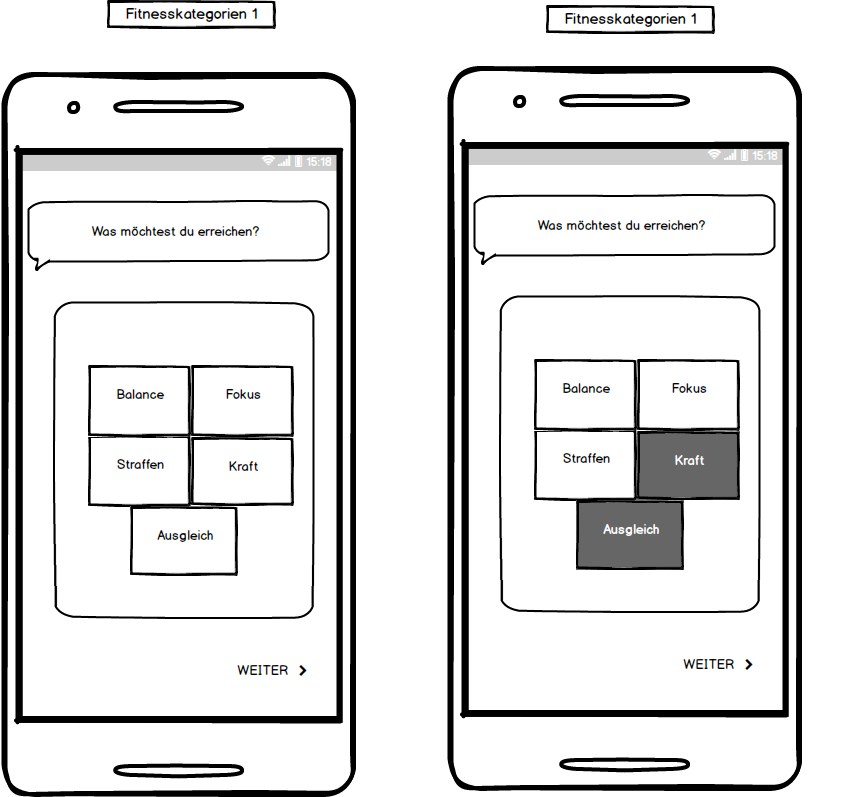


Abbildung : Wireframes Aufmachung und Start

**Fitnesskategorien**

Der „Personaltrainer“ fragt den Benutzer, was er genau mit Hilfe der App erreichen möchte. Der Benutzer kann sich zwischen den Fitnesskategorien („Balance“, „Straffen“, „Fokus“, „Kraft“ und „Ausgleich“) mindestens eine oder mehrere Fitnessziele aussuchen, in dem er auf die jeweilige Kategorie(n) klickt. Die ausgewählte Kategorie bzw. Kategorien werden durch Farben deutlich gemacht. Durch das Klicken des Buttons „Weiter“ wird der Benutzer automatisch auf die „Einstellungen“ weitergeleitet.



**Einstellungen**

Abbildung : Wireframes Fitnesskategorien

Nach der Auswahl der Fitnesskategorien werden eine Reihe von Einstellungen folgen. Dort werden den Benutzer nach seinen persönlichen Daten (zum Beispiel „Gewicht“, „Größe“) als auch Fragen zur Systemeinstellung (zum Beispiel „Möchtest du an dein Training erinnert werden?“) gefragt. Die Antworten der Fragen werden durch Schieberregler, Dropdowns und Buttons unterstützt. Einige Daten können optional eingetragen werden. Diese wurden mit einem kleinen Sternchen (\*) vermerkt. An einigen Stellen kann der Benutzer auf ein Fragezeichen klicken, sodass er Hinweise und Tipps von der App erhält. Durch das vertikale Scrollen wird ein Screen erweitert. Die Angabe in den Einstellungen kann jederzeit vom Benutzer geändert werden. Nachdem der Benutzer die Daten eingetragen hat, kann er auf den Button „Weiter“ klicken. Darauf folgen weitere „Einstellungen“. Sobald alle Einstellungen ausgefüllt worden sind gelangt er auf „Einstufungstest“.

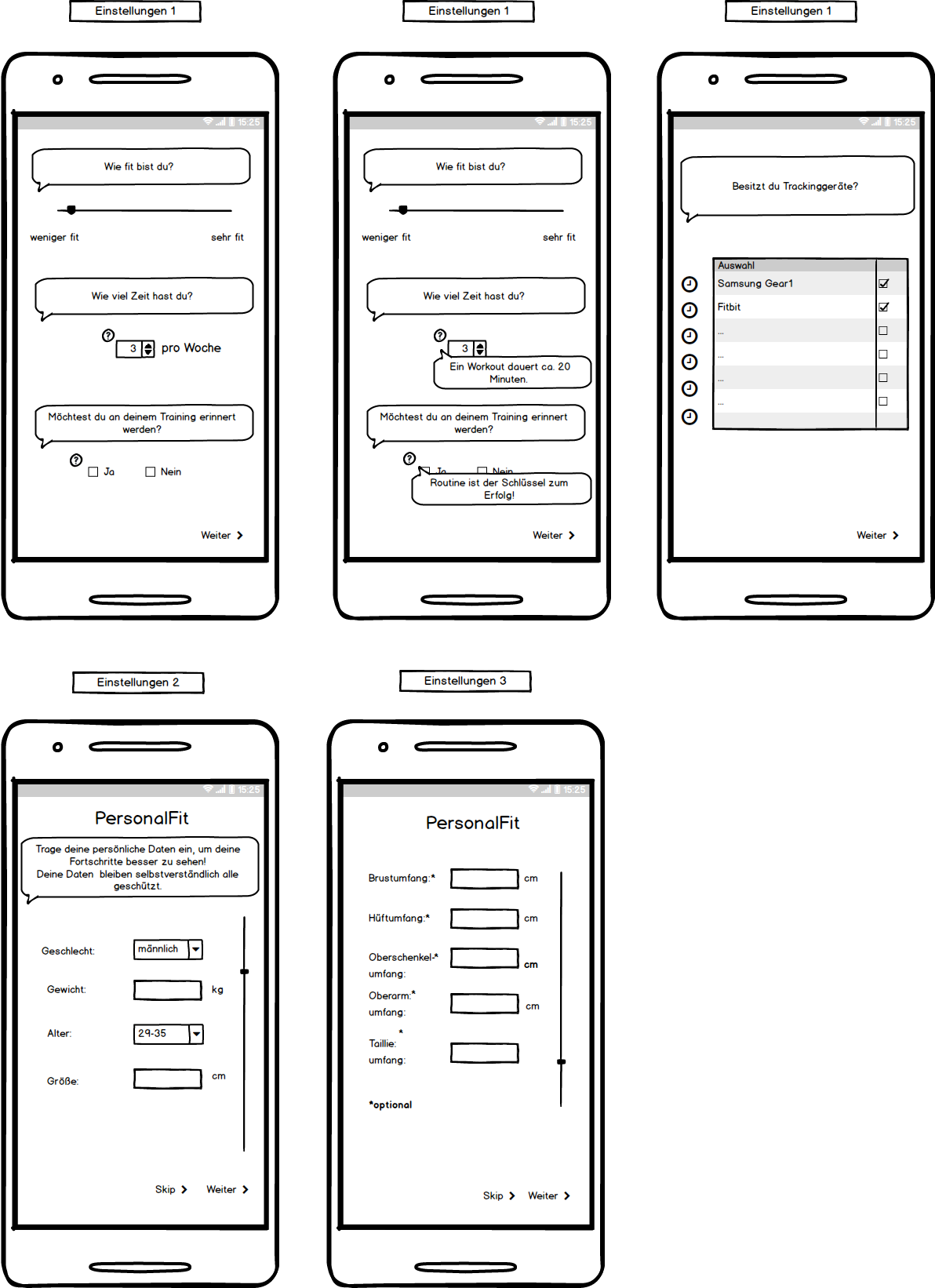


Abbildung : Wireframes Einstellungen

**Einstufungstest**

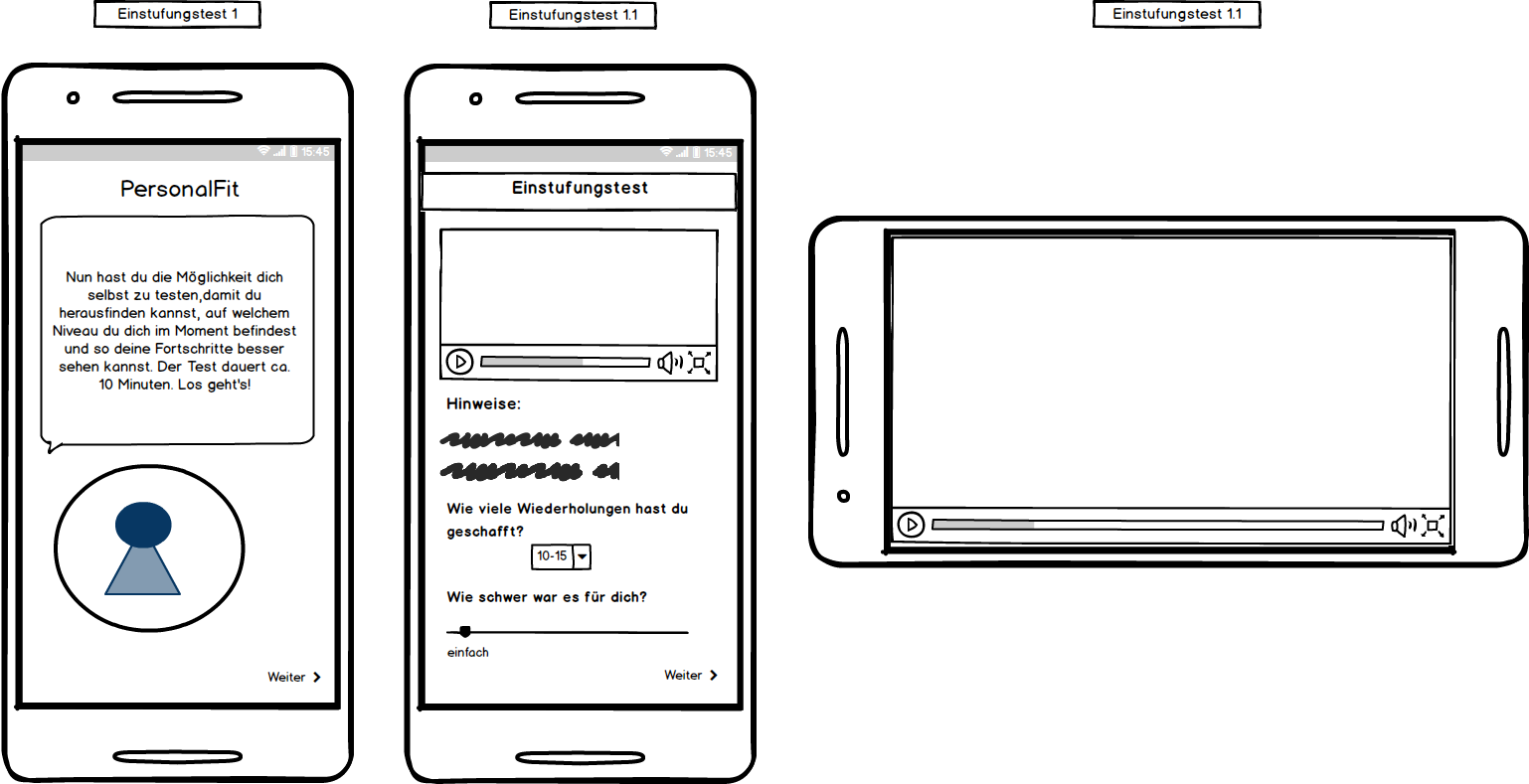
Der Benutzer hat die Möglichkeit an einen Einstufungstest teilzunehmen. Der virtuelle Personaltrainer klärt den Benutzer darüber auf. Der Benutzer kann sich dann entscheiden, ob er den Einstufungstest „Starten“ oder ihn später oder gar nicht machen möchte „Skippen“. Wenn er den Test nicht durchführen möchte, gelangt der Benutzer auf das „Dashboard“. Möchte aber der Benutzer an den Test teilnehmen, gelangt er auf die nächste Seite des Einstufungstests. Dort befindet sich Video wie der Einstufungstest abläuft. Das Video kann ebenfalls auf „Vollmodus“ geschaut werden. Unterhalb des Videos werden Hinweise zur Ausführung der Übung in textueller Form genannt. Unter den Hinweisen werden Fragen zur persönlichen Einschätzung der jeweiligen ausgeführten Übung gestellt. Der Benutzer kann diese beantworten mit Hilfe eines Dropdows und eines Schieberegels. Durch den Button „Weiter“ erfolgt immer wieder eine andere Übung des Testes bis der Test vorbei ist. Daraufhin wird der Benutzer zum Dashboard weitergeleitet.

Abbildung : Wireframes Einstufungstest

**Dashboard**

Auf dem Dashboard werden alle wichtigen Daten veranschaulicht wie zum Beispiel „Zielkategorie“, „Fortschritt“, „Level“ und die „Häufigkeit“. Einige angezeigten Daten kann der Benutzer sich aus der Statistikauswahl zusammenstellen, zum Beispiel „Gesamte Schritte“, „Aktuelles Gewicht“. Durch den Button „mehr Details“ gelangt der Benutzer auf die „Statistiken“. Der Button „ZUM WORKOUT“ führt das System den Benutzer auf sein persönliches Training „Mein PersonalTraining“.

WIREFRAMES SKIZZE

**Statistiken**

****Es werden alle Statistiken angezeigt, die der Benutzer sich in der „Auswahl“ der Statistiken ausgesucht hat. In der Auswahl befinden sich alle Statistiken, die das System zur Verfügung stellt. Alle ausgewählten Statistiken werden untereinander aufgelistet, sodass durch vertikales „scrollen“ erweitert wird. Mit Hilfe des Buttons „Zurück“ gelangt der Benutzer auf sein Dashboard.

Abbildung : Wireframes Statistiken

**Mein PersonalWorkout**

Alle Übungen des Workouts werden auf „Mein PersonalWorkout“ mit Namen angezeigt. Die erledigten Übungen werden mit einem Häkchen gekennzeichnet. Die „optionalen Übungen“ kann der Benutzer manuell aus einer vorgegeben Liste mit allen Übungen sich aussuchen, in dem er auf das „+“-Zeichen klickt. Beim Anklicken des Fragezeichens bekommt der Benutzer einen Hinweis.

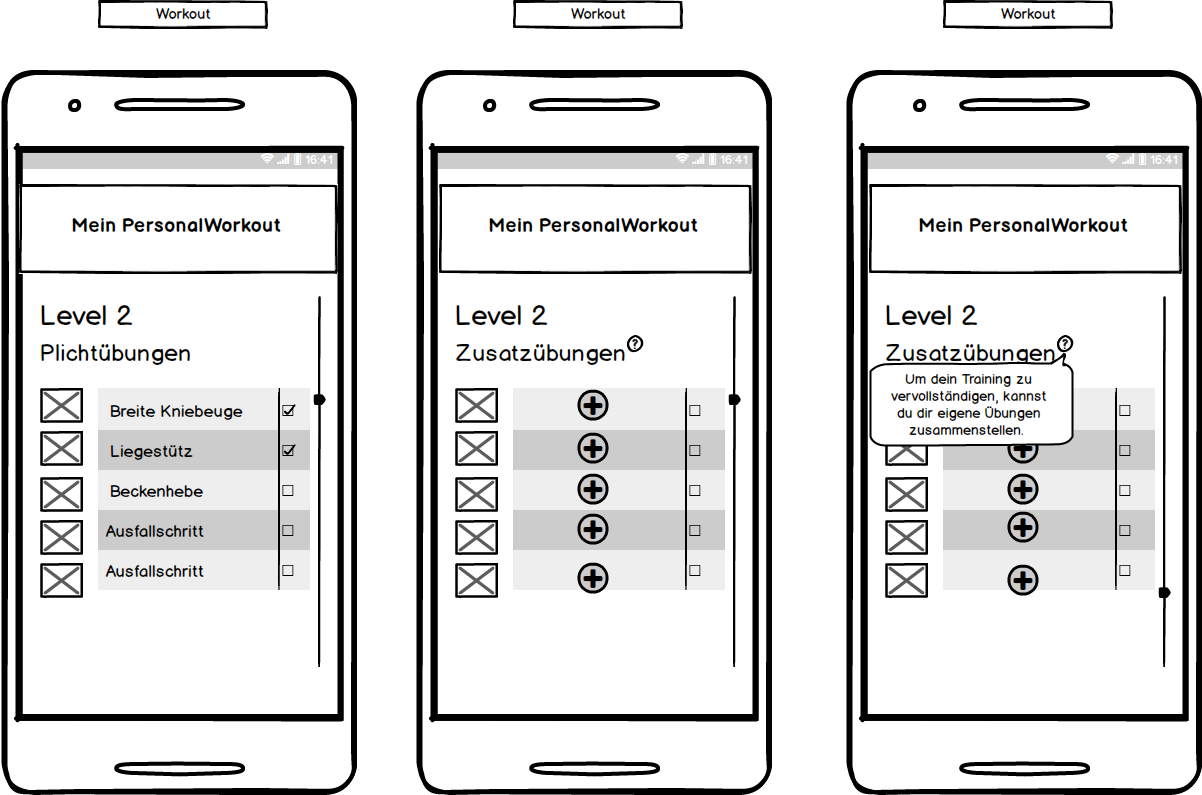


Abbildung : Wireframes Mein PersonalWorkout

**Übung**

Eine Übung wird durch ein Video und drei weitere Kategorien unterstützt, die als TabList dargestellt sind: „Planvorgabe“, „Ausführungen“ und „Muskulatur“. Die Kategorien „Planvorgabe“ und „Ausführungen“ werden in textueller Form beschrieben. Die Kategorie „Muskulatur“ wird mit Hilfe eines Bildes, worauf die beanspruchten Partien bei der Ausführung der Übungen gekennzeichnet worden sind, dargestellt. Zusätzlich werden die beanspruchteten Partien nochmal in textueller Form erwähnt.

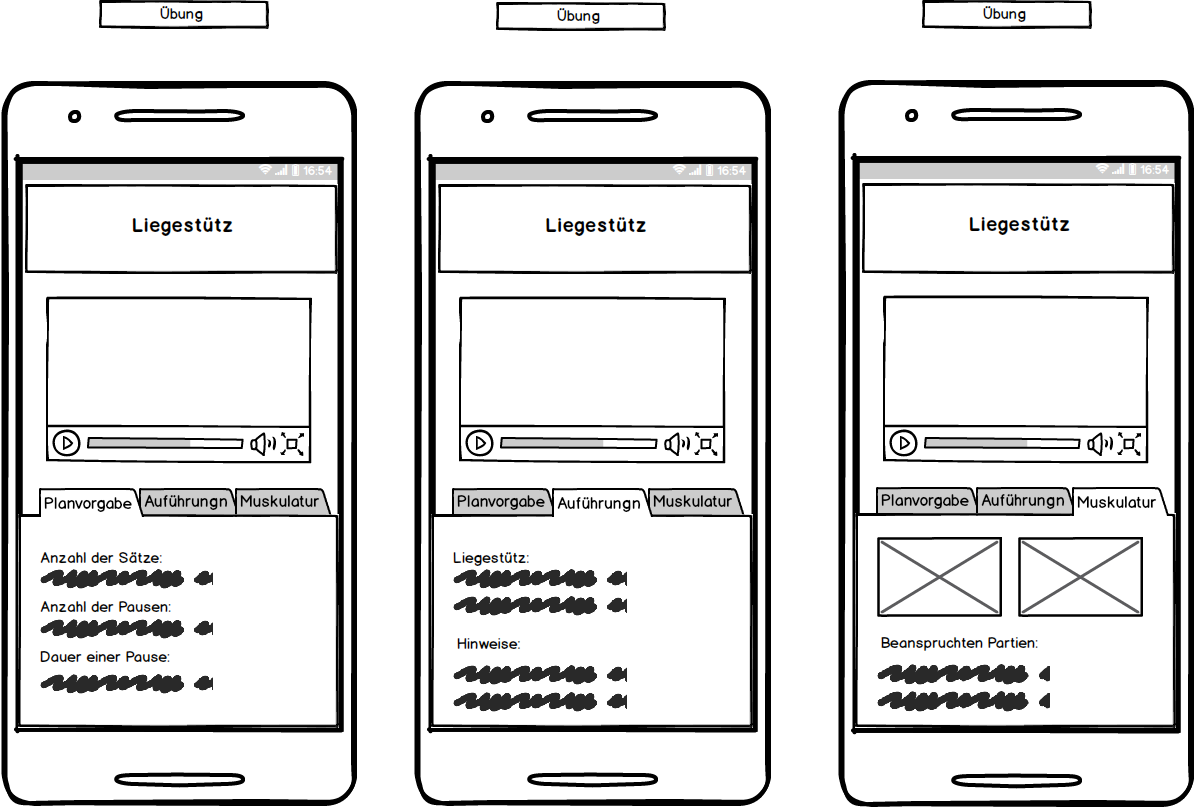


Abbildung : Wireframes Übung

Abbildung Aufmachung und Start

### Mock-Ups

Mockups (englisch für Lehrmodell) ist ein Werkzeug des Prototypings und unterstützt die Erstellung der Oberfläche der späteren App mit Hilfe von ersten Designentwürfen. Die Entwürfe zeigen dem Benutzer exemplarisch die Funktionen. Diese können mit Hilfe von Bildbearbeitungsprogramme oder speziellen Mockup-Tools erstellt werden. Der Aufwand der Erstellung der Entwürfe sollten gering wie möglich gehalten werden, um das Design dem Benutzer bzw. dem Auftragsgeber präsentieren und über die Anforderungen bzw. Applikation diskutieren zu können. Die Versionen sollten daher schnell und einfach an den jeweiligen veränderten Anforderungen der Stakeholder angepasst werden können. Bei der Ausarbeitung der Darstellung ist es wichtig zwischen den gewünschten Kern- und der ergänzenden Funktionen zu differenzieren. Grundsätzlich sollten die wesentliche n Funktionen dargestellt werden, um einen besseren Überblick über die App zu erhalten. Die ergänzenden Funktionen müssen nicht zwangsläufig veranschaulicht werden. Bei dem Design einer App müssen besondere Anforderungen berücksichtigt werden, zum Beispiel die begrenzte Bildschirmdiagonale und die sehr spezifischen Interaktionsmöglichkeiten. Für eine gute Bedienbarkeit wird eine entsprechendes App-Design vorausgesetzt. Daher ist der Einsatz der Daumenregel besonders sinnvoll. Die Daumenregel gilt für eine kleinere Displaydiagonale und die sich daraus ergebenden weitreichenden Darstellungsmöglichkeiten. Für größere Displaydiagonalen zum Beispiel Tablet-PCs gestaltet sich die Daumenregel schwierig. Vor der Erstellung der Entwürfe sollte überlegt werden für welches mobile Gerät die App zur Verfügung gestellt werden sollt und welche Einschränkungen und Möglichkeiten durch entstehen können. Die Daumenregel besagt, dass der Benutzer alle Funktionen mit seinem Daumen erreichen und bedienen können sollte, während er das Smartphone in einer Hand hält. Das Design und die Bedienbarkeit sind besonders wichtig, da das ein „Aushängeschild“ für das Unternehmen ist (eBusiness-Lotse 2015, S. 11, 22, 24).

Jedoch dürfen die bestehenden Guidelines nicht verletzt werden. Im Folgenden werden die für unsere App benötigten vorgestellt.

**User-Interaction**

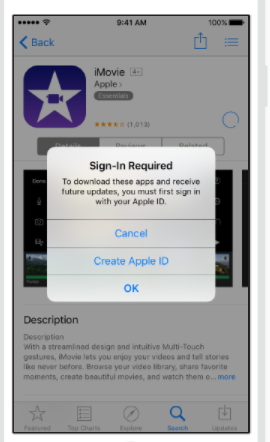
* **Authentification:** Eine Authentifizierung bittet den Benutzer sich zu authentifizieren, zum Beispiel für den Zugriff auf zusätzliche Funktionen, den Kauf von Inhalten oder die Synchronisierung von Daten. Wenn eine App eine Authentifizierung erfordert, sollte der Anmeldevorgang schnell, einfach und unauffällig dargestellt werden und damit nicht den Genuss der App für den Benutzer beeinträchtigen. Daher sollten Verzögerungen vermieden werden. Oftmals verlassen die Benutzer die App, sobald sie gezwungen sind sich anzumelden bevor sie etwas von der App sehen konnten. Den Benutzer sollten die Chance gegeben werden sich in die App zu verlieben bevor er sich verpflichten musst (Apple)

Abbildung : Authenfication

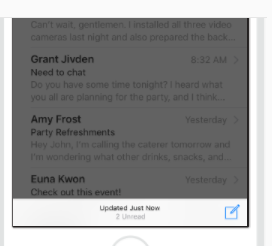
* **Data Entry:** Die Eingabe von Informationen, kann ein Prozess verlangsamen, in dem zu Beginn der App zu viele Eingaben verlangt werden. Die Benutzer können dadurch schnell entmutigt werden die App zu verwenden. Auf Grund dessen sollte die Dateneingabe zu effizient wie möglich gestaltet werden. Zum Beispiel können Pickers oder eine Tabelle anstelle eines Textfeldes stehen, um es durch eine Liste von vordefinierte Optionen, die Eingabe einer Antwort zu vereinfachen. Zusätzlich kann dies erleichtert werden, in dem die Listen und Pickern alphabetisch oder in einer anderen logischen Weise aufgebaut sind. Zusätzlich soll der Benutzer nicht dazu gezwungen werden Informationen, die automatisch oder die Berechtigung des Benutzers benötigen, zu übermitteln. Wenn aber wichtige Informationen erforderlich sind, die der Benutzer eintrage soll, muss darauf geachtet werden, dass eine Next- der Continue-Taste erst dann aktiviert wird, wenn alle erforderlichen Daten vollständig eingetragen worden sind. Die Eintragung sollte sofort nach der Eingabe überprüft werden, damit der Benutzer sie sofort korrigieren kann und so keine Frustration entsteht, da Fehler schnell beseitigt worden sind. (Apple)
* **Feedback:** Das hilft den Menschen zu wissen, was eine App macht, was Benutzer als nächstes tun können und verstehen die Ergebnisse der Aktionen. Unauffällige integrierte Status und anderen Arten von Feedback können dem Benutzer wichtige Informationen geben. Allerdings sollte eine zu hohe Anzahl an Warnungen vermieden werden, da die Benutzer ansonsten schnell anfangen zukünftige Warnungen zu ignorieren. Auf diesem Grund sollten nur wichtige Informationen vermittelt werden. (Apple)

Abbildung : iOS Feedback

**Visual Design**

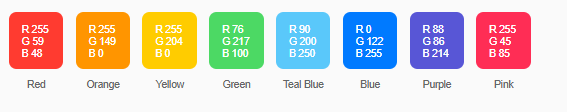
* **Branding:** Erfolgreiche App drücken ihre einzigartige Markenidentität durch eine intelligente Schriftart, Farbe und Bilder aus. Branding ist dazu da, um den Inhalt der App an den Benutzer zu vermitteln. Diese sollten in Maßen gehalten, sodass es nicht Benutzer von dem wesentlichen Inhalt ablenkt. Daher sollten die Brandings eingebettet und unauffällig sein. Die Marke drückt das App-Design aus, in dem zum Beispiel die Farben des App-Symbols sich in der App-Gestaltung wiederfindet.
* **Color:** Farben geben die Möglichkeit in der App mehr Vitalität zu schaffen, visuelle Kontinuität zu vermitteln, den Benutzer Feedback zu geben und den Benutzer bei der Visualisierung von Daten zu unterstützen. Des folgenden wird durch ein Farbschema erklärt, welche Farben sich besonders gut eignen und mit hellen und dunklen Hintergründe kombinierbar sind.

Abbildung : Color

Farben können die Kommunikation zum Benutzer verbessern. Allerdings funktioniert dies nur, wenn Farben sinnvoll genutzt werden. Menschen nehmen zum Beispiel ein rotes Dreieck als ein Warnsignal auf.

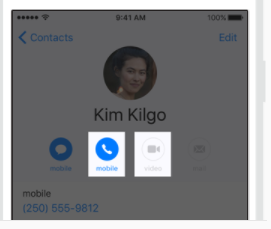
Bei der Farbwahl ist es ebenfalls wichtig zu beachten, dass die Farben in der App untereinander harmonieren und nicht vom wesentlichen ablenken. Deshalb ist es sinnvoll sich auf begrenzten Farbpalletten zu konzentrieren, die auch im App-Logo wiedergespiegelt wird. Ebenfalls vorteilhaft ist es sich für eine Hauptfarbe zu entscheiden, die die Interaktivität zwischen dem Benutzer und der App darstellt. Im Kalender zum Beispiel ist die interaktive Farbe häufig Rot. Verwirrungen kann durch die Verwendung der gleichen Farben für interaktive und nicht interaktive Elemente entstehen. Deshalb ist es wichtig Farben klar zu definieren.

Abbildung :Interaktive und Nicht-Interaktive Farben

Da Farben unterschiedliche Wirkungen erzeugen können, ist es wichtig, diese auch bei verschiedenen Lichtverhältnisse zu testen, um zukünftige Missverständnisse zu vermeiden. Weitere Missverständnisse können durch die Berücksichtigung von Farblinden, zum Beispiel rot-grün-Schwäche, verringert oder sogar verhindert werden, in dem besonders auf die Farbkombination geachtet wird. Ist dies aber nicht möglich, sollten Zustände mit Hilfe von anderen Werten oder Symbolen unterscheidbar gemacht werden. (Apple)

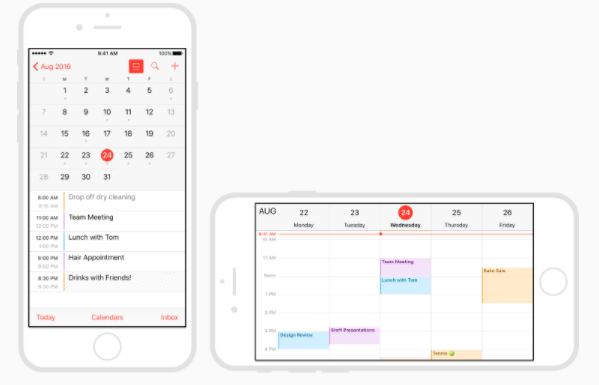
* **Layout:** In iOS können Schnittstellenelemente und Layouts so konfiguriert werden, dass sich die Form und Größe automatisch auf verschiedene Geräte verändert. Dies ist wichtig im Voraus zu planen und die App so zu entwerfen, dass der Kontext in jeder Ansicht gut aussieht. Sodass der Inhalt im Fokus steht und sich nicht durch andere Umgeben verändert. Denn das kann bei den Benutzern für Verwirrung und Frustration sorgen. In der Standardgröße sollte der Inhalt klar und deutlich zu erkennbar sein ohne dabei scrollen oder zoomen von wichtigen Informationen und Bilder zu müssen.

Abbildung Layouts

Die App sollte ein ganzheitliches Erscheinungsbild haben, in dem die Elemente und Funktionen allgemein ähnlich entworfen worden sind. Die Verwendung von unterschiedlichen Gewichtungen von Elementen und Funktionalitäten lässt die Bedeutung der Informationen anders wirken. Große Gegenstände fallen mehr auf und erscheinen wichtiger als die kleineren zu sein. Die Hauptinformation sollten am Besten in der oberen Hälfte des Bildschirmes platziert werden, sodass es bei unterschiedlichen Situationen, zum Beispiel beim Kochen, schnell erkennbar ist. Durch die Anordnung kann der Überblick besser beibehalten werden und dadurch wirkt die App ordentlicher und organisierter. Daher sollte der Abstand für interaktive Elemente ausreichend sein. Empfohlen wird zwischen den Elementen minimal 44pt\*44pt zu nehmen. Das Layout kann zusätzlich durch die Anpassung durch Textgrößenänderungen verbessert werden. Um einige Änderungen bei der Textgröße vorzunehmen, müsste teilweise das Layout erneut angepasst werden. (Apple)

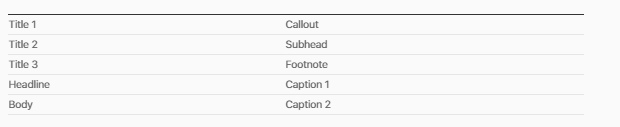
* **Typography:** San Francisco (SF) ist die Systemschrift von iOS. Die Schrift unterstützt die einfache Lesbarkeit, Klarheit und Konsistent von Texten. Diese Schrift wird hauptsächlich verwendet für lateinische, griechische und kyrillische Alphabete. Es gibt aber auch eine Vielzahl anderer Schriften von iOS. Wichtige Informationen können durch die Schriftart, Größe und Farbe hervorgehoben werden. Es ist sinnvoll sich für eine Schriftart zu entscheiden, denn durch zu viele Schriften kann die App unordentlich und unübersichtlich wirken. Wenn unterschiedliche Schriftwarten benutzt werden, sollte eine Schriftart einen bestimmten Textstil oder Inhalt ausdrücken, um eine optimale Lesbarkeit beibehalten. IOS enthalten die folgenden Textstile:

Abbildung : IOS Text Styles

Benutzerdefinierte Schriften werden ebenfalls von iOS unterstützt, allerdings muss sichergestellt werden, dass sie auch bei einer kleineren Schriftgröße gut lesbar sind. Dynamic Typ bietet eine zusätzliche Flexibilität, da der Benutzer seine bevorzugte Textgröße auswählen kann. Inhalte, die der Benutzer besonders interessiert, sollen durch eine Schrift leichter lesbar machen.

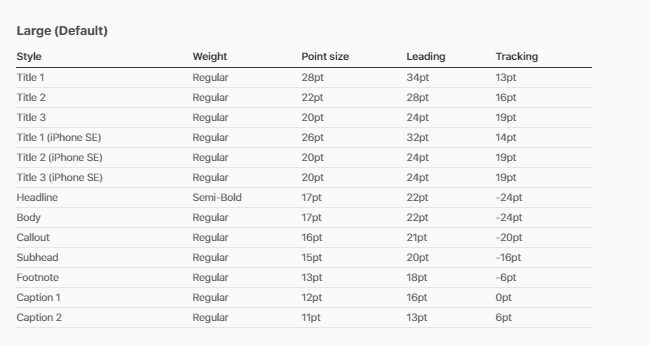


Abbildung Dynamic Typ

**Icons and Images**

* **Launch Screen:** Ein Startbildschirm erscheint sofort, wenn die App gestartet wird. Der Startbildschirm wird schnell durch den ersten Bildschirm der App ersetzt und erweckt den Eindruck, dass die App schnell und reaktionsschnell ist. Der Startbildschirm soll keinen künstlerischen Ausdruck sein, sondern der Benutzer soll eine schnellere Wahrnehmung der App erhalten. Jede App muss einen Startbildschirm liefern. Darauf sollte beim Entwerfen des Startbildschirms darauf geachtet werden, dass dieser fast identisch mit dem ersten Bildschirm ist. Da es sonst bei den Benutzern zur Konfusion führen kann. Auf Text sollte grundsätzlich verzichtet werden. Allerdings muss bei der Erstellung des Startbildschirms darauf geachtet werden, dass die Bildschirmgröße der Geräte variieren. (Apple)
* **System Icons:** IOS bietet viele kleine Symbole, die Funktionen und Arten von Inhalten darstellen. Es ist sinnvoll die eingebauten Ikonen zu benutzen, da sie den Menschen vertraut sind. Alternativ können auch Textbeschriftungen für Symbole stehen.

Abbildung Scroll Views

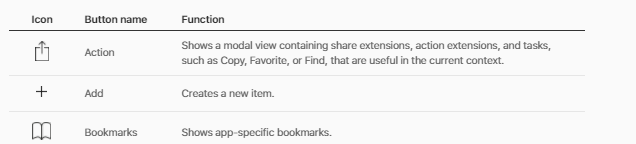


Abbildung Beispiele System Icons

**Views**

* **Maps:** Eine Kartenansicht ermöglicht die Anzeige von geografischen Daten in der App und unterstützt die meisten Funktionen, die von der eingebauten Maps App bereitgestellt werden. Eine Kartenansicht kann so konfiguriert werden, dass eine Standardkarte, Satellitenbilder oder beides angezeigt werden kann. Das Routing der App kann unterstützt werden, zum Beispiel bei einer Run-Tracking-App kann die Map-Ansicht verwendet werden, um die Route anzeigen zu lassen. (Apple)
* **Scroll Views:** Eine Scroll-Ansicht ermöglicht es den Benutzern Inhalte zu durchsuchen. Sobald die Benutzer wischen, schlagen, ziehen oder ähnliches folgt eine Scroll-Ansicht. Dabei werden die vorübergehenden Scrolling-Indikatoren an angezeigt, während sie benutzt wird. Das Aussehen der Scroll-Ansicht kann beliebig entworfen werden. (Apple)

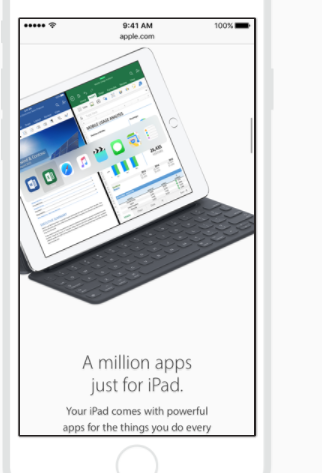
****

Abbildung Scroll Views

* **Table:** Tabellen werden genutzt, um große und kleine Mengen an Informationen sauber und effizient zu veranschaulichen. Im Allgemeinen sind Tabellen ideal für textbasierte Inhalte. Dabei sollte auf die Breite und Länge der Tabelle geachtet werden. Eine zu schmale Tabelle kann dafür sorgen, dass der Inhalt schwer lesbar ist. Eine Zeile mit komplexen Textdaten können zum Beispiel in Form von Bilder zur Verfügung gestellt werden. Diese Technik gibt den Benutzern nützliche Informationen und erhöht die wahrgenommene Reaktionsfähigkeit der App. (Apple)
* **Text Views:** Eine Textansicht zeigt mehrzeiligen Textinhalt. Diese können beliebig dargestellt werden und können durch das Scrollen erweitern werden, wenn der Inhalt außerhalb der Ansicht sich erstreckt. Standardmäßig ist der Inhalt innerhalb einer Textansicht linksbündig und die verwendete Schriftart ist schwarz. Wenn eine Textansicht bearbeitet werden kann, erscheint eine Tastatur, damit der Benutzer tippen kann. IOS bietet unterschiedliche Tastatur-Typen an, die jeweilig für eine andere Art von Eingabe geeignet sind. Auch wenn in der Textansicht unterschiedliche Schriften, Farben und Ausrichtungen verwendet werden, ist die Lesbarkeit des Inhaltes wichtig. (Apple)

**Controls**

* **Button:** Sie initiieren app-spezifische Aktionen, haben veränderbare Hintergründe und können einen Titel oder ein Symbol erhalten. IOS bietet eine Reihe von vordefinierten Button für unterschiedliche Anwendungsfälle. Es können aber auch benutzerdefinierte Button entworfen werden.
  + Systemtasten: Systemtasten erscheinen häufig in Navigationsleisten und Symbolleisten, aber sie können aber überall verwendet werden. Es können Verben oder Fall-Beschreibungen für den Titel genommen werden, um den Benutzer zu erklären, was passiert, wenn er auf den Button tippt. Die Titel sollten kurz bleiben, sodass der Button nicht allzu viel Platz auf dem Bildschirm einnimmt. Standardmäßig hat eine Systemtaste keinen Rahmen oder einen Hintergrund. Um einen Inhalt zu unterstreichen, kann das allerdings genutzt werden.
  + Detail Disclosure Buttons: Diese Buttons öffnen typischer weise eine modale Ansicht mit erhaltenden zusätzlichen Informationen oder Funktionalität, die sich auf ein bestimmtes Element auf dem Bildschirm bezieht. Sie können in jeder Art von Ansichten verwendet werden, aber hauptsächlich werden sie in Tabellen verwendet, um Informationen über die bestimmten Zeilen genauer zu erläutern.
  + Info Buttons: Ein Info-Button zeigt Konfigurationsdetails über eine App. Sie können in zwei Arten vorkommen- in dunkel und hell. Der ausgewählte Style sollte sich dem App Design anpassen und nicht auf dem Bildschirm verloren gehen. (Apple)
* **Labels:** Ein Label beschreibt ein Onescreen Interface-Element oder liefert eine kurze Nachricht. Es kann einen einfacher und gestylten Text erhalten. Wenn das Label an dem Stil der App angepasst wird oder benutzerdefinierten Schriften verwendet wird, sollte darauf geachtet werden, dass die Lesbarkeit dadurch nicht eingeschränkt wird. Es werden die vordefinierte Dynamic Typ von iOS empfohlen.

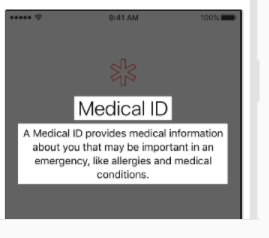


Abbildung Beispiel Label

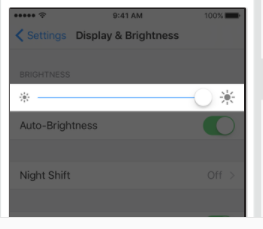
* **Pickers:** Ein Picker enthält eine oder mehrere scrollbare Listen mit verschiedenen Werten, wobei ein einziger Wert ausgesucht werden kann, der in einem dunkleren Text in der Mitte der Ansicht erscheint. Ein Picker wird oft am unteren Rand des Bildschirmes dargestellt. Die Höhe eines Pickers ist etwa die Höhe von fünfzeiligen Listenwerten. Die Breite des Pickers hat dieselbe Größe wie der Bildschirm. (Apple)
* **Sliders:** Ein Schieberegler ist eine horizontale Spur, die mit einem Finger gesteuert werden kann, um einen minimalen und maximalen Wert zu erreichen, zum Beispiel die Helligkeit des Bildschirms. Optimal wäre, wenn links und rechts neben dem Schieberegler Symbole angezeigt werden, die die Bedeutung der minimalen und maximalen Werte verdeutlichen. Das Design des Schiebereglers kann auf das Aussehen der App angepasst werden.

Abbildung Beispiel Sliders

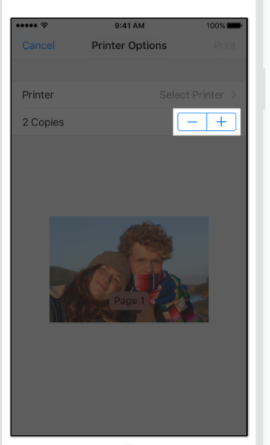
*  **Steppers:** Ein Stepper ist eine Zwei-Segment-Steuerung, die verwendet wird, um Werte zu erhöhen oder zu verringern. Standardmäßig besteht ein Stepper aus einem Plus- und Minus-Symbol. Diese Symbole können ebenfalls durch benutzerdefinierten Bildern ersetzt werden. Es sollte dem Benutzer verstehen, welche Werte sich durch die Nutzung des Steppers verändern. Das Intervall der Wertänderungen sollte klein sein, damit der gewünschte Wert schnell erreicht ist, zum Beispiel Anzahl der Kopien auf einem Druckbildschirm. (Apple)

Abbildung Steppers

* **Switches:** Ein Schalter kann visuell das Umschalten zwischen zwei Zuständen darstellen.

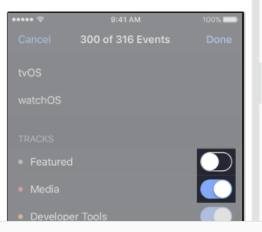
****

Abbildung Beispiel Switches

Das Aussehen eines Schalters kann auf das Design der App angepasst werden. Zusätzlich können Farben die zwei Zustände stärker unterstreichen. Ebenfalls können benutzerdefinierte Bilder verwendet werden. Schalter eignen sich besonders in Tabellenzeilen, um zum Beispiel eine Liste von Einstellungen mit Hilfe von Zuständen zu verändern. Das Hinzufügen von Labels an Schaltern ist redundant, da ein Schalter nur zwei Zustände annehmen kann. Es muss beachtet werden, dass die Verwendung von Swichtes die zugehörigen Schnittstellenelemente verwalten können, da zum Beispiel bei einer Deaktivierung des Flugzeugmodus automisch auch das Wifi auch ausgeschaltet wird.

* **Text Fields:** Ein Textfeld ist ein einzeiliges Feld mit einer festgelegten Höhe und oftmals mit abgerundeten Ecken. Sobald der Benutzer auf das Textfeld tippt, kommt eine Tastatur zum Vorschein. Der entsprechende Tastaturtyp sollte auf die jeweilige Art der Eingabe angepasst werden. In dem Textfeld sollten kleine Menge an Informationen angefordert werden, zum Beispiel Email-Adresse. Hinweise am Textfeld können helfen, den Benutzer den Zweck besser zu vermitteln. Button und Bilder können die Klarheit und Funktionalität des Textfeldes unterstützten. Generell können Stichwörter am linke Ende eines Textfeldes den Zweck eines Feldes besser erklären. Es kann ggf. eine Löschtaste am rechten Ende eines Textfeldes angezeigt werden, so dass vorhandene Elemente schnell und einfach erneut bearbeitet werden können.

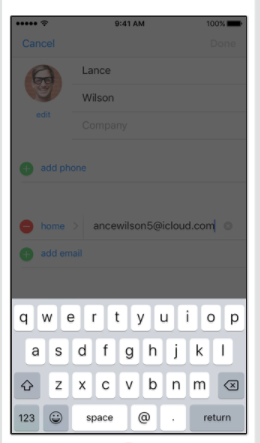
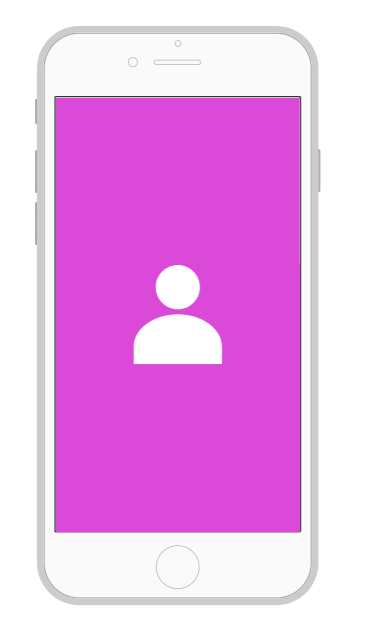


Abbildung Text Fields

Auf Basis der Wireframes und den iOS Guidelines wurden die Mockups erstellt. Es wurden folgende Mockups ausgesucht, wobei die Transformation von Wireframes mit der Berücksichtigung der Guidelines in Mockups am besten zu verdeutlichen sind. Dazu wurde das Tool [www.proto.io](http://www.proto.io) genutzt.

Zu Beginn öffnet sich sich der Startbildschirm. Der Startbildschirm hat eine wichtige Bedeutung, um den Wartevorgang der funktionsfähigen App zu beschleunigen. Der Startscreen muss nicht besonders künstlerisch durchdacht sein, sondern nur seinen Zweck erfüllen (s. Launchscreen).

Der Startbildschirm ist in einem kräftigen Lila mit dem App-Logo in weis (s. Abbildung 28). Das App-Logo ist eine Figur, die die Hauptrolle der App ist. Denn die Figur ist der Personaltrainer und begleitet den Benutzer während der Nutzung. Allerdings wurde auf den Wireframes die Figur mit einem Schriftzug dargestellt (s. Abbildung 5), die auf Grund von der Empfehlung der iOS Guidelines weggelassen worden ist (s. Launchscreen). Die Farbe Lila ist die Hauptfarbe der App, die sie sich zu einem gut auf hellen als auch auf dunklen Hintergründen anpassbar aber auch mehr Virtualität in die App bringt. Deshalb wurde auch entschieden, dass die lila ebenfalls die Interaktionsfarbe stellt, dich zum Beispiel immer auf Buttons wiederfinden (s. Abbildung 29). Die Figur ist auf dem Bilderschirm weiß, das hat den Grund, dass die Farbe zu der beschränkten Farbpallete angehört und das weiß sich hier besonders gut mit der Hintergrundfarbe kombinierbar ist. Die Farbpallete werden durch die Abbildungen 29 und 30 besonders deutlich. Es handelt es sich um die Farben: Schwarz, Grau, Dunkel Lila, helles Lila und weiß, weil sie gut mit dem Hintergrund harmonieren. Der Startbildschirm ähnelt dem ersten Bildschirm, um mögliche Verwirrung zu vermeiden (s. Launchscreen). Aus diesem Grund ist wieder der „Personaltrainer" mit bereits bekannten Farben.

Abbildung : Mockups Launch Screen

Auf dem Hintergrund ist ein Strand mit einem blauen Himmeln zu sehen zieht sich durch die ganze App (s. QUELLE) Denn laut Statistiken fühlen sich Menschen wohler, wenn sie draußen in der freien Natur Sport machen (s.QUELLE). Zusätzlich assoziieren einige Menschen einen Strand mit Bikinifigur. Sodass der Benutzer dazu motiviert wird sich mehr zu bewegen ****und sogar auch in der Natur aktiv zu sein und so die App aktiv genutzt wird. Weiterhin sollen die hellen Farben und den Anschein auf Sonne den Benutzer ebenfalls motivieren aktiv zu bleiben. Die von iOS vergebener Navigationsbar und Statusbar wurden berücksichtigt (s. Navigationsbar und Statusbar). Die Navigationsbar wurde durch Symbole erweitert. (s. Abbildung 30). Im Gegensatz zu den Wireframes wird statt einem Button „Weiter“ dies in der Navigationsbar integriert, denn der Benutzer ist mit den Symbolen, z.B. „Pfeil“ und „Zahnrad“ bereits vertraut und dies entspricht den iOS Richtlinien.

Eine weitere wichtige Richtlinie von iOS ist die Schriftfarbe und die Schriftart. Empfohlen wir Schriftfarbe schwarz. (s. Typograhy). Allerdings passt schwarz nicht zu dem restlichen Design und das Gesamtbild harmonieren soll, wurde ein dunkles grau ausgewählt. Sodass die Information gut lesbar ist. Auf lila Buttons ist die Schriftfarbe Weiß, dadurch ist die Schrift besser leserlich und klar. Die benutzte Schriftart ist in der App San Francisco (SF), weil diese auch in den iOS Richtlinien empfohlen wurde (s. Typography). Die Sprechblasen von dem Personaltrainer sollen die individuellen Förderung demonstrieren als wäre der Personaltrainer persönlich für die Benutzer da. Es wurde bewiesen, dass den meisten Menschen wichtig ist, dass sie beim Sport gut und individuell beraten werden. Der Personaltrainer stellt über die ganze App fragen nach persönlichen Daten, Systemeinstellungen etc. Es wird immer nur nach einer geringen Menge an Informationen gefragt, um den Prozess nicht zu verlangsamen (s. Data Entry). Durch zum Beispiel Schiebereger, Pickers, Checkboxen, Tabellen und Textfelder kann dem Benutzer der Benutzer eine Antwort auf die Fragen erleichtern und Zeit sparen (s. Abbildung 29, 30, 31). Sodass der Benutzer nicht entmutigt wird die App weiterhin zu benutzen. Der Benutzer wird dazu aufgefordert seinen Vornamen in das Textfeld einzugeben. Den Vornamen nur einzugeben, ist besonders sinnvoll, da der Anmeldevorgang so schnell und einfach gestaltet werden soll. Außerdem ist es wichtig, den Benutzer nicht zu verpflichten bevor er sich die App angeschaut hat (s.Authenfication). Durch die Abfrage des Vornamens wirkt die App invidueller. Zusätzlich ist es wichtig, die eingegeben Daten sofort zu überprüfen bevor der Benutzer auf die nächste Seite gelangt, sodass eine spätere Frustration durch späteres Korrigieren vermieden werden kann (s. Entry).

Abbildung : Mockups Willkommen

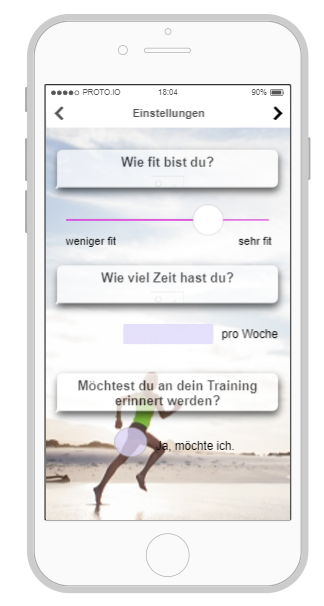
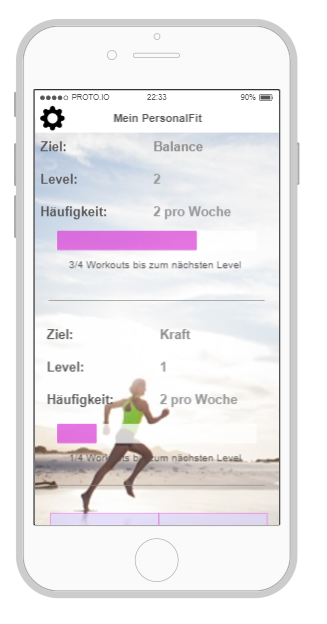


Abbildung : Mockups Einstellungen

Abbildung 31: Mockups Mein PersonalFit

Die Abbildung 31 veranschaulicht eine Scroll-Ansicht, um weitere Inhalte zu erweitern und die wichtigsten Informationen auf einem Screen zu haben. (s. Screen-View)

Insgesamt haben die iOS Richtlinien dabei geholfen aus den vorherigen Entwürfen eine realitätsnahe Version der zu entwickelnden App zu entwerfen. Einige Richtlinien wurden nicht mehr verwendet, da es auf Grund des Programms nicht visualisierbar war.

SCHLUSS:

## SWOT-Analyse

...

# Zusammenfassung & Fazit

...

# Ausblick

...

# Literaturverzeichnis

I Aichele C, Schönberger M (Hrsg) (2014) App4U; Mehrwerte durch Apps im B2B und B2C. Springer Vieweg, Wiesbaden

Barry W. Boehm\*, Terence E. Gray, and Thomas Seewaldt PROTOTYPING VS. SPECIFYING: A MULTI-PROJECT EXPERIMENT

eBusiness-Lotse (2015) Von der Idee zur eigenen App; Ein praxisorientierter Leitfaden für Unternehmer mit Checkliste. http://ikt-forum.de/sites/default/files/Online-Version\_Von\_der\_Idee\_zur\_eigenen\_App.pdf

G. Pomberger, W. Pree, A. Stritzinger Methoden und Werkzeuge für das Prototyping und ihre Integration

GEDA GEDA 2012 Studie; Gesundheit in Deitschland aktuell 2012. http://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GBEDownloadsF/Geda2012/Sportliche\_Aktivitaet.pdf?\_\_blob=publicationFile

GEDA GEDA 2012 Studie; Gesundheit in Deitschland aktuell 2012. http://www.gbe-bund.de/pdf/GEDA\_2012\_gesundh\_einschraenkungen.pdf

Pahmeier I (2008) Sportliche Aktivität aus der Lebenslaufperspektive. Zeitschrift fur Gerontologie und Geriatrie 41:168–176. doi:10.1007/s00391-008-0543-x

Prof. Dr. Ingo Froböse, Dr. Birgit Wallmann-Sperlich Der DKV Report „Wie gesund lebt Deutschland?“ 2016. file:///C:/Users/Kimngan/Downloads/20160808-DKV-Report-2016-Studienbericht.pdf

Robert Koch Institut Statisches Bundesamt Körperliche Aktivität. https://campus.uni-muenster.de/fileadmin/einrichtung/epi/download/vorlesungen/Literatur/Gesundheitsberichterstattung\_Koerperliche\_Aktivitaet.pdf

Spitta T (1989) Software Engineering und Prototyping; Eine Konstruktionslehre für administrative Softwaresysteme. Springer, Berlin, Heidelberg

m aktuellen Dokument sind keine Quellen vorhanden.

Eidesstattliche Versicherung

„Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.“

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ort, Datum, Unterschrift

Literatur

Aichele C, Schönberger M (Hrsg) (2014) App4U; Mehrwerte durch Apps im B2B und B2C. Springer Vieweg, Wiesbaden

APP4U Mehrwete druch Apps im B2B and B2C

Apple Human Interface Guideline; Color. https://developer.apple.com/ios/human-interface-guidelines/visual-design/color/

Apple Human Interface Guideline; Maps. https://developer.apple.com/ios/human-interface-guidelines/ui-views/maps/

Apple Human Interface Guideline; Layout. https://developer.apple.com/ios/human-interface-guidelines/visual-design/layout/

Apple Human Interface Guideline; Launch Screen. https://developer.apple.com/ios/human-interface-guidelines/graphics/launch-screen/

Apple Human Interface Guidelines; Buttons. https://developer.apple.com/ios/human-interface-guidelines/ui-controls/buttons/

Apple Human Interface Guidelines; Authentification. https://developer.apple.com/ios/human-interface-guidelines/interaction/authentication/

Apple Human Interface Guidelines; Steppers. https://developer.apple.com/ios/human-interface-guidelines/ui-controls/steppers/

Apple Human Interface Guidelines; Scroll Views. https://developer.apple.com/ios/human-interface-guidelines/ui-views/scroll-views/

Apple Human Interface Guidlines; Text Views. https://developer.apple.com/ios/human-interface-guidelines/ui-views/text-views/

Apple Human Interface Guidlines; Pickers. https://developer.apple.com/ios/human-interface-guidelines/ui-controls/pickers/

Apple Human Interface Guidlines; Tables. https://developer.apple.com/ios/human-interface-guidelines/ui-views/tables/

Apple Human Interface Guidlines; Data Entry. https://developer.apple.com/ios/human-interface-guidelines/interaction/data-entry/

Apple Human Interface Guidslines; Feedback. https://developer.apple.com/ios/human-interface-guidelines/interaction/feedback/

Barry W. Boehm\*, Terence E. Gray, and Thomas Seewaldt PROTOTYPING VS. SPECIFYING: A MULTI-PROJECT EXPERIMENT

eBusiness-Lotse (2015) Von der Idee zur eigenen App; Ein praxisorientierter Leitfaden für Unternehmer mit Checkliste. http://ikt-forum.de/sites/default/files/Online-Version\_Von\_der\_Idee\_zur\_eigenen\_App.pdf

G. Pomberger, W. Pree, A. Stritzinger Methoden und Werkzeuge für das Prototyping und ihre Integration

GEDA GEDA 2012 Studie; Gesundheit in Deitschland aktuell 2012. http://www.gbe-bund.de/pdf/GEDA\_2012\_gesundh\_einschraenkungen.pdf

GEDA GEDA 2012 Studie; Gesundheit in Deitschland aktuell 2012. http://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GBEDownloadsF/Geda2012/Sportliche\_Aktivitaet.pdf?\_\_blob=publicationFile

Pahmeier I (2008) Sportliche Aktivität aus der Lebenslaufperspektive. Zeitschrift fur Gerontologie und Geriatrie 41:168–176. doi:10.1007/s00391-008-0543-x

Prof. Dr. Ingo Froböse, Dr. Birgit Wallmann-Sperlich Der DKV Report „Wie gesund lebt Deutschland?“ 2016. file:///C:/Users/Kimngan/Downloads/20160808-DKV-Report-2016-Studienbericht.pdf

Robert Koch Institut Statisches Bundesamt Körperliche Aktivität. https://campus.uni-muenster.de/fileadmin/einrichtung/epi/download/vorlesungen/Literatur/Gesundheitsberichterstattung\_Koerperliche\_Aktivitaet.pdf

Spitta T (1989) Software Engineering und Prototyping; Eine Konstruktionslehre für administrative Softwaresysteme. Springer, Berlin, Heidelberg