



Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Medientechnik
Forschungsgruppe Verteilte Multimodale Informationsverarbeitung
Prof. Dr.-Ing. Eckehard Steinbach

Entwurf und Implementierung einer Rollator- Trainingsanwendung für Android

Design and Implementation of a Rollator Training App- lication for the Android Platform

Michael Hoffmann

Bericht

Verfasser:	Michael Hoffmann
Anschrift:	[REDACTED]
Matrikelnummer:	[REDACTED]
Professor:	Prof. Dr.-Ing. Eckehard Steinbach
Betreuer:	Dipl.-Ing. (Univ.) Stefan Diewald
Beginn:	17.03.2014
Abgabe:	23.05.2014

Inhaltsverzeichnis

Inhaltsverzeichnis	ii
1 Einleitung	1
2 Aktueller Stand der Forschung und Technik	3
2.1 Inhalte	3
2.2 Altersbedingte Einschränkungen	3
2.2.1 Visuelle Einschränkungen	4
2.2.2 Einschränkungen des Gedächtnisses	4
2.2.3 Motorische Einschränkungen	5
2.3 Smartphone für ältere Menschen	5
3 Entwurf der Applikation	7
3.1 Inhalte	7
3.2 Anforderungen an die Applikation	7
3.3 Speichern und Abfrage der Übungen und Trainings	8
3.4 Layout	12
3.5 Ablaufdiagramm	13
4 Implementierung der Applikation	16
4.1 Inhalte	16
4.2 Quellcode	16
4.2.1 de.tum.ei.vmi.passage.rollatortraining	16
4.2.2 de.tum.ei.vmi.passage.rollatortraining.customviews	18
4.2.3 de.tum.ei.vmi.passage.rollatortraining.navigationdrawer	18
4.2.4 de.tum.ei.vmi.passage.rollatortraining.service	18
4.2.5 de.tum.ei.vmi.passage.rollatortraining.sqlite	18
4.2.6 de.tum.ei.vmi.passage.rollatortraining.utils	18
4.2.7 de.tum.ei.vmi.passage.rollatortraining.xmlparser	18
5 Beschreibung der Trainingsanwendung	19
6 Zusammenfassung	25

A Informationen zur Auswertung der SQLite Datenbank	26
Abbildungsverzeichnis	29
Tabellenverzeichnis	30
Abkürzungsverzeichnis	31
Literaturverzeichnis	32

Kapitel 1

Einleitung

Die Zahl von älteren Menschen in Europa steigt kontinuierlich von Jahr zu Jahr. So soll der Bevölkerungsanteil der über 65-jährigen in Europa vom Jahr 2010 bis 2050 um mehr als 11 % steigen¹. Die Anzahl an Senioren in Europa steigt von 119 Millionen auf 194 Millionen. Weltweit wird im selben Zeitraum die Zahl der 65-jährigen von 500 Millionen auf 1,5 Milliarden ansteigen. Die meisten davon haben wenig bis gar keine Erfahrung im Umgang mit Smartphones und können somit die vielen Vorteile davon nicht im Alltag nutzen. Beispielsweise werden Smartphone-Applikationen immer häufiger dazu eingesetzt, um die persönliche Fitness zu überwachen oder steigern [1]. Einige Apps bieten mithilfe von vorgefertigten Trainingsplänen und Übungen ein fertiges Trainingsprogramm und das Smartphone dient als Bindeglied zwischen dem Nutzer und der Anwendung. Gerade für ältere Leute ist viel Bewegung für das Wohlbefinden und die Gesundheit wichtig, daher könnte das Smartphone als Trainingshilfe eingesetzt werden [2].

Rollatoren sind bei Menschen im höheren Lebensalter im Alltag oft ein wichtiges Mobilitätshilfsmittel und können somit für unterschiedliche Fitness-Übungen verwendet werden. Sportmediziner der Technischen Universität München haben einige Übungen extra für Rollatoren entworfen und mithilfe einer altersgerechten Smartphone-Applikation soll eine eigenständige Bearbeitung des Trainings ermöglicht werden. Das Smartphone wird am Rollator befestigt und die älteren Leute können die Übung am Gerät durchführen.

Im Rahmen der Forschungspraxis wurden der Entwurf und die Implementierung einer solchen Android-Applikation für ein Smartphone bearbeitet. Die App soll für ältere Menschen leicht zu bedienen sein und die Übungen einfach und verständlich den Benutzern darstellen. Neben Trainingseinheiten mit vorgefertigten Übungsabfolgen soll auch die manuelle Auswahl und Durchführung der Übungen möglich sein. Der Trainingsverlauf soll einsehbar sein und Benachrichtigungen vom Smartphone angezeigt werden, wenn ein Training ansteht. Die Entwicklung der Trainingsanwendung wurde im Rahmen des Projekts PASSAge durchgeführt [3–5].

Die Arbeit besteht aus drei Kapiteln. Im ersten Kapitel werden die Einschränkungen von älteren

¹<http://de.statista.com/statistik/daten/studie/224010/umfrage/prognostizierte-bevoelkerungsentwicklung-der-ueber-65-jaehrigen/>

Menschen und Anforderungen an das Smartphone thematisiert. Das zweite Kapitel befasst sich mit dem Entwurf der Applikation und das letzte Kapitel mit der Implementierung.

Kapitel 2

Aktueller Stand der Forschung und Technik

2.1 Inhalte

In diesem Kapitel wird der aktuelle Stand der Forschung und Technik betrachtet. Diese Erkenntnisse werden anschließend für den Entwurf und die Implementierung der Applikation verwendet. Ziel ist eine Smartphone-App, die auch ältere Menschen, ohne Erfahrung im Umgang mit solchen Geräten, problemlos bedienen können.

2.2 Altersbedingte Einschränkungen

Ältere Menschen haben einige Einschränkungen aufgrund des hohen Alters. Diese lassen sich in die drei Kategorien Sinne, Körper und Geist eingliedern, wie in Tabelle 2.1 zu sehen ist¹.

Tabelle 2.1: Einschränkungen im Alter

Sinne	Körper	Geist
Sehen	Beweglichkeit	Informationsverarbeitung
Hören	Kraft	Gedächtnis
Fühlen/Tasten	Fingerfertigkeit	Reaktion
Riechen/Schmecken		Koordination

Dadurch ergeben sich besondere Anforderungen für mobile Applikationen für Smartphones, auf die man als Entwickler achten muss. Durch das eingeschränkte Sehen können kleinere Schriften und Elemente schwieriger gelesen und erkannt werden. Die begrenzte Fingerfertigkeit und Beweglichkeit erschwert die Bedienung an Geräten, wie beispielsweise das Drücken kleiner Schaltflächen.

Ein weiteres Problem stellt die verminderte Informationsverarbeitung von älteren Menschen dar, die mehr Zeit in Anspruch nimmt. Ab dem Alter von ungefähr 20 Jahren sinkt die fluide Intelligenz.

¹<http://www.emk.tu-darmstadt.de/~weissmantel/sensi/kap4.pdf>

Dies hat zur Folge, dass das Abspeichern und Abrufen von Informationen aus dem Gehirn länger dauert und das Erlernete schneller wieder vergessen wird².

2.2.1 Visuelle Einschränkungen

Einschränkungen der Sicht durch Altersweitsichtigkeit, verminderte Sehschärfe oder Augenkrankheiten sind ein großes Hindernis für die Bedienung von Benutzungsoberflächen auf kleinen Bildschirmen. Diese Einschränkungen treten bereits im Alter von 35 bis 40 Jahren auf [6]. Darüber hinaus werden animierte Icons oder Schaltflächen von älteren Menschen oft als störend empfunden oder gar nicht wahrgenommen [7]. Eine Animation führt nicht zu einem besseren Verständnis der Funktion eines Bedien- oder Anzeigeelementes und verlängert die Zeit bis zur Wahrnehmung des Elements.

Aufgrund der eingeschränkten visuellen Wahrnehmung werden folgende Punkte beim Entwurf der Applikation beachtet:

- Keine animierten Icons, Textfelder oder Schaltflächen.
- Gut lesbare Schrift in ausreichender Größe.
- Einfacher Hintergrund ohne störende Muster.
- Ausreichende Größe der Schaltflächen für bessere Erkennbarkeit.

2.2.2 Einschränkungen des Gedächtnisses

Ältere Menschen haben vermehrt Probleme Informationen aufzunehmen, abzuspeichern und wiederzugeben. Eine schnellere Ermüdung und ein frühzeitiger Verlust der Aufmerksamkeit bei Tätigkeiten tritt häufig auf [8].

Diese Einschränkungen bedingen einige Anforderungen, die bei der Entwicklung der mobilen Applikation beachtet werden sollen:

- Leicht nachvollziehbare Navigationsstruktur mit wenigen Elementen.
- Jederzeit ersichtliche Position innerhalb der Software.
- Auswahl von Grafiken bezüglich Relevanz und Informationsgehalt, nicht wegen der Ästhetik.
- Erinnerungsfunktionen zur Erinnerung an bestimmte Ereignisse.

²http://fazit-forschung.de/fileadmin/_fazit-forschung/downloads/fachtagung_lt1_gesamttext.pdf#page=35

2.2.3 Motorische Einschränkungen

Motorische Fähigkeiten beschreiben die Bewegungsmöglichkeiten des Körpers. Im Laufe des Alters verschlechtert sich vor allem die Feinmotorik [9]. Dies hat zur Folge, dass die Bedienung von Schaltflächen auf Touchscreens eingeschränkt wird. Um die Fehlerwahrscheinlichkeit bei der Bedienung von Schaltflächen möglichst gering zu halten, sollten diese entsprechend groß dimensioniert werden. Eine Studie der India University ergab, dass Schaltflächen mit einer Seitenlänge von 10-25 mm von Senioren im Alter zwischen 75 und 85 Jahren bevorzugt werden [10]. Abbildung 2.1 zeigt die bevorzugte Größe von Schaltflächen, die im Rahmen der Studie ermittelt wurden.

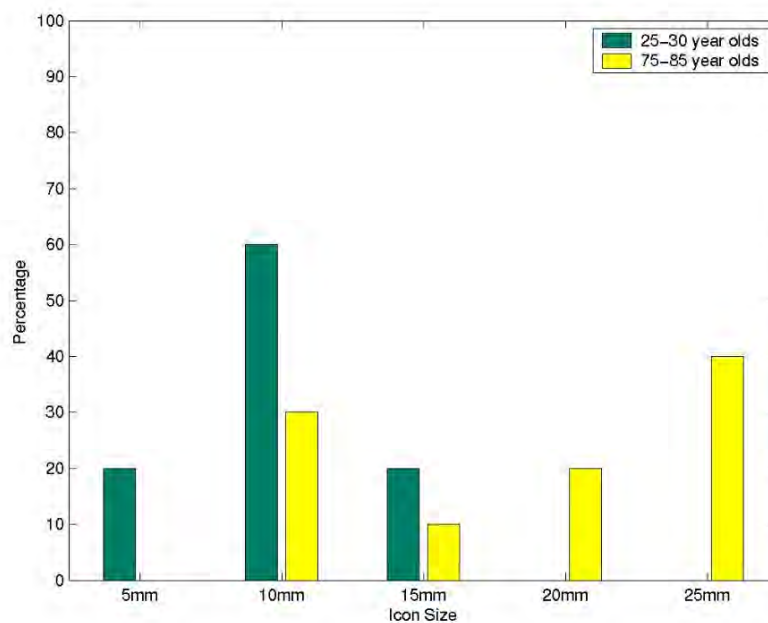


Abbildung 2.1: Bevorzugte Größe der Schaltflächen, Quelle: [10]

Daraus ergeben sich folgende Anforderungen für die Trainingsanwendung:

- Bevorzugte Größe von Schaltflächen einhalten.
- Smartphone mit großem Display verwenden, um Inhalte in größerer Darstellung anzeigen zu können.
- Längere Reaktionszeit der Benutzer miteinplanen.

2.3 Smartphone für ältere Menschen

Ein Smartphone eignet sich in vielerlei Hinsicht für ältere Menschen. Durch das große Display können Inhalte gut vermittelt werden. Durch die Verwendung des Touchscreens können Be-

dienelemente am Bildschirm frei platziert und in der Größe angepasst werden. Dadurch kann eine Benutzeroberfläche entworfen werden, die an die Bedürfnisse der Nutzer angepasst ist.

Durch den Touchscreen können beispielsweise Wischgesten in der Applikation verwendet werden, was für ältere Menschen oft eine einfachere Art der Bedienung von Software darstellt [11].

Kleine, kaum lesbare Beschriftungen und zu kleine Bedienelemente, die nur schwer aktiviert werden können, führen schnell zu Frust beim Anwender und sollten daher vermieden werden [12]. Die Navigation in der Anwendung soll einfach zu handhaben und der aktuelle Fortschritt und Zustand erkennbar sein.

Smartphones eignen sich sehr gut für Lern- und Trainingsanwendungen [13], da die integrierte Technik viele Möglichkeiten zum Aufzeichnen und Wiedergeben von Trainingsaktivitäten bietet. So wurde beispielsweise an der Technischen Universität München ein Smartphone mit einem Kippbrett gekoppelt und die Sensoren des Mobiltelefons zur Aufzeichnung und Analyse von Trainings verwendet [14]. In Google's Play Store³ und Apple's App Store⁴ findet man in der Top 10 der kostenpflichtigen Apps die App "Runtastic Pro Laufen & Fitness", die sich großer Beliebtheit erfreut. Diese mobile Applikation zeichnet verschiedene Trainingsaktivitäten wie Laufen oder Fahrradfahren auf und nutzt dazu Sensoren wie den GPS-Sensor oder einen separat erwerbbaaren Brustgurt zur Pulsmessung.

³https://play.google.com/store/apps/collection/topselling_paid

⁴<http://www.apple.com/de/itunes/charts/paid-apps/>

Kapitel 3

Entwurf der Applikation

3.1 Inhalte

In diesem Kapitel wird auf die Implementierung der Applikation eingegangen. Dabei wird beschrieben, wie bestimmte Aufgabenstellungen programmatisch gelöst wurden und der generelle Aufbau der Software aussieht.

3.2 Anforderungen an die Applikation

Folgende Anforderungen wurden im Voraus an die mobile Anwendung gestellt:

- Vorgegebene Übungen sollen in Form von Trainingseinheiten oder einzeln auswählbar sein.
- Die Übungen steigern sich im Verlauf des Trainings.
- Am Ende einer Trainingseinheit soll eine Bewertung dafür abgegeben werden können.
- Videos und Bilder zu den Übungen sollen auf der SD-Karte des Smartphones hinterlegt werden.
- Die App soll Benachrichtigungen erzeugen, die an das Training erinnern.
- In einer Datenbank soll die Dauer der Übung, der Abbruch einer Übung, die Startzeit, die aktuelle Trainingseinheit und Übung, der Start durch eine Benachrichtigung und das Wiederholen einer Trainingseinheit gespeichert werden.
- Die Datenbank soll exportiert werden können, damit diese mit einem entsprechenden Programm ausgewertet werden kann.
- Die App soll auf die Anzeige im Querformat ausgelegt sein.

Für die Entwicklung der Applikation stand ein Android Telehealth Smartphone „hFon plus“ der Firma HMM Diagnostics GmbH zur Verfügung. Das hFon plus besitzt einen 4.7 Zoll großen

Touchscreen mit einer Auflösung von 540x960 Pixel (WVGA) und Android in der Version 4.1.2. Ferner verfügt das Gerät über einen integrierten Speicher von 512 MB RAM, 4 GB internen Speicher, einen 3D Beschleunigungssensor, GPS und unterstützt eine micro SD Karte bis 32 GB. Ein zusätzlicher Modulsteckplatz eignet sich zum Einsatz von Blutzuckermessmodulen.

3.3 Speichern und Abfrage der Übungen und Trainings

Zur Speicherung und Abfrage der Übungen und Trainingseinheiten fiel die Wahl auf das Extensible Markup Language (XML)-Format. Das Android-Betriebssystem verfügt intern über einen XML-Parser, mit dem Informationen aus einer XML-Datei gelesen werden können. Google empfiehlt die Verwendung des XmlPullParser, der ein besonders effizienter und leicht erweiterbarer XML-Parser für Android ist¹.

Eine Trainingseinheit besteht aus einer vorbestimmten Abfolge von Übungen. Daher werden in einer XML-Datei (exercises.xml) alle Übungen hinterlegt und wie folgt definiert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML File to store exercises of RollatorTraining app -->
<exercises>
  <exercise>
    <exercise_id>1</exercise_id>
    <stepCount>2</stepCount>
    <exercise_name>Geschlossene Fuesse</exercise_name>
    <repetitions>Dauer: 20 Sekunden</repetitions>
    <duration></duration>
    <clockAvailable>0</clockAvailable>
    <pathToVideo>/RollatorTraining/exercise1/Video_Uebung_1.mp4
      </pathToVideo>
    <step>
      <step_id>1</step_id>
      <exercise_id>1</exercise_id>
      <step_description>Bremsen feststellen. An beiden
        Griffen festhalten. Fuesse schliessen.</
        step_description>
      <path_to_picture>/RollatorTraining/exercise1/ex1_step1.
        jpg</path_to_picture>
    </step>
  </exercise>
</exercises>
```

¹<http://developer.android.com/training/basics/network-ops/xml.html>

```

    <step>
      <step_id>2</step_id>
      <exercise_id>1</exercise_id>
      <step_description>Haende ca. 5 cm von Rollatorgriffen
        anheben. Gleichgewicht halten.</step_description>
      <path_to_picture>/RollatorTraining/exercise1/ex1_step2.
        jpg</path_to_picture>
    </step>
  </exercise>
  .
  .
  .
</exercises>

```

Eine Trainingseinheit beinhaltet eine Auflistung der zu absolvierenden Übungen und die jeweilige Dauer der Übung. Anhand des vordefinierten eindeutigen Identifikators (IDs) jeder Übung, kann eine Zuordnung zwischen den beiden XML-Dateien erfolgen. Die XML-Datei mit hinterlegten Trainingseinheiten (levels.xml) sieht wie folgt aus:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML File to store levels of RollatorTraining app -->
<levels>
  <level>
    <lvl_id>1</lvl_id>
    <lvl_name>Trainingseinheit 1</lvl_name>
    <exerciseCount>10</exerciseCount>
    <exercise>
      <exercise_id>1</exercise_id>
      <repetitions>Dauer: 20 Sekunden</repetitions>
    </exercise>
    <exercise>
      <exercise_id>2</exercise_id>
      <repetitions>Dauer: 20 Sekunden pro Bein</
        repetitions>
    </exercise>
    <exercise>
      <exercise_id>3</exercise_id>
      <repetitions>Dauer: 20 Sekunden pro Bein</

```

```
        repetitions>
</exercise>
<exercise>
    <exercise_id>5</exercise_id>
    <repetitions>10 Wiederholungen</repetitions>
</exercise>
<exercise>
    <exercise_id>6</exercise_id>
    <repetitions>10 Wiederholungen</repetitions>
</exercise>
<exercise>
    <exercise_id>11</exercise_id>
    <repetitions>Dauer: 20 Sekunden</repetitions>
</exercise>
<exercise>
    <exercise_id>12</exercise_id>
    <repetitions>7 Wiederholungen</repetitions>
</exercise>
<exercise>
    <exercise_id>13</exercise_id>
    <repetitions>10 Wiederholungen pro Bein</
    repetitions>
</exercise>
<exercise>
    <exercise_id>14</exercise_id>
    <repetitions>10 Wiederholungen</repetitions>
</exercise>
<exercise>
    <exercise_id>16</exercise_id>
    <repetitions>10 Wiederholungen</repetitions>
</exercise>
</level>
.
.
.
</levels>
```

Tabelle 3.1 zeigt die vorgegebene Verteilung der Übungen in den jeweiligen Trainingseinheiten.

In den Trainingseinheiten 13-24 und 25-36 wechselt sich der 1. und 2. Block an Übungen abwechselnd ab. Auch die Trainingsdauer und Anzahl der Wiederholungen variiert und wird von den Sportmedizinern vorgegeben. Die Übungen in den Trainingseinheiten, die sich wiederholen, werden durchgemischt, damit etwas mehr Abwechslung beim Training vorhanden ist. So würden beispielsweise die Trainingseinheiten 1-6 ansonsten sechsmal die gleiche Abfolge an Übungen zur Folge haben, was zur Demotivation des Anwenders führen könnte.

Tabelle 3.1: Struktur der Trainingseinheiten

Trainingseinheiten	Übungen
1-6	1, 2, 3, 5, 6, 11, 12, 13, 14, 16
7-12	1, 2, 3, 5, 6, 8, 10, 11, 12, 13, 14, 16
13-24	1. Block: 2, 3, 4a, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16
13-24	2. Block: 2, 3, 4a, 5, 6, 8, 9, 11, 12, 13, 14, 15, 17
25-36	1. Block: 3, 4a, 4b, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16
25-36	2. Block: 3, 4a, 4c, 5, 6, 8, 9, 10, 11, 12, 13, 15, 17

Sollten Übungen geändert, entfernt oder hinzugefügt werden, kann dies in der `exercises.xml`-Datei erfolgen, ohne diese in jeder Trainingseinheit einzeln editieren zu müssen. Andererseits kann auch die Reihenfolge und Anzahl der Übungen in der `levels.xml`-Datei leicht bearbeitet werden.

Zur Speicherung der absolvierten Trainingseinheiten oder einzelnen Übungen, fiel die Wahl auf eine SQLite-Datenbank. Dabei handelt es sich um ein Open Source-Datenbanksystem, welches zur Laufzeit nur etwa 250 kB an Speicher benötigt und somit optimal für das Android-Betriebssystem geeignet ist. SQLite ist fest in die Android-Plattform integriert und kann einfach genutzt werden. Die Wahl der zu speichernden Werte in der Datenbank wurde durch die Anforderungen aus Kapitel 3.2 bestimmt. Tabelle 3.2 zeigt die Struktur der geplanten Datenbank.

Tabelle 3.2: SQLite Datenbank Struktur

FIELD	TYPE	KEY
id	INTEGER	Primary Key
training_type	TEXT	-
start_time_date	TEXT	-
current_level	INTEGER	-
current_exercise	INTEGER	-
exercise_name	TEXT	-
feedback_mood	TEXT	-
duration	TEXT	-
user_canceled?	INTEGER	-
started_from_notification?	INTEGER	-
level_repeated?	INTEGER	-

Die Felder „user_canceled?“, „started_from_notification?“ und „level_repeated?“, werden als Integer-Werte abgespeichert, sollen aber boolesche Werte darstellen. Da man in eine SQLite-

Datenbank keine booleschen Wert wie „true“ und „false“ ablegen kann, werden Integer-Werte mit 0 (false) und 1 (true) verwendet².

Ein Eintrag in die Datenbank soll in folgenden Fällen angelegt werden:

- Eine Trainingseinheit wurde beendet oder abgebrochen.
- Eine einzelne Übung wurde beendet oder abgebrochen.

Der Export der SQLite-Datenbank erfolgt im Comma-separated values (CSV)-Format, da diese Dateien mit gängigen Editoren oder Tabellenkalkulationsprogrammen geöffnet und ausgewertet werden können.

3.4 Layout

Durch die Vorgabe des Querformats aus Kapitel 3.2 muss ein passendes Layout entworfen werden, das die geringe Höhe und größere Breite des Displays optimal ausnutzt. Im Startbildschirm soll der Fortschritt der aktuellen Trainingseinheit und die dazugehörigen Übungen dargestellt werden. Für die Fortschrittsanzeige wird eine ProgressBar und für die Darstellung der Übungen eine HorizontalScrollView verwendet, siehe Abbildung 3.1.

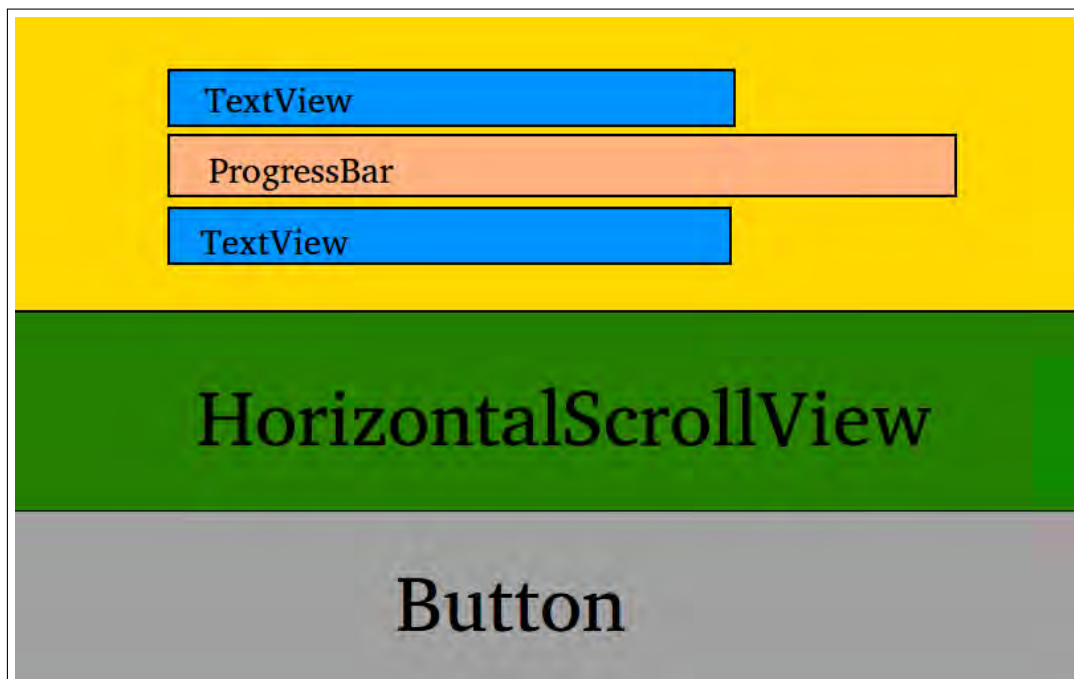


Abbildung 3.1: Überblick über das Layout vom Startbildschirm

² <http://www.sqlite.org/datatype3.html>

Eine einzelne Übung wird ebenfalls durch eine horizontal scrollbare Ansicht dargestellt. Die ActionBar wird zur Anzeige des Übungsnamens, der aktuellen Trainingseinheit und der aktuellen Übung genutzt. Im unteren Bereich wird die Wiederholung oder Dauer der aktuellen Übung, sowie die verstrichene Übungsdauer nach dem Start angezeigt. In der HorizontalScrollView werden dynamisch das Video in einer VideoView, die einzelnen Schritte der Übung mit ImageView und Buttons zum Starten, Pausieren und Abschließen erzeugt. Abbildung 3.2 zeigt den schematischen Aufbau einer Übung. Das Layout eines dynamischen erzeugten Elements ist in Abbildung 3.3 dargestellt.

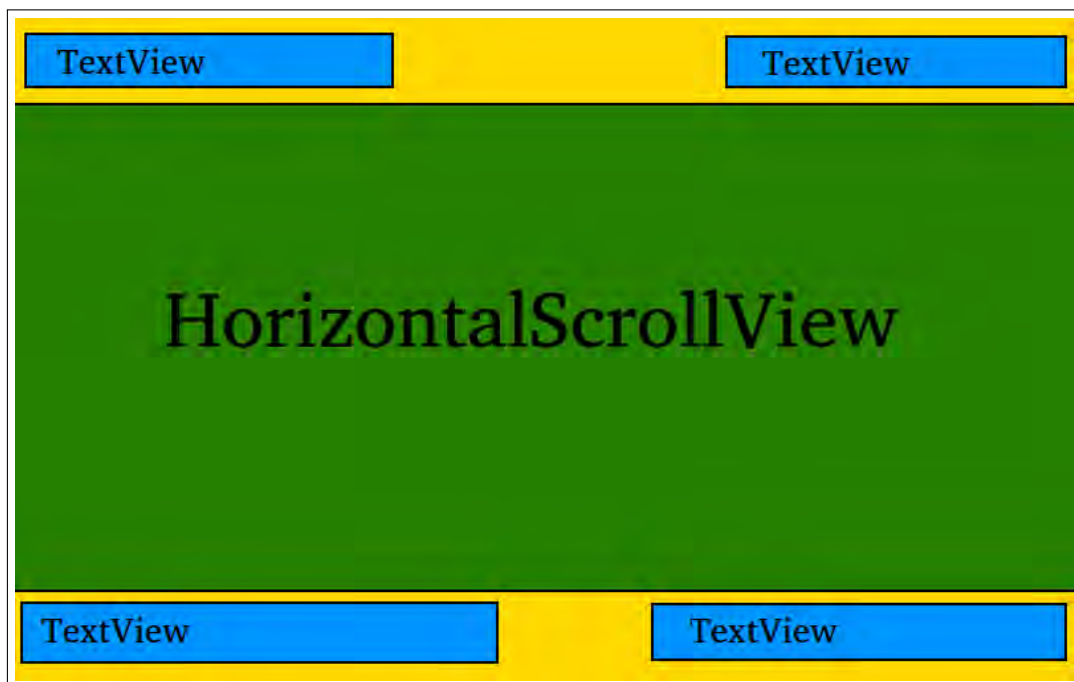


Abbildung 3.2: Überblick über das Layout einer Übung

Das dynamische Layout passt sich je nach BenutzerAuswahl an, so werden beispielweise Buttons zum Starten der Übung nach dem Start entfernt. Dafür werden nach der VideoView zwei neue Buttons zum Pausieren und Abschließen der Übung eingefügt.

3.5 Ablaufdiagramm

Das Erstellen eines Ablaufdiagramms ist ein sehr gutes Hilfsmittel um sich des Ablauf des Programms bewusst zu werden und um sich erste Gedanken über die graphische Oberfläche machen zu können. In dieser Phase entstand das Ablaufdiagramm, welches in Abbildung 3.4 dargestellt ist.

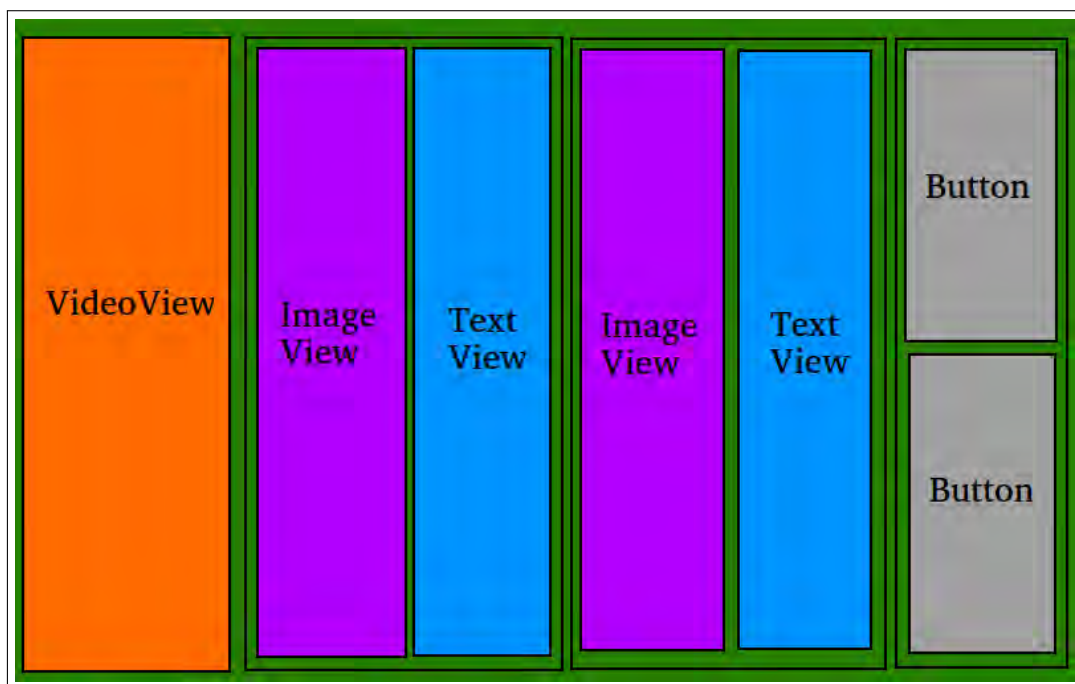


Abbildung 3.3: Überblick über das Layout eines dynamisch erzeugten Elements in der `HorizontalScrollView`

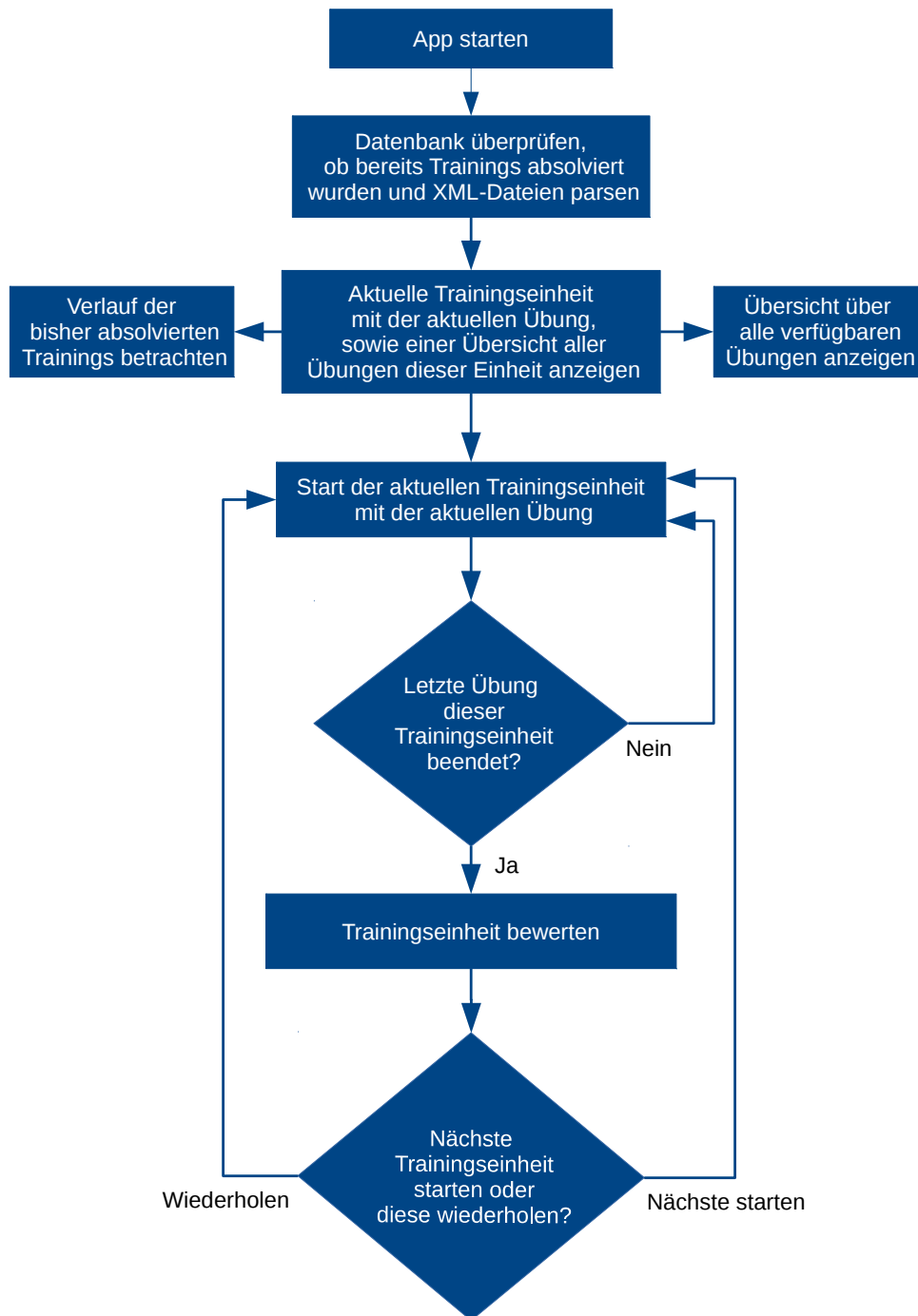


Abbildung 3.4: Ablaufdiagramm

Kapitel 4

Implementierung der Applikation

4.1 Inhalte

In diesem Kapitel wird auf die Implementierung der Applikation eingegangen. Dabei wird beschrieben, wie bestimmte Aufgabenstellungen programmatisch gelöst wurden und der generelle Aufbau der Software aussieht.

4.2 Quellcode

Der Quellcode wird anhand der verwendeten Packages erklärt.

4.2.1 `de.tum.ei.vmi.passage.rollatortraining`

In diesem Package sind alle verwendeten Activities und weitere Packages enthalten. Folgende Activities finden Anwendung:

- **SplashActivity.java:** Parst die beiden [XML](#)-Dateien, speichert die ausgelesenen Informationen als Objekte, prüft ob bereits eine Datenbank angelegt wurde und wechselt dann zur `MainActivity.java`. Beim ersten Start der Applikationen werden außerdem die Bilder und Videos aus dem `assets`-Ordner auf die SD-Karte kopiert, damit diese von der App verwendet werden können. Ein Sicherheitshinweis zu den Übungen wird zusätzlich beim ersten Start der App angezeigt.
- **MainActivity.java:** Zeigt den aktuellen Trainingsverlauf, wie in Kapitel [3.3](#) geplant wurde. Es kann entweder die aktuelle Trainingseinheit durch die Klasse `ExerciseActivity.java` gestartet werden oder mit dem `NavigationDrawer`-Menü eine andere Activity gestartet werden. Zur Auswahl stehen die Klassen `AllExercisesActivity.java`, `SettingsActivity.java` und `TrainingHistoryActivity.java`. Beim ersten App-Start

wird außerdem der Service gestartet, der die Benachrichtigungen versendet. Standardmäßig erfolgt jeden dritten Tag eine Benachrichtigung.

- **ExercisesActivity.java:** Diese Klasse zeigt die aktuelle Übung mit dem dazugehörigen Video, Bildern und Erklärungen an, wie in Abschnitt 3.3 diskutiert wurde. Die dazu nötigen Informationen werden aus den gespeicherten Objekten gelesen. Nach dem Ende einer Übung wird erneut die Klasse `ExerciseActivity.java` aufgerufen, falls die letzte Übung der aktuellen Trainingseinheit noch nicht erreicht wurde. Sobald die letzte Übung der Trainingseinheit beendet wurde wird zur Klasse `FeedbackActivity.java` gewechselt. Es wird bei jedem Aufruf die Dauer der absolvierten Übung registriert und an einen String angehängt, so dass die Dauer jeder einzelnen Übung in der Datenbank gespeichert wird.
- **FeedbackActivity.java:** In dieser Activity kann der Benutzer anhand von fünf ImageButtons eine Bewertung der Trainingseinheit abgeben. Mit zwei weiteren Buttons kann entweder das gerade beendete Training wiederholt werden, oder die nächste Einheit gestartet werden. Wenn das nächste Level gestartet werden soll, wird zur Klasse `MainActivity.java` gewechselt. Beim Wiederholen der Trainingseinheit startet die Klasse `ExerciseActivity.java` mit der ersten Übung der gleichen Einheit.
- **TrainingHistoryActivity.java:** Diese Klasse dient zur Auflistung aller gespeicherten Trainings aus der Datenbank mithilfe einer optisch angepassten ListView. Angezeigt wird das Startdatum der Trainingseinheit beziehungsweise der einzelnen Übung. Ein Klick auf eines der Elemente in der Liste öffnet die Klasse `SavedExerciseActivity.java`.
- **SavedExerciseActivity.java:** Eine Auflistung der Informationen aus der Datenbank wird in der Klasse `SavedExerciseActivity.java` dargestellt. Der Inhalt entspricht den Einträgen der Tabelle 3.2.
- **Settings.java:** In dieser Activity werden Einstellungen der Applikationen vorgenommen. So kann dort das Intervall der Benachrichtigungen eingestellt werden, die Datenbank exportiert und gelöscht werden. Außerdem können Informationen über die Entwickler und Ansprechpartner angezeigt werden.
- **AllExercisesActivity.java:** Hier werden alle Übungen aus der XML-Datei mit den Übungen ausgelesen und in einer ListView dargestellt. Ein Klick auf einen Eintrag der Liste startet die Klasse `SingleExerciseActivity.java`.
- **SingleExerciseActivity.java:** Diese Activity entspricht mehr oder weniger der Klasse `ExerciseActivity.java`, es wird aber nur jeweils eine Übung angezeigt. Das Beenden oder Abbrechen der Übung führt zurück zur Klasse `AllExercisesActivity.java`.

4.2.2 de.tum.ei.vmi.passage.rollatortraining.customviews

Dieses Package beinhaltet die Klassen CustomHorizontalScrollView.java und CustomListAdapter.java, die angepasste Versionen der Layout-Elementen HorizontalScrollView und ListView enthalten.

4.2.3 de.tum.ei.vmi.passage.rollatortraining.navigationdrawer

Der Inhalt dieses Packages wird zur Initialisierung und Platzierung der Elemente im Navigation-Drawer verwendet.

4.2.4 de.tum.ei.vmi.passage.rollatortraining.service

Die enthaltene Klasse NotifyService.java wird zur Konfiguration der Notification verwendet.

4.2.5 de.tum.ei.vmi.passage.rollatortraining.sqlite

Die Klasse DatabaseHandler.java kümmert sich um das Erstellen und Aktualisieren der SQLite-Datenbank. Außerdem beinhaltet diese die Funktionen zum Erstellen, Löschen oder Auslesen von Trainingseinheiten. Die Klasse TrainingActivity.java beinhaltet Getter- und Setter-Methoden und dient als Objekt einer Trainingseinheit.

4.2.6 de.tum.ei.vmi.passage.rollatortraining.utils

ActivityConstants.java beinhaltet einige statische Variablen. Diese werden zur Identifizierung der Activity genutzt, die mittels eines Intents eine andere Activity gestartet hat. Die Klasse Stopwatch.java wird zur Messung der verstrichenen Zeit nach dem Start einer Übung genutzt.

4.2.7 de.tum.ei.vmi.passage.rollatortraining.xmlparser

Die beiden Klassen XmlExerciseParser.java und XmlLevelParser.java werden entsprechend zum Parsen der exercises.xml- und levels.xml-Datei aus dem assets-Ordner verwendet. Die Klassen Exercise.java, Level.java und Step.java sind Objekte mit Getter- und Setter-Methoden um die geparsten Informationen speichern zu können. Die Klassen ExerciseFeed.java und LevelFeed.java bestehen jeweils aus einem Array der ausgelesenen Objekte aus den beiden XML-Dateien.

Abschließend wurde die mobile Anwendung noch umfangreich getestet [15].

Kapitel 5

Beschreibung der Trainingsanwendung

Nach dem Klicken des Icons im Android-Betriebssystem erscheint zuerst ein Splashscreen, der solange angezeigt wird, bis die Datenbank abgerufen und die XML-Dateien fertig geparkt wurden. Dieser Bildschirm ist in [Abbildung 5.1](#) zu sehen.



Abbildung 5.1: Splashscreen RollatorTraining

Danach erscheint die Anzeige, die in [Abbildung 5.2](#) dargestellt ist. Von hier aus gibt es zwei Möglichkeiten, wie man innerhalb der Applikationen weitergehen kann. Entweder man klickt die grüne Schaltfläche „Training starten“ und startet die nächste Übung der aktuellen Trainingseinheit oder man wählt mittels des NavigationDrawers, siehe [Abbildung 5.3](#), einen anderen Menüpunkt (Trainingsverlauf, Alle Übungen oder Einstellungen) aus.

[Abbildung 5.4](#) zeigt den Bildschirm nach dem Start einer Übung durch einen Klick auf die Schaltfläche „Training starten“. Durch Wischgesten nach links und rechts können die verschiedenen Schritte der Übung betrachtet werden und mithilfe der Buttons am Anfang und Ende die Übung



Abbildung 5.2: Startbildschirm RollatorTraining

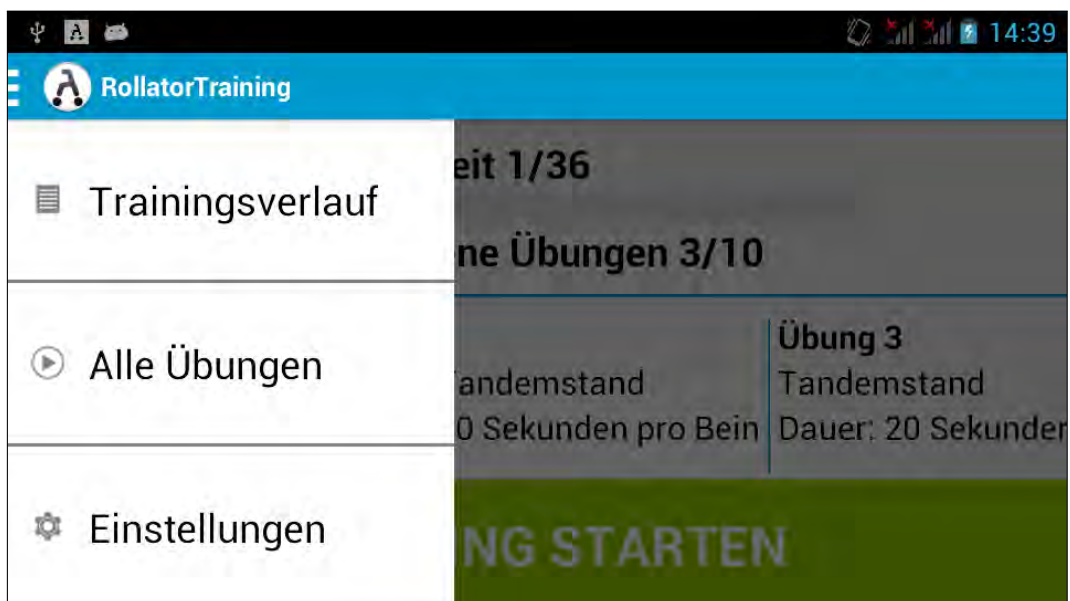


Abbildung 5.3: NavigationDrawer RollatorTraining

gestartet werden. Dadurch startet die Wiedergabe des Videos und zwei neue Buttons zum Abschließen und Pausieren der Übung werden eingeblendet, siehe Abbildung 5.5.

Nachdem alle Übungen der Trainingseinheit abgeschlossen wurden, wechselt die Ansicht zum Bildschirm, der in Abbildung 5.6 zu sehen ist.

Wählt man im NavigationDrawer-Menü den Punkt „Trainingsverlauf“ aus, wechselt die Ansicht zur



Abbildung 5.4: Übungsmodus RollatorTraining

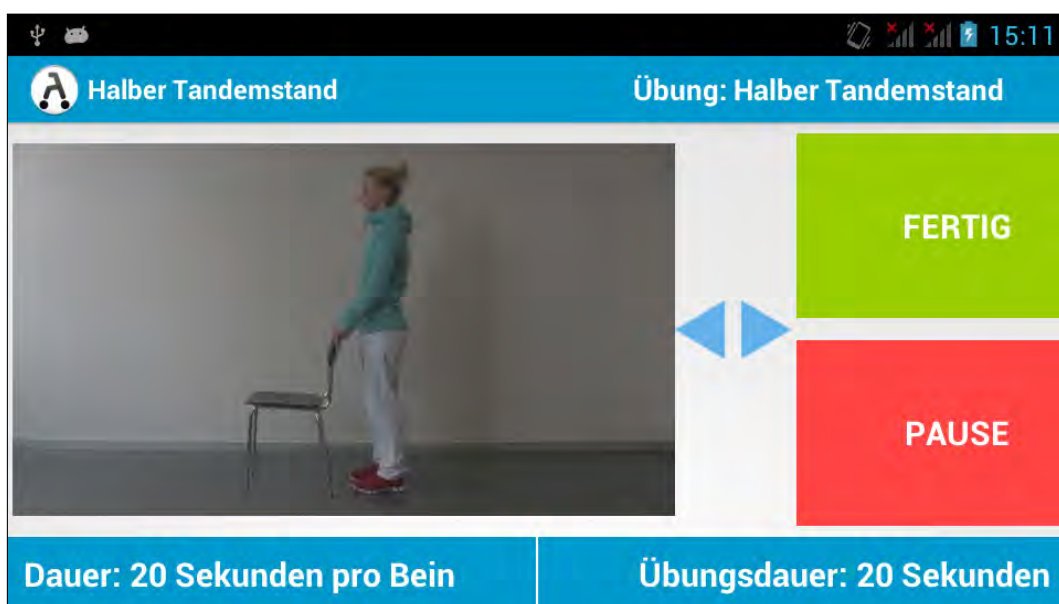


Abbildung 5.5: Übungsmodus RollatorTraining

Übersicht über alle gespeicherten Trainingseinheiten und einzelnen Übungen, die in der Datenbank angelegt wurden. Abbildung 5.7 stellt diese Ansicht dar.

Ein Klick auf einen Eintrag im Trainingsverlauf führt zu einer detaillierten Ansicht, wie in Abbildung 5.8 dargestellt.

Abbildung 5.9 zeigt den Einstellungsbildschirm, in dem verschiedene Konfigurationen vorgenom-

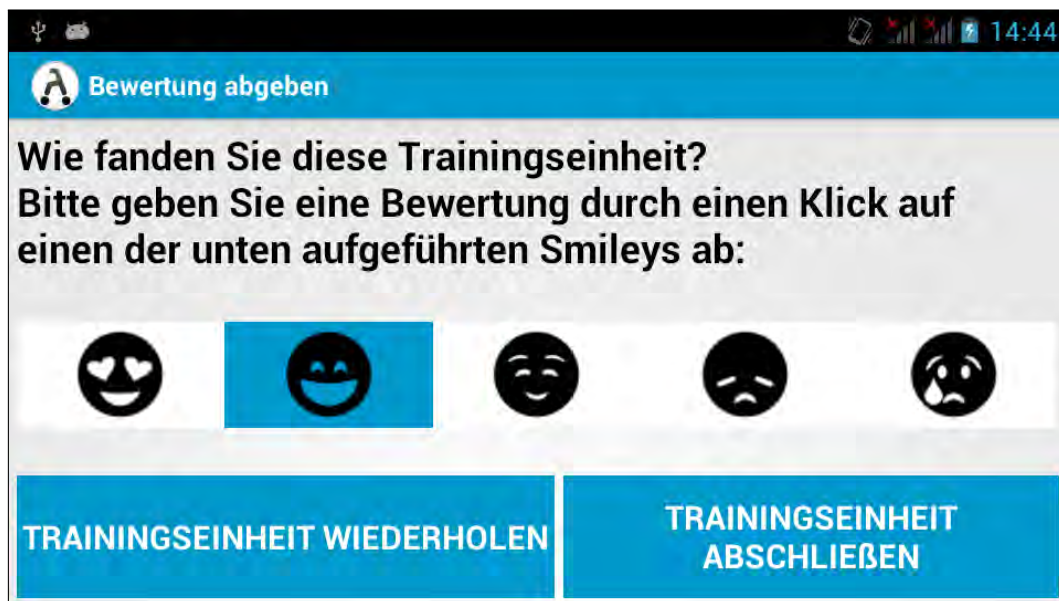


Abbildung 5.6: Feedback RollatorTraining



Abbildung 5.7: Trainingsverlauf RollatorTraining

men werden können. In [Abbildung 5.10](#) ist die Liste aller Übungen zu sehen, ein Klick darauf führt zum Übungsmodus, der bereits in [Abbildung 5.4](#) gezeigt wurde.



Abbildung 5.8: Gespeicherte Trainingseinheiten RollatorTraining

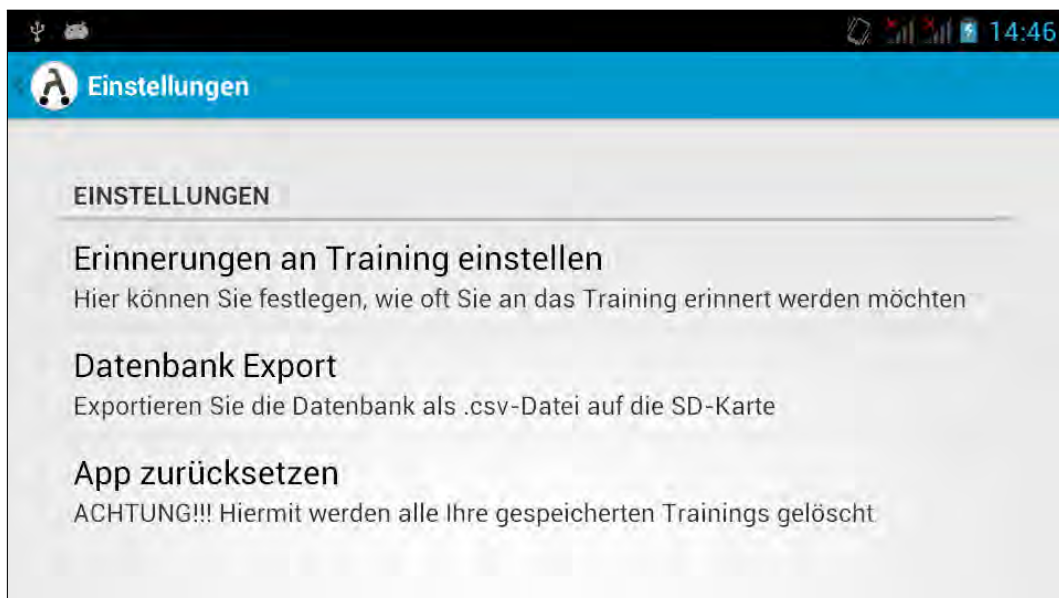


Abbildung 5.9: Einstellungen RollatorTraining



Abbildung 5.10: Alle Übungen RollatorTraining

Kapitel 6

Zusammenfassung

Die größte Herausforderung der Arbeit war es, die nötigen Inhalte in ausreichender Größe auf dem Bildschirm zu platzieren. Da Applikationen für ältere Menschen eine große Schriftart und große Schaltflächen benötigen, ist der Platz eines 4 Zoll Smartphones im Querformat sehr gering. Durch die Wahl der horizontal scrollbaren Ansicht, konnte mehr Platz geschaffen werden, der durch Wischgesten zugänglich ist. Die finale Anpassung des Layouts ist erst dann möglich, wenn alle Medien (Bilder und Videos) zu den Übungen vorhanden sind. Aktuell wurde das Layout basierend auf ersten Beispiel-Bildern und -Videos entworfen und ist daher noch nicht ganz ausgereift. Auch die Lesbarkeit aller Schriften und Erreichbarkeit aller Schaltflächen sowie Funktionen müssten in einem ersten Praxistest mit Senioren erprobt werden. Um die Akzeptanz und Verwendbarkeit der Anwendung sinnvoll zu evaluieren, sollte eine Langzeitstudie durchgeführt werden [16].

Folgende Funktionen könnten nachträglich noch implementiert werden:

- Einstellungen zum Ändern der verwendeten Farben in der App, um beispielsweise ein dunkleres, kontrastreicheres Design zu ermöglichen.
- Entwicklung einer Tablet-Version der App um auf einem größeren Display die Informationen besser darstellen zu können.
- Anbindung der Applikation an einen Server, um die Trainings dort speichern und auswerten zu können.
- Einbindung eines Belohnungssystem, um die Motivation der Trainierenden zu steigern [17].
- Verwendung externer Hardware, beispielsweise eines Pulsmessers, der via Bluetooth an die App gekoppelt werden kann und die Vitaldaten aufzeichnet.

Anhang A

Informationen zur Auswertung der SQLite Datenbank

Tabelle [A.1](#) zeigt eine exportierte Datenbank im [CSV](#)-Format. Im Folgenden werden die eingetragenen Werte und die dazugehörigen Spalten erklärt, um eine Auswertung der Daten zu ermöglichen.

- **id:** Die **ID** ist eine fortlaufende Nummer, die von 1 beginnend die Einträge in der Datenbank durchnummeriert.
- **training_type:** Es gibt zwei Arten von Trainingstypen, entweder die Durchführung von Trainingseinheiten (Training) oder das Absolvieren einzelner Übungen (SingleExercises).
- **start_time_date:** Dieser Zeitstempel gibt an, zu welcher Zeit und welchem Datum das Training gestartet wurde.
- **current_level:** Dieser Eintrag gibt die aktuelle Trainingseinheit an.
- **current_exercise:** Diese Zahl gibt die **ID** der aktuellen Übung an.
- **exercise_name:** Der Name der aktuellen Übung in ausgeschriebener Form.
- **feedback_mood:** Der gespeicherte Text kann sechs verschiedene Einträge beinhalten: „happy“, „good“, „normal“, „not good“, „bad“ und „-“. Die ersten fünf Strings beschreiben die Bewertung, die in der FeedbackActivity.java anhand der Smileys abgegeben wurde. Der Text „-“ wird abgespeichert, falls eine einzelne Übung absolviert wurde, oder eine Trainingseinheit unterbrochen wurde und somit keine Bewertung zu diesem Zeitpunkt verfügbar ist.
- **duration:** Dieser Eintrag beinhaltet die Übungsdauer in Sekunden. Falls eine Trainingseinheit durchgeführt wurde, wird beispielsweise der Text „Start, 4, , 4 , , 3, ,“ abgespeichert. Das bedeutet, dass drei Übungen gemacht wurden. Die erste Übung dauerte vier Sekunden, die Zweite ebenfalls vier Sekunden und die Dritte dauerte drei Sekunden. Je nach Anzahl der Übungen variiert dieser String in der Länge.

Tabelle A.1: Exportierte SQLite Datenbank im CSV-Format

id	training_type	start_time_date	current_level	current_exercise	exercise_name
1	Training	07.05.2014 12:04:23	1	2	Halber Tandemstand
2	SingleExercise	07.05.2014 12:04:36	1	42	Einbeinstand
3	Training	Übung vor Start abgebrochen	1	2	Halber Tandemstand
4	SingleExercise	08.05.2014 13:59:25	1	2	Geschlossene Füße
5	SingleExercise	08.05.2014 14:02:18	1	4	Tandemstand

id	feedback_mood	duration	user_canceled	started_from_notification	level_repeated
1	-	Start, 4, , 4 , , 3, ,	1	0	1
2	happy	2	0	0	0
3	-	Übung vor Start abgebrochen	1	0	1
4	-	11	1	1	0
5	-	400	1	0	0

- **user_canceled:** Mithilfe von 0 (false) oder 1 (true) wird angezeigt, ob die Übung oder Trainingseinheit vorzeitig vom Benutzer abgebrochen wurde.
- **started_from_notification:** Mithilfe von 0 (false) und 1 (true) wird angezeigt, ob die App durch einen Klick auf die Android-Benachrichtigung gestartet wurde.
- **level_repeated:** Mithilfe einer 0 (false) und 1 (true) wird angezeigt, ob eine Trainingseinheit vom Benutzer durch einen Klick auf die entsprechende Schaltfläche in der Klasse FeedbackActivity.java wiederholt wurde.

Abbildungsverzeichnis

2.1	Bevorzugte Größe der Schaltflächen, Quelle: [10]	5
3.1	Überblick über das Layout vom Startbildschirm	12
3.2	Überblick über das Layout einer Übung	13
3.3	Überblick über das Layout eines dynamisch erzeugten Elements in der Horizontal-ScrollView	14
3.4	Ablaufdiagramm	15
5.1	Splashscreen RollatorTraining	19
5.2	Startbildschirm RollatorTraining	20
5.3	NavigationDrawer RollatorTraining	20
5.4	Übungsmodus RollatorTraining	21
5.5	Übungsmodus RollatorTraining	21
5.6	Feedback RollatorTraining	22
5.7	Trainingsverlauf RollatorTraining	22
5.8	Gespeicherte Trainingseinheiten RollatorTraining	23
5.9	Einstellungen RollatorTraining	23
5.10	Alle Übungen RollatorTraining	24

Tabellenverzeichnis

2.1	Einschränkungen im Alter	3
3.1	Struktur der Trainingseinheiten	11
3.2	SQLite Datenbank Struktur	11
A.1	Exportierte SQLite Datenbank im CSV-Format	27

Abkürzungsverzeichnis

XML	Extensible Markup Language
CSV	Comma-separated values
ID	Identifikator

Literaturverzeichnis

- [1] J. A. Blaya, H. S. F. Fraser, und B. Holt, "E-health technologies show promise in developing countries.," *Health Aff (Millwood)*, Band 29, Nr. 2, Seiten 244–51, 2010.
- [2] A. Kalache und A. Gatti, "Active ageing: a policy framework," *World Health Organization - Noncommunicable Disease Prevention and Health Promotion - Ageing and Life Course, Genf*, 2002.
- [3] B. Geilhof, J. Güttler, M. Heuberger, S. Diewald, und D. Kurz, "Weiterentwicklung existierender Assistenz- und Mobilitätshilfen für Senioren – Nutzen, Akzeptanz und Potenziale," in *uDay XII - Assistenztechnik für betreutes Wohnen* (G. Kempster und W. Ritter, Hrsg.), Pabst Science Publisher, Mai 2014.
- [4] T. Bock, S. Klein, S. Diewald, B. Geilhof, K. Wessig, und M. Kranz, "Better Living by Technical Assistance and Mobility in an Aging Society," in *Proceedings of 4th European Conference on Technically Assisted Rehabilitation (TAR 2013)*, Mar. 2013.
- [5] M. Bähr, S. Klein, S. Diewald, C. Haag, G. Hofstetter, M. Khoury, D. Kurz, A. Winkler, A. König, N. Holzer, M. Siegrist, A. Pressler, L. Roalter, T. Linner, K. Wessig, M. Heuberger, V. Warmuth, M. Kranz, und T. Bock, "PASSAge - Personalized Mobility, Assistance and Service Systems in an Ageing Society," in *6. Deutscher AAL-Kongress, AAL '13*, (Berlin, Germany), Seiten 260–269, VDE Verlag, Jan. 2013.
- [6] R. Prinzinger, *Das Geheimnis des Alterns - Die programmierte Lebenszeit bei Mensch, Tier und Pflanze*. Campus-Verlag, 1996.
- [7] Y. Batu Salman, K. Young-Hee, und C. Hong-In, "Senior - friendly icon design for the mobile phone," in *Digital Content, Multimedia Technology and its Applications (IDC), 2010 6th International Conference on*, Seiten 103–108, Aug 2010.
- [8] H. Kullmann und E. Seidel, *Lernen und Gedächtnis im Erwachsenenalter*. Perspektive Praxis, Bertelsmann W., 2005.
- [9] S. Wirtz, E. Jakobs, und M. Ziefle, "Age-specific usability issues of software interfaces," *Proceedings of the IEA 2009 - 17th World Congress on Ergonomics*, Beijing, 2009.

-
- [10] K. A. Moor, K. Connelly, und Y. Rogers, "A Comparative Study of Elderly, Younger, and Chronically Ill Novice PDA Users," tech. rep., Indiana University, Bloomington, IN, 2004.
- [11] M. Burkhard und M. Koch, "Evaluating Touchscreen Interfaces of Tablet Computers for Elderly People," in *Mensch & Computer 2012 – Workshopband: interaktiv informiert – allgegenwärtig und allumfassend!?* (H. Reiterer und O. Deussen, Hrsg.), (München), Seiten 53–59, Oldenbourg Verlag, 2012.
- [12] H. Kim, J. Heo, J. Shim, M. Kim, S. Park, und S. Park, "Contextual Research on Elderly Users' Needs for Developing Universal Design Mobile Phone," in *Universal Access in Human Computer Interaction. Coping with Diversity* (C. Stephanidis, Hrsg.), Band 4554 von *Lecture Notes in Computer Science*, Seiten 950–959, Springer Berlin Heidelberg, 2007.
- [13] A. Möller, A. Thielsch, B. Dallmeier, L. Roalter, S. Diewald, A. Hendrich, B. E. Meyer, und M. Kranz, "MobiDics – Improving University Education With A Mobile Didactics Toolbox," in *Ninth International Conference on Pervasive Computing (Pervasive 2011), Video Proceedings*, (San Francisco, CA, USA), June 2011.
- [14] A. Möller, L. Roalter, S. Diewald, J. Scherr, M. Kranz, N. Hammerla, P. Olivier, und T. Plötz, "Gymskill: A personal trainer for physical exercises," in *International Conference on Pervasive Computing and Communications*, Seiten 213–220, März 2012.
- [15] S. Diewald, L. Roalter, A. Möller, und M. Kranz, "Towards a Holistic Approach for Mobile Application Development in Intelligent Environments," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, MUM '11, (New York, NY, USA), Seiten 73–80, ACM, Dez. 2011.
- [16] A. Möller, M. Kranz, B. Schmid, L. Roalter, und S. Diewald, "Investigating self-reporting behavior in long-term studies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, (New York, NY, USA), Seiten 2931–2940, ACM, 2013.
- [17] S. Diewald, A. Möller, L. Roalter, T. Stockinger, und M. Kranz, "Gameful design in the automotive domain: review, outlook and challenges," in *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '13, (New York, NY, USA), Seiten 262–265, ACM, 2013.