ulm university universität ulm

Fakultät für Ingenieurwissenschaften und Informatik
Institut für Datenbanken und Informationssysteme

Masterarbeit
im Studiengang Informatik

Konzeption, Implementierung und Evaluation eines Rahmenwerks zur Auslesung der Herzfrequenz durch Fitnesstracker in Android

vorgelegt von

Florian Malsam

Januar 2018

1. Gutachter	Prof. Dr. Manfred Reichert
2. Gutachter	Dr. Rüdiger Pryss
Betreuer:	Dr. Rüdiger Pryss
Matrikelnummer	900021
Arbeit vorgelegt am:	22.01.2018

Danksagungen

An dieser Stelle möchte ich mich bei meinen Themengebern und den Personen, die mich bei dieser Arbeit unterstützt haben, bedanken.

Vielen Dank an Dr. Rüdiger Pryss für die Bereitstellung des Themas und die Betreuung der Arbeit.

Ein weiterer Dank gilt Prof. Dr. Manfred Reichert für die Gutachtertätigkeit bei dieser Arbeit.

Ich danke auch Dr. Winfried Schlee für die Unterstützung bei der Evaluation.

Meinen Eltern danke ich für die Unterstützung während meines Studiums.

Außerdem danke ich Philipp für das Korrekturlesen dieser Arbeit und allen Testpersonen, die sich die Zeit genommen haben an der Evaluation teilzunehmen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im Rahmen der Forschungsarbeit des DBIS Instituts entstand in Zusammenarbeit mit der Tinnitus Research Initiative (TRI) der Universität Regensburg ein Projekt namens Track Your Tinnitus (TYT) das auf der Mobile Crowd Sensing Forschung basiert.

> „*Das Institut für Datenbanken und Informationssysteme (DBIS) der Universität Ulm forscht seit 1990 in den Bereichen der prozess- bzw. daten-orientierten Informationssysteme und aktueller Datenbank-Management-Systeme (Process and Data Science). Im Fokus der Forschung der Arbeitsgruppe von Prof. Dr. Manfred Reichert (Leiter des Instituts) steht die Flexibilität von prozessorientierten Informationssystemen entlang des Lebenszyklus (Design, Konfiguration, Betrieb und Optimierung)*“ [18].

Die Erfassung von Daten durch große Menschenmengen mittels Smartphones oder anderen mobilen Geräten wird als Mobile Crowd Sensing definiert [40]. Bei TYT sind speziell die persönlichen Daten von Tinnitus Patienten relevant. Ziel war es, ein System zu entwerfen, durch das Tinnituspatienten mit Hilfe von mobilen Fragebögen aktuelle Tinnitusparameter wie aktuelle Wahrnehmung des Tinnitus, momentane Belastung durch den Tinnitus, aktuelle Stimmungslage und so weiter in einem System persistieren können. Diese Daten werden dann für Forschungszwecke, aber auch vom Patienten, zur Einsicht in die persönliche Historie, verwendet.
2014 konzipierte und implementierte Jochen Herrmann dieses System in seiner Diplomarbeit an der Universität Ulm mit dem Titel “Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten“ [25]. Durch dieses Client-Server System wird Tinnituspatienten durch mobile Android- und iOS-Applikationen die Möglichkeit gegeben, ihre Tinnitusparameter mobil zu erfassen und unter ihrem Profil online zu speichern. Crowd Sensing profitiert von einer hohen Zahl an Parametern, da so aufschlussreichere Analysen möglich sind. Zudem ist es auch für den Pateienten von Vorteil, wenn er selbst viele Parameter historisch verfolgen kann. Ein Parameter, der noch nicht vom TYT System erfasst wird, ist die Herzfrequenz. Dafür wird in dieser Arbeit

ein Rahmenwerk zur Integration von Fitnesstrackern entworfen und implementiert, um die Herzfrequenz abzufragen.

1.1 Problemstellung

Das Track Your Tinnitus System beziehungsweise die Android Applikation erfasst neben den Antworten aus einem Fragebogen den Geräuschpegel der Umgebung als weiteren Parameter. Potentielle andere Faktoren mit Auswirkung auf den Tinnitus wie der Blutdruck, die Sauerstoffsättigung im Blut oder die Herzfrequenz werden bisher nicht berücksichtigt. Eine weitere Anforderung ist es, die Herzfrequenz von einem vom Pateienten getragenen Fitnesstracker zu beziehen um so zukünftig eventuell neue Erkenntnisse, im Sinne des Crowd Sensing, über das Auftreten des Tinnitus zu erlangen.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, eine Architektur für ein modulares Rahmenwerk zur Auslesung der Herzfrequenz von Fitnesstrackern zu erstellen und diese für Android zu implementieren. Dabei wird sich nicht auf einen Übertragungsstandard wie zum Beispiel Bluetooth festgelegt, sondern die Architektur soll flexibel für theoretisch alle Protokolle erweiterbar sein. In dieser Arbeit wird das Protokoll Bluetooth Low Energy am Beispiel vom Mio Alpha [11] und dem Mio Alpha 2 [12], sowie das Protokoll ANT+ am Beispiel vom Garmin vivosmart HR+ [6] für die Herzfrequenzmessung implementiert. Nach der Implementationsphase wird das System evaluiert. Dafür wurde in Zusammenarbeit mit dem Tinnituszentrum der Universität Regensburg eine Teststudie entwickelt. Diese Teststudie wird durch eine Prototyp Applikation an 20 Testpersonen durchgeführt, die im Anschluss Feedback durch einen Fragebogen geben sollen.

1.3 Aufbau der Arbeit

Zu Beginn werden in Kapitel 2 die Grundlagen erläutert. Dazu gehört das Track Your Tinnitus Projekt, die im Projekt verbundene Hardware und die eingesetzten Verbindungsprotokolle. In Kapitel 3 wird die Architektur erläutert sowie die Anforderungen an das Rahmenwerk und die Testapplikation definiert. Kapitel 4 handelt von der Implementierung des Rahmenwerks. Hier wird auf spezielle Implementierungsdetails eingegangen und die Erweiterungsmöglichkeit geschildert. Außerdem wird die empfohlene Benutzung erläutert. In Kapitel 5 werden die Anforderungen an das Rahmenwerk mit den realisierten Funktionen abgeglichen. In Kapitel 6 wird die Evaluation des Rahmenwerks beziehungsweise den eingesetzten Fitnesstrackern beschrieben. Es wird das Evaluationsszenario geschildert und die Ergebnisse präsentiert. Kapitel 7 fasst die Ergebnisse zusammen und gibt einen Ausblick für mögliche Weiterentwicklungen.

1.4 Zusammenfassung

Im Rahmen der Forschungsarbeit des DBIS Instituts entstand in Zusammenarbeit mit der Tinnitus Research Initiative (TRI) der Universität Regensburg ein Projekt namens Track Your Tinnitus (TYT) das auf der Mobile Crowd Sensing Forschung basiert. Für die Android Applikation dieses Projekts entstand die Anforderung, Fitnesstracker bei der Patientenbefragung miteinzubunden, um deren Herzfrequenz zu beziehen. Ziel dieser Arbeit ist die Architektur und Implementierung eines Rahmenwerks, das dies auf flexible weise ermöglicht.

Kapitel 2

Grundlagen

In diesem Kapitel wird das Track your Tinnitus Projekt vorgestellt, es wird die eingesetzte Hard- und Software beschrieben und es werden die benötigten Verbindungsprotokolle detailliert beschrieben. Es werden verwandten Arbeiten vorgestellt und die Studienlage in Bezug auf die eingesetzten Fitnesstracker wird untersucht.

2.1 Track Your Tinnitus

Track Your Tinnitus (TYT) ist ein Projekt der Tinnitus Research Initiative und dem Institut für Datenbanken und Informationssysteme der Universität Ulm. Tinnitus ist definiert als eine akkustische Wahrnehmung, obwohl keine physikalische Quelle existiert [39]. Fast jeder Mensch wird in seinem Leben einmal irgendeine Form von Tinnitus wahrnehmen, jedoch wird dieses Phänomen meist nach einigen Sekunden bis Minuten wieder verschwinden. Allerdings nehmen 10% der Bevölkerung den Tinnitus dauerhaft wahr und oft bleibt er chronisch, das ganze Leben lang. Auch im chronischen Fall kann die Wahrnehmung des Tinnitus zwischen den Tageszeiten beziehungsweise Tagen variieren, indem das störende Geräusch mehr oder weniger wahrnehmbar ist. Diese unterschiedliche Wahrnehmung hängt von verschiedenen Faktoren wie Umgebungslautstärke, Tagesaktivität, Stress, Tageszeit und vielem mehr, ab. Viele Menschen haben ein gutes Gefühl für diese Variation und können den Verlauf aus dem Gedächtnis rekonstruieren. Mit dem TYT System ist die systematische Aufzeichnung des Tinnitus über einen längeren Zeitraum realisierbar. Durch die Webseite in Kombination mit dem Smartphone ist es möglich, den Tinnitus durch zeitlich zufällig auszufüllende Fragebögen aufzuzeichnen. Somit kann man eventuell herausfinden, wie das Auftreten des Tinnitus mit der täglichen Routine oder bestimmten Alltagsaktivitäten zusammenhängt [19].

> *„Hierfür haben wir eine Webseite und eine App für iOS und Android entwickelt. Die Webseite dient dabei als Informationsquelle für neue Benutzer und bietet darüber hinaus noch weitere Funktionen. Das Abfragen der Schwan-*

> *kungen der Tinnituswahrnehmung findet nur in den Apps statt. Ein Benutzer wird dabei in unregelmäßigen Abständen daran erinnert einen kurzen Fragebogen auszufüllen. Diese Methode nennt man auch "Experience Sampling" oder "Experience Sampling Method"und geht auf die Forscher Larson und Csikszentmihalyi zurück.*" [25]

Das TYT System ist in "Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback" [35], "Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback" [34] und "Measuring the Moment-to-Moment Variability of Tinnitus: The TrackYourTinnitus Smart Phone App" [36] beschrieben. Andere Arbeiten lassen vermuten, dass es mehre Faktoren für die Wahrnehmung des Phantomgeräusches Tinnitus gibt. Dazu könnten Medikation, emotionale Aufgewühlltheit, Stress, Alkohol, Koffeinkonsum, Infektionen, Hormonlevel, Schlafqualität oder auch die Herzfrequenz, gehören [35]. Eine Anforderung des TYT Systems, beziehungsweise der Android Applikation, ist die Aufnahme der Herzfrequenz, worum es in dieser Arbeit geht. Dies geschieht über potentiell vom Nutzer getragene Fitnesstracker.

2.2 Verwandte Arbeiten

In diesem Abschnitt werden vorherige, verwandte Arbeiten die sich mit dem Tinnitus und dem TYT System außeinandersetzen kurz zusammengefasst.

Outpatient Tinnitus Clinic, Self-Help Web Platform, or Mobile Application to Recruit Tinnitus Study Samples? [32]

In dieser Arbeit von Probst et. al. wurden verschiedene Methoden verglichen um Daten von Tinnituspatienten zu beziehen. Dafür wurden Daten von 9670 Personen bezogen. 5017 davon waren auf der Selbsthilfeplattform Tinnitus Talk angemeldet, 867 nutzten eine TYT Applikation und 3786 Personen waren in Kontakt mit der ambulanten Tinnitus Klinik in Regensburg. Es wurden Alter, Geschlecht und Dauer des Tinnitus verglichen. Auffällig war dabei, dass die Nutzer der TYT Applikation jünger waren, bei Nutzern von Tinnitus Talk war der Anteil an weiblichen Personen höher und die Nutzer der neuen Technologien (Tinnitus Talk und TYT Applikation) hatten öfter und länger Tinnitus Beschwerden. Die Implikation der Forscher daraus ist, dass durch crowdsensing und crowdsourcing Plattformen der Kontakt zu Tinnituserkrankten, die nicht in Kontakt mit einer ambulanten Tinnitus Klinik sind, erleichtert wird.

Does tinnitus depend on time-of-day? An ecological momentary assessment study with the "TrackYourTinnitus" application [30]

In Dieser Arbeit wurden Daten der TYT Applikation genutzt um zu untersuchen, ob ein Zusammenhang zwischen dem Auftreten und der stärke des Tinnitus mit der Tageszeit besteht. Die Forscher kamen zu dem Ergebnis, dass der Tinnitus nachts und morgens stärker wahrgenommen wird.

Mobile Crowdsensing for the Juxtaposition of Realtime Assessments and Retrospective Reporting for Neuropsychiatric Symptoms [33]

Normalerweise werden Daten von Tinnitus Patienten rückblickend aufgenommen. Dies geschieht über Gespräche oder Fragebögen. Man weiß bisher wenig darüber, wie aussagekräftig solche Berichte über die Vergangenheit sind. Pryss et. al. beschäftigten sich mit dieser Frage und verglichen herkömmlich aufgenommene Daten mit Daten die durch eine TYT Applikation nicht rückblickend aufgenommen wurden. Es wurde aufgezeigt, dass es einen Unterschied zwischen herkömmlich aufgenommenen Daten und Daten die durch eine TYT Applikation aufgenommen wurden, gibt.

Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the "TrackYourTinnitus" application [31]

In dieser Untersuchung wurden Daten von der TYT Applikation genutzt um die Rolle des emotionalen Zustands "Stress" in Bezug auf die Wahrnehmung und Belastung durch den Tinnitus zu bestimmen. Die Untersuchung zeigte auf, dass die Tinnitusbelastung durch die Lautstärke des Tinnitus von der emotionalen Lage abhängig ist.

2.3 Fitnesstracker

Ein Fitnesstracker ist ein elektronisches Gerät das am Körper, heutzutage meistens am Handgelenk, getragen wird und dazu dient, gesundheitsrelevante Daten zu versenden und aufzuzeichnen. Zu diesen Daten gehören oft die Herzfrequenz und die Anzahl zurückgelegter Schritte. Oft wird auch Aufschluss über den Energieumsatz oder die Schlafqualität gegeben [5, 7]. Die Messung der Herzfrequenz geschieht über eine optische Pulsmessung. Bei der optischen Pulsmessung wird das Blutvolumen in den Arterien gemessen, welches sich beim Pulsschlag ändert. Auf der Unterseite der Tracker sind LED-Lampen und ein optischer Sensor angebracht. Die LED leuchtet auf die Haut und die Reflektion des Lichtstrahls ist abhängig vom Blutvolumen. Durch die Messung der Reflektion mit Hilfe des optischen Sensors können Rückschlüsse auf den Puls gezogen werden [23, S. 38]. Die meisten Hersteller bieten den Nutzern eine Smartphone Applikation an um diese Daten auszulesen. Es gibt bei einigen Fitnesstrackern allerdings auch die Möglichkeit, die Rohdaten auszulesen. Dadurch ist es möglich den Fitnesstracker mit verschiedenen Geräten, wie beispielsweise einem Laufband, zu koppeln. Dies geschieht beispielsweise über das ANT+ oder das Bluetooth Low Energy Protokoll. Laut einer repräsentativen Umfrage von Bitkom Research nutzen 31% der Personen über 14 Jahren bereits Geräte, um Gesundheits- oder Fitnessdaten zu erfassen. 18% davon sind die hier erläuterten Fitnesstracker [28].

2.4 Bluetooth

Bluetooth ist ein offener Industriestandard für die Datenübertragung über kurze Distanz über Funk. Es wird hauptsächlich genutzt um “wireless personal area networks“ (WPANs) einzurichten. Es wird mittlerweile in vielen Geräten wie Smartphones, Laptops, Fahrzeugen, Druckern, et cetera, eingesetzt. Neuerdings findet es auch in medizinischen und persönlichen Geräten verwendung, wozu Fitnesstracker zählen, die für diese Arbeit relevant sind. Bluetooth erlaubt es, ad-hoc Netzwerke zwischen verschiedenen Arten von Geräten herzustellen um (Audio-)Daten zu übertragen. Vorteile von Bluetooth sind die niedrigen Kosten der Chips und der geringe Stromverbrauch. Zudem ist die Übertragung durch das große Frequenzspektrum (2.4 GHz ISM Band, 79 Kanäle [26, S. 65]) und Frequency Hopping Spread Spectrum (FHSS), durch die sich die Übertragungsfrequenz bis zu 1600 mal pro Sekunde ändert, robust. Beim klassischen Bluetoothstandard werden sogenannte Piconets [26, S. 69] (siehe Abbildung 2.1) erstellt, um den Datenaustausch zu ermöglichen. Dieses besteht aus mindestens zwei Bluetoothgeräten die sich in physikalischer Nähe befinden und auf dem selben Kanal senden beziehungsweise empfangen. In einem Piconet gibt es stets einen Master, der die Hopping Frequenz vorgibt, an die sich alle Clients halten. Ein Beispiel wäre hierfür ein Computer als Master und eine Bluetoothmaus und eine Bluetoothtastatur als Slaves.

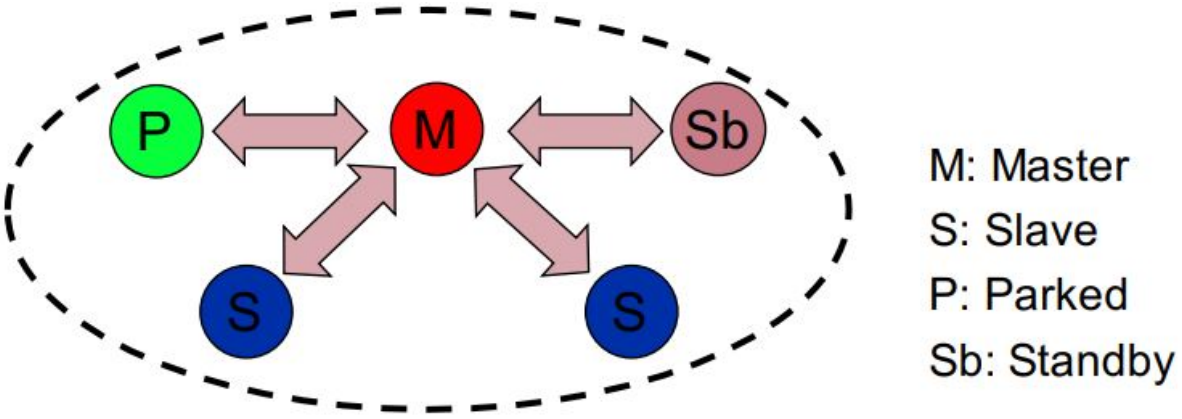


Abbildung 2.1: Piconet (Quelle: FKR Skript Prof. Kargl)

Erfunden wurde Bluetooth ursprünglich 1994 von Ericsson. Ericsson, IBM, Intel, Nokia und Toshiba gründeten die Bluetooth Special Interest Group (SIG) [3], einen non-profit Verbund zur Ausarbeitung eines Standards, der verbindliche Spezifikationen festlegt. Bluetooth ist in der IEEE 802.15 Gruppe [10] für WPANs standardisiert [17, S. 3].

Auf der Applikationsebene wird über Bluetooth Profile kommuniziert. Bluetooth Profile sind Schnittstellenspezifikationen die von der Bluetooth Special Interest Group festgelegt wurden um die Kommunikation in einer Bluetooth Umgebung zu standardisieren. So-

mit kann eine Kommunikation zwischen verschiedendsten Geräten implementiert werden. Beispielsweise ist das Headset Profil in den meisten Smartphones implementiert, um die Kommunikation mit Freisprecheinrichtungen oder Headsets zu ermöglichen.

2.4.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE) wurde in der Bluetooth 4.0 Spezifikation eingeführt und in 4.1 und 4.2 geupdated. BLE arbeitet ebenfalls auf dem 2.4 GHz ISM Band [22], allerdings nur noch auf 40 Kanälen, die dafür 2 statt 1 MHz breit sind [26, S. 112][17, S. 4]. Es wurde entwickelt um Bluetooth in Geräten zu verwenden, die per Knopfbatterie (circa 50mAh, am Beispiel der PowerStream Li-ion Coin Cell Lir2032 [13]) betrieben werden. Im Gegensatz zum klassischen Piconet können bei BLE 1:1 Verbindungen hergestellt werden, wie es bei der Kopplung von Fitnesstrackern der Fall ist [9].

Generic Attribute Profile

Eins der erwähnten Bluetooth Profile ist das Generic Attribute Profile (GATT), welches bei BLE zum Einsatz kommt. Das GATT Profil ist ein allgemein gehaltenes Profil und das einzige Profil, das mit BLE kompatibel ist. Im Gegensatz zu den anderen Profilen ist das GATT Profil für keinen bestimmten Einsatzzweck vorbestimmt. Das GATT Profil repräsentiert lediglich Schlüssel-Wert Paare.

2.5 ANT

ANT ist ebenfalls ein offener Industriestandard für die Datenübertragung über kurze Distanz über Funk, wie BLE. Es ist bei den Anwendungsfällen und den technischen Details BLE sehr ähnlich. Es basiert ebenso auf dem 2,4 GHz ISM Band und wurde auch für eine sehr energiesparende Übertragungen entwickelt [2, S. 8]. In ANT nimmt ein Gerät entweder die Position eines Masternodes oder die eines Slavenodes ein. Der Masternode ist der Initiator der Verbindung, der Verantwortliche des Kanalmanagements und der Hauptübertrager der Daten. Am Beispiel eines Fitnesstrackers ist der Tracker der Masternode und es kann mehrere Slavenodes geben, die die gesendete Herzfrequenz empfangen. Somit sind die Slaveknoten primäre Empfänger von Daten [2, S. 11]. . Es ist ebenfalls möglich, gleichzeitig Slave und Master zu sein [27]. Dadurch unterstützt ANT eine große Zahl an Netztopologien. Anders im Vergleich zu Bluetooth findet kein konstantes Frequency Hopping statt. Stattdessen wird, sofern eine Störung des Kanals erkannt wird, auf einen anderen Kanal gewechselt. Durch Time Division Multiple Access (TDMA) kann einer, der 125 verfügbaren, Kanäle für mehrere, unabhängige, Verbindungen benutzt werden [2, S. 17]. ANT wurde von Dynastream (mittlerweile ein Tochterunternehmen von Garmin)

entwickelt. Für Ihre Produkte wurde ein entsprechender Funkstandard benötigt, den es zu dieser Zeit nicht gab. Aus diesem Grund wurde ANT entwickelt und 2003 eingeführt [21].

2.5.1 ANT+

Um für ANT Interoperabilität zu gewährleisten wurde die Erweiterung ANT+ entwickelt. Hier werden Standards und Datenstrukturen festgelegt um die Kommunikation zu anderen ANT+ Produkten zu standardisieren. Dadurch wird der Datenaustausch extrem vereinfacht. Es wird hauptsächlich im Sport-, Wellnes- und Homehealth-Bereich verwendet. Dazu zählen (Körper)Temperatur Sensoren, Herzratenmesser, Geschwidigkeit- und Distanzmesser, Blutdruckmesser und so weiter [27].

2.6 Android

Android ist ein weit verbreitetes Smartphone Betriebssystem und eine Open Source Software Plattform [14]. Zu den Zielgeräten gehören Smartphones, Netbooks und Tabletcomputer. Auf dem Smartphonemarkt hatte Android im dritten Quartal 2016 laut Strategy Analytics einen weltweiten Marktanteil von 87,5% [15]. Das in dieser Arbeit vorgestellte Rahmenwerk wurde für die Android Plattform mit der Benutzung von Smartphones ab Android Version 18 entwickelt, da in dieser Version BLE eingeführt wurde.

2.7 Eingesetzte Hardware

In diesem Unterkapitel wird die eingesetzt Hardware näher beschrieben. Es folgen die Fitnesstracker, die mit dem Rahmenwerk laut Anforderungsanalyse mindestens kompatibel sein sollten. Der Garmin Vivosmart HR+ wurde dabei nicht in der Evaluation eingesetzt. Außerdem werden die getestetn beziehungsweise verwendeten Smartphones und deren Android Version aufgelistet.

2.7.1 Mio Alpha 1

Der Mio Alpha 1 Fitnesstracker der Firma Mio Global ist laut Beschreibung der erste Fitnesstracker, der die Messung optisch über das Handgelenk durchführt. Der Fitnesstracker kam im Jahr 2013 auf den Markt. Der Vorteil dieses Trackers in Bezug auf diese Arbeit ist, dass er über einen Modus verfügt, bei dem die Herzfrequenz über BLE im offiziell festgelegten Format, ohne Verschlüsselung, gesendet wird. Somit ist es möglich, mit beliebigen BLE fähigen Geräten die Daten zu interpretieren.
Im Folgenden wird ein Auszug der Studienlage wiedergegeben.

How accurate are the wrist-based heart rate monitors during walking and running activities? Are they accurate enough? [38]

Eine Studie durchgeführt von Stahl et. al. verglich unter anderem die Herzfrequenzmessung des Mio Alpha 1 mit einer Messung via Brustgurt. Die Probanten wurden angewiesen 30 Minuten auf einem Laufband zu gehen beziehungsweise zu laufen. Aufgeteilt waren die 30 Minuten in 6 Intervalle a 5 Minuten.

Methode	Dauer	Einstellung
Laufband	5 Minuten	3,2 km/h
Laufband	5 Minuten	4,8 km/h
Laufband	5 Minuten	6,4 km/h
Laufband	5 Minuten	8,0 km/h
Laufband	5 Minuten	9,6 km/h
Laufband	5 Minuten	4,8 km/h

Tabelle 2.1: Stahl et. al. Testprotokoll

Die Messung des Mio Alpha 1 korrelierte hier sehr stark (r=0,929) mit der des Brustgurtes.

Mio Heart Rate Accuracy vs EKG [24]

Ebenso untersuchte Dr. Mark Gorelick der State University San Francisco die Genauigkeit der Herzratenmessung des Mio Alpha 1 anhand eines EKG Geräts. Hierbei wurden die Probanden dazu angewiesen einen Fahrradergometer Test und im Anschluss einen Laufbandtest zu machen. Beim Fahrradergometertest handelte es sich um einen YMCA Test, bei dem die Belastung während des Tests an die Leistung des Probanden angepasst wird. Hierfür wird die Belastung nach der ersten Belastungsphase individuell an die Testperson angepasst, je nachdem wie stark die Herzfrequenz schon bei der ersten Stufe reagiert hat. Abbildung 2.2 veranschaulicht dieses Vorgehen. Der Laufbandtest war aufgeteilt in 3 Intervalle a 4 Minuten.

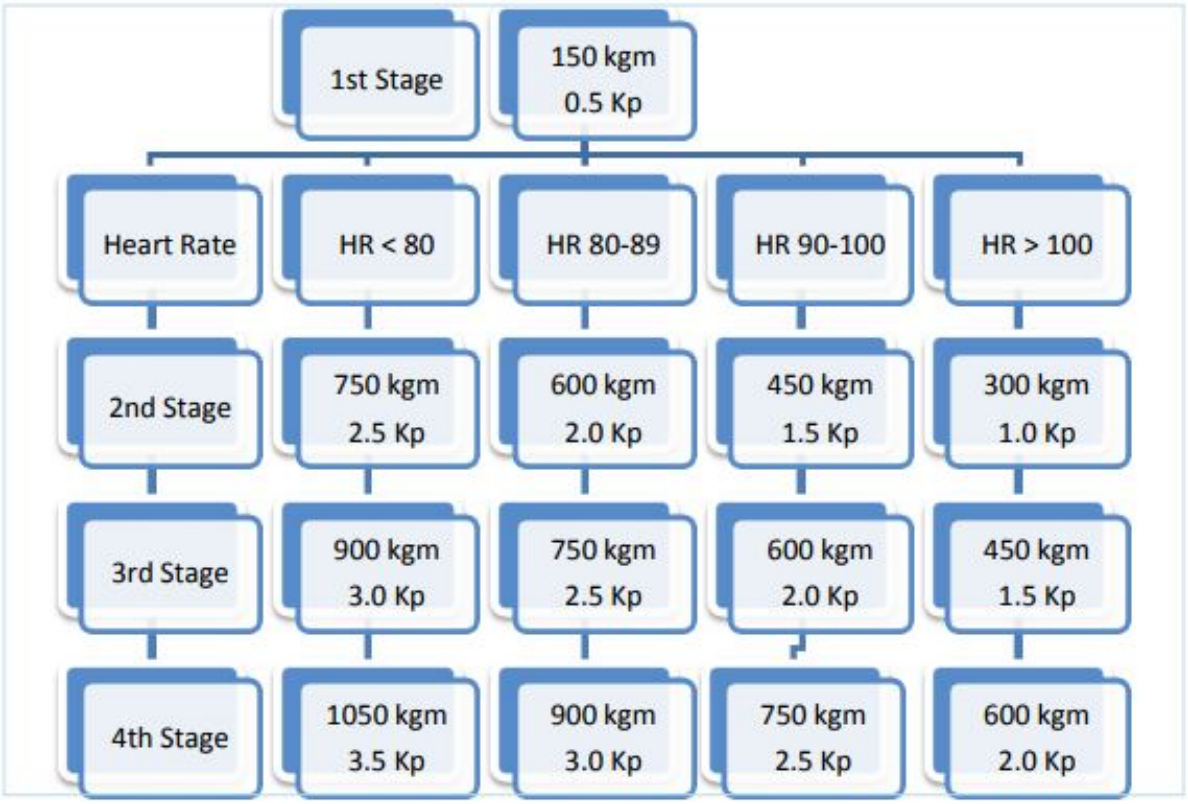


Abbildung 2.2: YMCA Test (Quelle: www.laxymca.org) kgm = Kilogram x Meter, Leistungsangabe

Der Mio Alpha 1 zeigte eine sehr starke Korrelation von r=0,99 im Vergleich mit dem EKG Gerät bei einer Differenz von 0,33∓1,19 Schlägen / Minute im Mittel. Während des Tests hatte der Mio Alpha 1 zu 93% eine kleinere Abweichung als 5 Schläge / Minute zum EKG.

Methode	Dauer	Einstellung
Laufband	4 Minuten	3,2 km/h
Laufband	4 Minuten	6,4 km/h
Laufband	4 Minuten	9,6 km/h

Tabelle 2.2: Gorelick Testprotokoll

Auch beim Laufbandtest erzielte der Mio Alpha 1 bei 6,4 und 9,6 km/h eine sehr stake Korrelation mit dem EKG Gerät von r=0,99 bei einer Differenz von 0,26∓1,13 Schlägen / Minute im Mittel. Während des Tests hatte der Mio Alpha 1 zu 97% eine kleinere Abweichung als 5 Schläge / Minute zum EKG. Zusätzlich wurde noch ein Maximaltest an einer Testperson durchgeführt. Die Testperson rannte 1 Minute lang ihre Höchstgeschwindigkeit von 23 km/h. Bei diesem Versuch lag die mittlere Differenz zwischen dem Mio Alpha 1 und dem EKG Gerät bei ∓1 Schlag / Minute.

Beurteilung

Anzumerken ist, dass Dr. Mark Gorelick die Stelle “Director of Product Science and Innovation“ bei Mio Global besetzt.

Evaluation of wearable consumer heart rate monitors based on photopletysmography [29]

Parak und Korhonen testeten das Gerät an mit einem Übungsprotokoll aus Sitzen, Liegen, Gehen, Laufen, Fahrradfahren und gängigen Alltagsbewegungen. Hierbei erreichte der Fitnesstracker nur zu 77,83% eine Genauigkeit von weniger als 5% Abweichung zum Referenzgerät. Die Forscher kamen zu dem Ergebnis, dass die Messgenauigkeit womöglich von mehreren Faktoren abhängen kann, wie ausgeführte Aktivität oder auch genaue Platzierung des Geräts.

Comparison of Non-Invasive Individual Monitoring of the Training and Health of Athletes with Commercially Available Wearable Technologies [20]

Düking et. al. analysierten die Sinnhaftigkeit von Fitnesstrackern für Athleten zur Trainingskontrolle beziehungsweise -steuerung. Dabei erwähnten sie im Zusammenhang mit dem Mio Global 2, dass der Mio Global 1 anfällig für Bewegungsartefakte sei.

2.7.2 Mio Alpha 2

Der Nachfolger des Mio Alpha 1 kam 2015 auf den Markt und erweitert diesen hauptsächlich um Funktionen wie Workoutmanagement, Hintergrundbeleuchtung, Stoppuhr und um einen Beschleunigungsmesser. Der Mechanismus zum Messen der Herzfrequenz bleibt wie bisher.

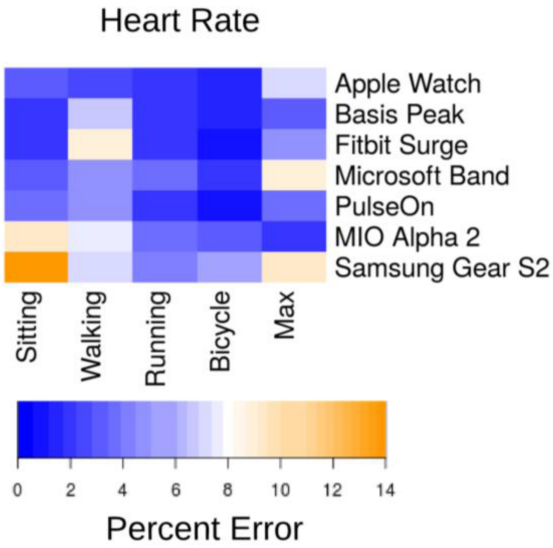
Accuracy in Wrist-Worn, Sensor-Based Measurements of Heart Rate and Energy Expenditure in a Diverse Cohort [37]

Shcherbina et. al. untersuchten die Fehlerrate der Herzfrequenz- und Energieverbrauchmessung von kommerziell erwerbbaren Geräten, die am Handgelenk getragen werden. Die Messungen wurden im Sitzen, Gehen, Laufen und auf dem Fahrradergometer gemacht.

Methode	Dauer	Einstellung
Sitzen	5 Minuten	-
Laufband, gehen	5 Minuten	4,8 km/h mit 0,5% Steigung
Laufband, gehen	5 Minuten	6,4 km/h mit 0,5% Steigung
Laufband, laufen	5 Minuten	9 km/h mit 0,5% Steigung
Laufband, laufen	5 Minuten	11 km/h mit 0,5% Steigung
Pause	3 Minuten	-
Fahrradergometer	5 Minuten	88 Watt
Fahrradergometer	5 Minuten	160 Watt
Pause	1 Minute	-

Tabelle 2.3: Shcherbina et. al. Testprotokoll

Der Mio Alpha 2 hatte bei dem Versuch beim Laufen und Fahrradergometer eine Fehlerrate von <5%, beim Gehen 8% und beim Sitzen 10%.



(A)

Abbildung 2.3: Mio Alpha 2 Ergebnis [37]

2.7.3 Garmin Vivosmart HR+

Der Garmin Visosmart HR+ ist seit 2016 auf dem Markt und verfügt ebenfalls über Sensoren zur optischen Pulsmessung. Ebenso ist es möglich, die Herzfrequenzdaten im Rohformat über das ANT+ Protokoll zu senden, wodurch er für diese Arbeit als Testgerät dient.

2.7.4 Fazit Fitnesstracker

Die Ergebnisse der Untersuchungen der Genauigkeit von Fitnesstrackern divergieren stark. Es lässt sich vermuten, dass die Genauigkeit der Messung stark von weiteren Variablen abhängig ist. Düking et. al. kamen zu dem Entschluss, dass Fitnesstracker noch nicht genug wissenschftlich untersucht wurden, um Athleten stabile Daten bezüglich ihrer Herzfrequenz zu liefern. Als Begründung nennen sie die Bewegungsartefakte und eine zu niedrige Abtastfrequenz. Stattdessen empfehlen sie verschiedene Geräte an verschiedenen Körperteilen zu kombinieren um ein besseres Gesamtbild zu bekommen [20]. Auch Shcherbina el. al. weisen darauf hin, dass Faktoren wie dunklere Haut, größerer Handgelenkumfang und höherer BMI zu einer erhöhten Fehlerrate beitragen können [37].

2.7.5 Smartphones

Bei der Evaluation der Arbeit wurden verschiedene Android Smartphones verwendet. Das älteste Gerät war das Nexus 4 von Google, das Ende 2012 auf den Markt kam. Auf dem Gerät befand sich allerdings die neuere Android Version 6.0.1. Das Desire 626G von HTC ist seit April 2015 auf dem Markt und war im Betrieb mit der Android Version 4.3. Das aktuellste Gerät war das vom chinesischen Hersteller Cubot entwickelte X18 mit der Android Version 7.0, dicht gefolgt vom Sasmsung Galaxy S8, das im April 2017 auf dem Markt erschien und mit der Android Version 7.0 betrieben wurde.

Es folgt eine tabellarisierte Darstellung der verwendeten Geräte.

Name	**Hersteller**	**Android Version**	**Erscheinungsdatum**
Nexus 4	Google	6.0.1	11/2012
Galaxy S7	Samsung	6.0	3/2016
Galaxy S8	Samsung	7.0	4/2017
G4	LG	6.0	6/2015
Desire 626G	HTC	4.3	04/2015
One Mini	HTC	4.4.2	8/2013
X18	Cubot	7.0	8/2017
3T	OnePlus	7.1.1	11/2016

Tabelle 2.4: Liste der verwendeten Smartphones

2.8 Zusammenfassung

Track Your Tinnitus (TYT) ist ein Projekt der Tinnitus Research Initiative und dem Institut für Datenbanken und Informationssysteme der Universität Ulm. Tinnitus ist definiert als eine akkustische Wahrnehmung, obwohl keine physikalische Quelle existiert. Durch die entwickelte Software im TYT Projekt ist es möglich den Tinnitus durch zeitlich zufällig auszufüllende Fragebögen aufzuzeichnen, was Informationen für die Tinnitusforschung bereitstellen soll.
Eine weitere Anforderung des TYT ist es, die Herzfrequenz mittels Fitnesstrackern auszulesen. Fitnesstracker sind elektronische Geräte, die meistens am Handgelenk getragen werden und unter anderem die Herzfrequenz bestimmen können.
Für die Übertragung der vom Fitnesstracker generierten Daten gibt es verschiedene Protokolle. Diese Arbeit befasst sich mit dem ANT+ und dem BLE Protokoll, zwei Protokolle die auf dem 2,4 GHz ISM Band basieren und entwickelt wurden, um mit wenig Energie auszukommen.

Die Implementierung geschieht für das Smartphone Betriebssystem Android.
Die bei dieser Arbeit eingesetzte Hardware besteht aus dem Mio Alpha 1, dem Mio Alpha 2 und dem Garmin Vivosmart HR+ Fitnesstracker und 7 Smartphones mit teilweise verschiedenen Android Versionen.

Kapitel 3

Konzeption

Nachdem die Grundlagen erläutert wurden, wird jetzt das entwickelte Konzept vorgestellt. Dieses Konzept gilt als Grundlage der Implementierung und legt die spezifischen Anforderungen fest. Das Ziel aus Kapitel 1 wird präzisiert und es wird auf technische Details eingegangen. Die Anforderungesanalyse mit funktionalen und nicht funktionalen Anforderungen wird in Abschnitt 3.2 durchgeführt. Im Abschnitt 3.3 werden die grundlegenden Komponenten und deren Zusammenspiel, sowie externe Schnittstellen, definiert.

3.1 Ziel

Das Ziel dieser Software ist es, ein Rahmenwerk darzustellen durch das die Kommunikation der Herzfrequenz mit gängigen Fitnesstrackern vereinfacht wird. Der Android-Entwickler sollte sich nicht mit spezifischen Implementierungen der einzelnen Verbindungsprotokolle auseinander setzen müssen. Der Entwickler soll gegen eine Schnittstelle programmieren können, damit nur marginale Änderungen nötig sind, falls angesprochene Geräte variieren. Es soll ohne großen Aufwand möglich sein, das Rahmenwerk durch neue Protokolle zu erweitern. Dies geschieht durch einen modularen, schnittstellenbasierten Aufbau des Rahmenwerks.
Zudem sollte das Rahmenwerk ab Android 4.3 lauffähig sein.

3.2 Anforderungsanalyse

Bei der Anforderungsermittlung haben sich die folgenden Anforderungen an das Rahmenwerk herausgestellt. Die Auflistungen der Anforderungen sind aufgeteilt in funktionale und nicht funktionale Anforderungen.

3.2.1 Nicht funktionale Anforderungen

Nicht funktionale Anforderungen beschreiben Anforderungen an die Nutzbarkeit des Systems. Sie sind auch bekannt als Bedingungs- oder Qualitätsanforderungen und beschreiben wie die Software arbeiten soll [16].

Nummer	Titel	Beschreibung
1	Herzfrequenzdaten	Es soll dem Benutzer des Rahmenwerks möglich sein, ohne Kenntnisse der einzelnen Verbindungsprotokolle oder Standards, standardisierte Herzfrequenzdaten von verschiedenen Fitnesstrackern auszulesen.
2	Modulare Architektur	Das Rahmenwerk soll loose gekoppelt sein, damit Änderungen am Framework keine oder minimale Änderungen am restlichen Code mit sich ziehen. Außerdem soll die Erweiterbarkeit von Fitnesstrackern beziehungsweise Verbindungsprotokollen berücksichtigt werden.
3	Ab Android 4.3	Das Rahmenwerk soll ab Android 4.3 (Einführung von BLE) funktionsfähig sein.
4	Ab Android 5.0	Das Rahmenwerk soll die aktualisierte BLE Implementierung ab Android 5.0 berücksichtigen.

Tabelle 3.1: Nicht funktionale Anforderungen des Rahmenwerks

3.2.2 Funktionale Anforderungen

Funktionale Anforderungen beschreiben die Aufgaben, die die Software lösen kann und das Verhalten der Software [16].

Nummer	Titel	Beschreibung
1	Typ	Es soll möglich sein, den Typ des zu benutzenden Fitnesstrackers anzugeben.
2	Scan	Es soll eine Scan Methode existieren, die die Umgebung nach Fitnesstrackern des entsprechenden Typs scannt.
3	Scan Callback	Es soll eine Callback-Methode existieren, die bei einem gefundenen Gerät aufgerufen wird.
4	Connect	Es soll möglich sein, sich mit einem zuvor gefundenen Gerät zu verbinden.
5	Connect to Adress	Es soll möglich sein, sich anhand einer Adresse zu einem Gerät zu verbinden, falls dieses verfügbar ist, ohne davor einen Scan machen zu müssen.
6	Connection State Callback	Es soll eine Callback-Methode existieren, die bei einer Änderung des Verbindungsstatus aufgerufen wird.
7	Start Measurement	Es soll möglich sein, eine Herzfrequenzmessung auf unbestimmte Zeit zu starten.
8	Start Measurement 2	Es soll möglich sein, eine Herzfrequenzmessung für eine bestimmte Zeit zu starten.
9	New Value Callback	Es soll eine Callback-Methode existieren die, bei einem neu eingetroffenen Herzfrequenzwert, aufgerufen wird.
10	Second Callback	Es soll eine Callback-Methode existieren die nach jeder vergangenen Sekunde im Fall einer Messung auf Zeit aufgerufen wird.
11	Finished Callback	Bei einer Messung auf Zeit, wird, falls sie existiert, eine Callback-Methode aufgerufen, wenn die Messung beendet ist.
12	Stop Measurement	Es soll möglich sein, eine Herzfrequenzmessung manuell zu stoppen.

Tabelle 3.2: Funktionale Anforderungen des Rahmenwerks

3.3 Architektur

In diesem Kapitel werden die vorhandenen Komponenten und deren Beziehungen zueinander beschrieben. Eine wichtige Anforderung an die Architektur ist die Flexibilität und die Erweiterbarkeit. Das bedeutet im Einzelnen:

- Die Möglichkeit gegen eine Schnittstelle zu programmieren
- Die Möglichkeit das Rahmenwerk um weitere Sensoren beziehungsweise Übertragungsprotokolle zu erweitern

Für diese Architekturanforderung ist das Strategiemuster sehr gut geeignet. Durch das Strategiemuster ist es möglich, unterschiedliche Implementierungen zur Laufzeit zu wählen. Es werden Familien von “Algorithmen“ definiert, jeder wird für sich abgekapselt und durch eine Schnittstelle austauschbar gemacht.

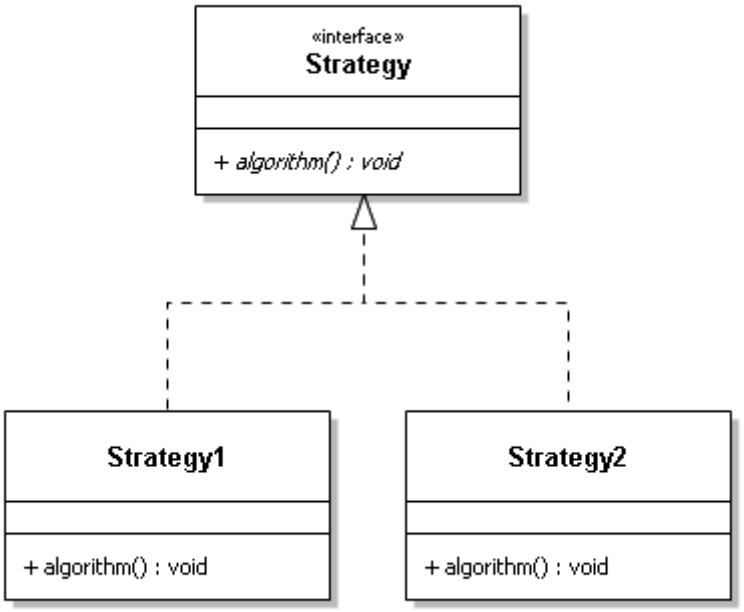


Abbildung 3.1: Strategiemuster

Durch das Strategiemuster ist die Anwendung flexibel und nie an die jeweiligen Implementierungen gebunden. So lassen sich verschiedene Sensoren, die verschiedene Protokolle benutzen, austauschbar einsetzen.

Statt einem Interface als Strategiegeber wird eine abstrakte Klasse verwendet, um bestimmte Implementierungen vorzugeben. Durch das festgelegte Datenmodell können so schon Daten, unabhängig von Sensor oder Protokoll, verarbeitet werden ohne dass eine

weitere Implementierung notwendig ist. Im Folgenden ist der Aufbau der Manager Schnittstelle zu sehen. Die Manager-Klasse kapselt die Interaktion mit den Sensoren.

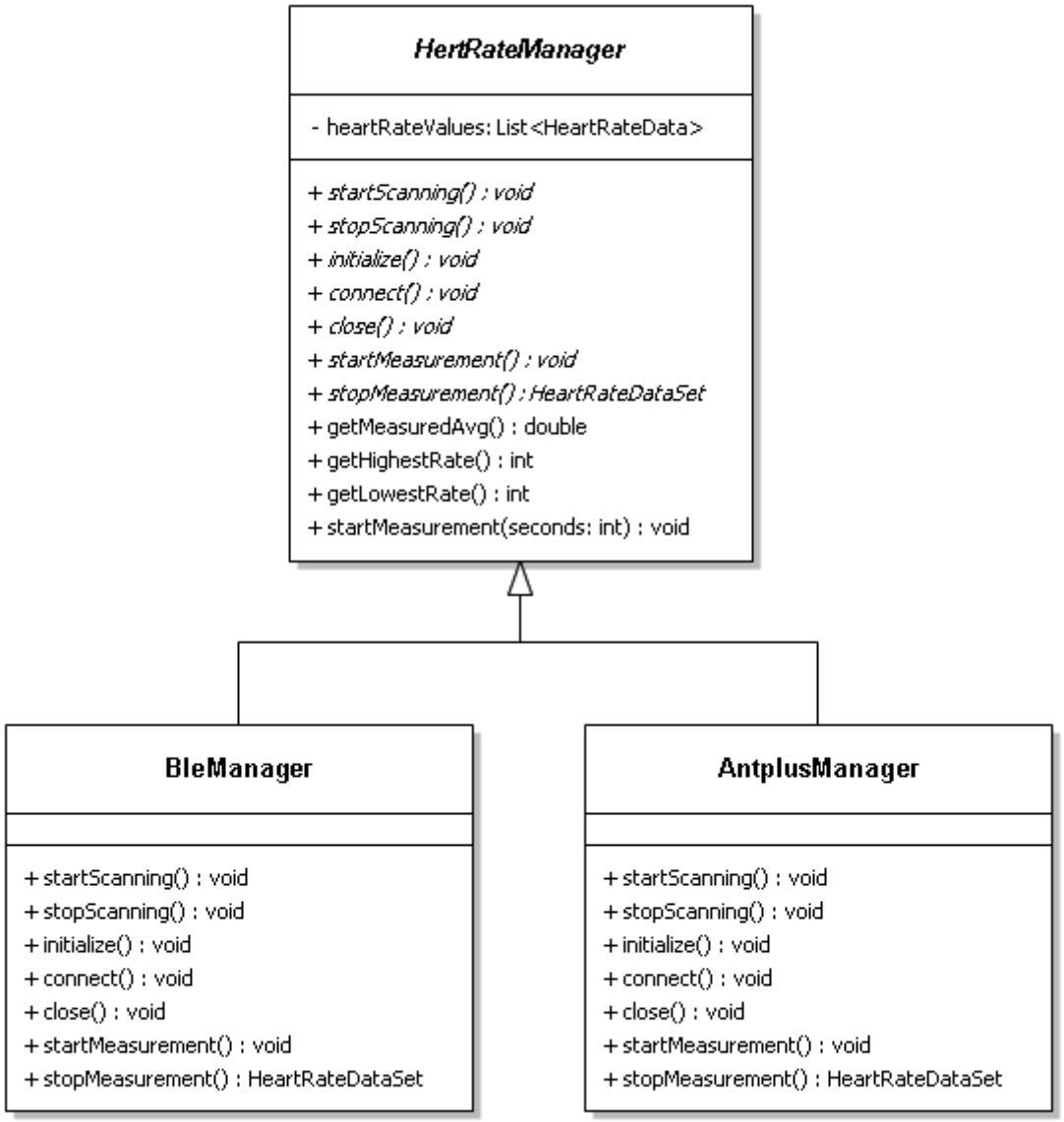


Abbildung 3.2: Klassenhierarchie Managerklassen

Die Manager interagieren mit Sensoren ihres Typs. Die Architektur der Sensoren ist ebenfalls im Stil des Strategiemusters. Außerdem ist in der Schnittstellendefinition der Sensoren ein Feld eines beliebigen Typs für den Originalsensor vorgesehen.

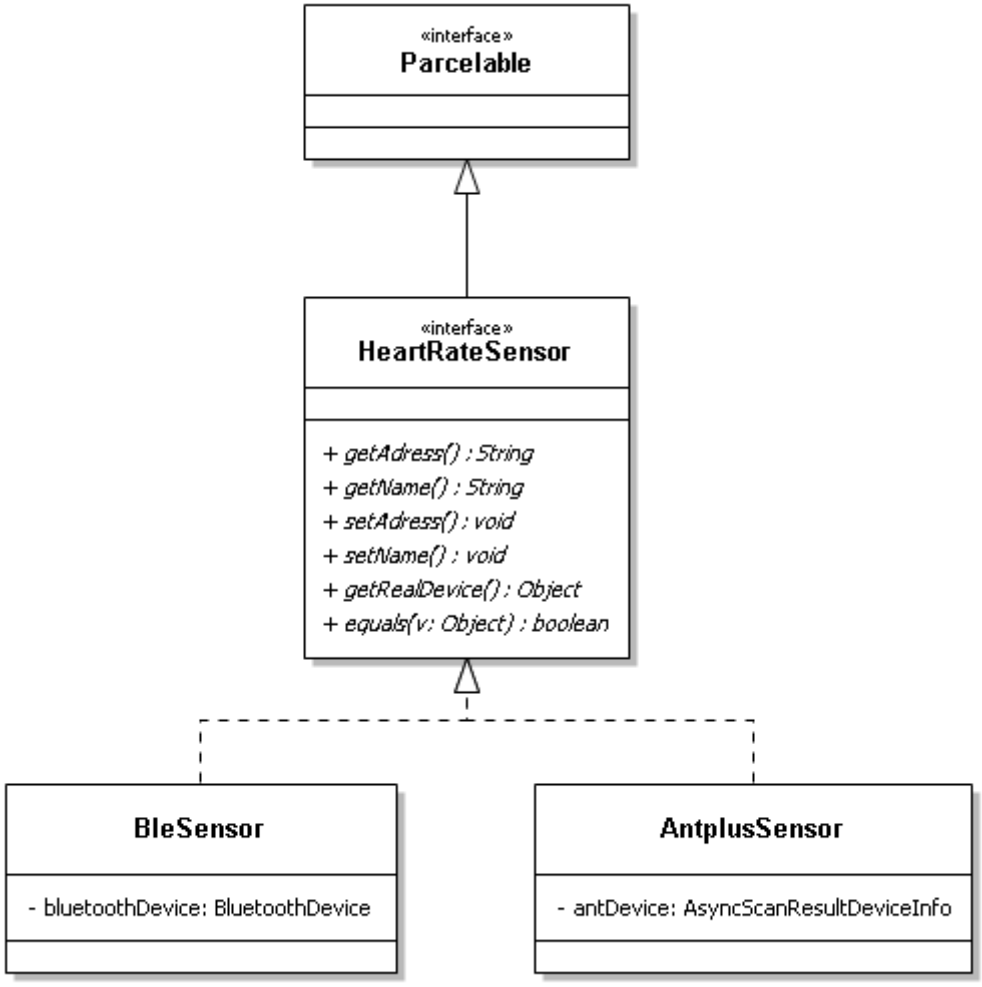


Abbildung 3.3: Klassenhierarchie Sensoren

3.3.1 Callbacks

Da die Anwendung größtenteils asynchron abläuft, muss der Aufrufer zwangsläufig über Änderungen oder neue Ereignisse informiert werden. Es wird ein sogenanntes “Umkehrung der Steuerung“ Paradigma benötigt.

Beobachtermuster

Das Beobachtermuster gehört zu den “Umkehrung der Steuerung“ Paradigmen. In Java wird das Beobachtermuster mit Hilfe der Klasse **Observables** und der Schnittstelle **Observer** realisiert. Objekte, die von **Observable** ableiten, können von Implementierungen der Schnittstelle **Observer** beobachtet werden. Das ist vorallem hilfreich, wenn man eine

Art publish-subscribe Anwendung aufbauen will oder mehrere Beobachter benachrichtigt werden sollen.

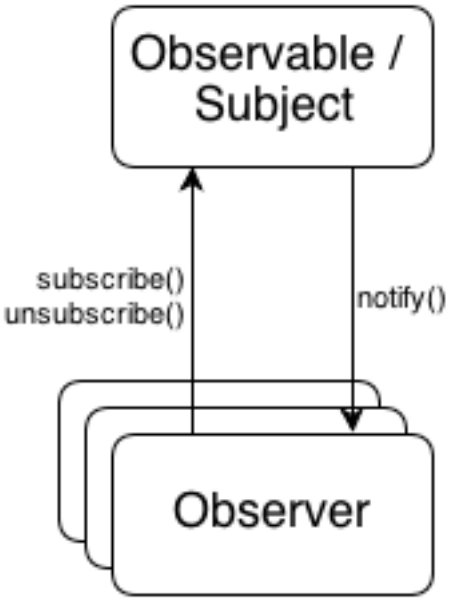


Abbildung 3.4: Prinzip des Beobachtermusters

Klassische Rückruffunktion

Die klassische Rückruffunktion gehört ebenso zu den “Umkehrung der Steuerung“ Paradigmen. Beim Instanziieren des Rahmenwerks werden Methoden über implementierte Schnittstellen registriert, die dann vom Rahmenwerk aufgerufen werden und somit den Programmablauf beeinflussen können.

Da in dieser Arbeit nur eine simple Rückruffunktion benötigt wird, wurde die klassische Rückruffunktion implementiert.

Die Rückruffunktionen werden in der Schnittstelle **HeartRateCallbacks** definiert.

deviceFound
Beim Scan wurde ein Sensor gefunden

connectionStateChanged
Verbindungszustand zum Sensor hat sich verändert

measurementDone
Messung auf Zeit ist abgeschlossen

newHeartrateData
Neuer Herzfrequenz-Wert ist eingetroffen

secondDone
Bei der Messung auf Zeit ist eine Sekunde abgelaufen

```
1 public interface HeartRateCallbacks {
2
3 void deviceFound(final HeartRateSensor device);
4 void connectionStateChanged(HeartRateSensor s,
5 HeartRateManager.CONNECTION state);
6 void  measurementDone(HeartRateDataSet s);
7 void  newHeartrateData(HeartRateData d);
8 void  secondDone(int counter);
9 }
```

Listing 3.1: Interface HeartRateCallbacks

3.3.2 Datenmodell

Für die generierten Daten wird ein entsprechendes Datenmodell benötigt, das die Daten nach einer festgelegten Struktur kapselt. Die Anforderungen an ein Objekt dieses Datenmodells sind folgende:

- Enthält eine gesamte Messung
- Enthält den Maximalwert der Messung
- Enthält den Minimalwert der Messung
- Enthält den Durchschnittswert der Messung
- Enthält alle Einzelwerte der Messung inkl. Timestamp
- Enthält Start- und Endzeitpunkte der Messung

Aus den Anforderungen an das Datenmodell ergibt sich eine zwei Klassen Konstellation.

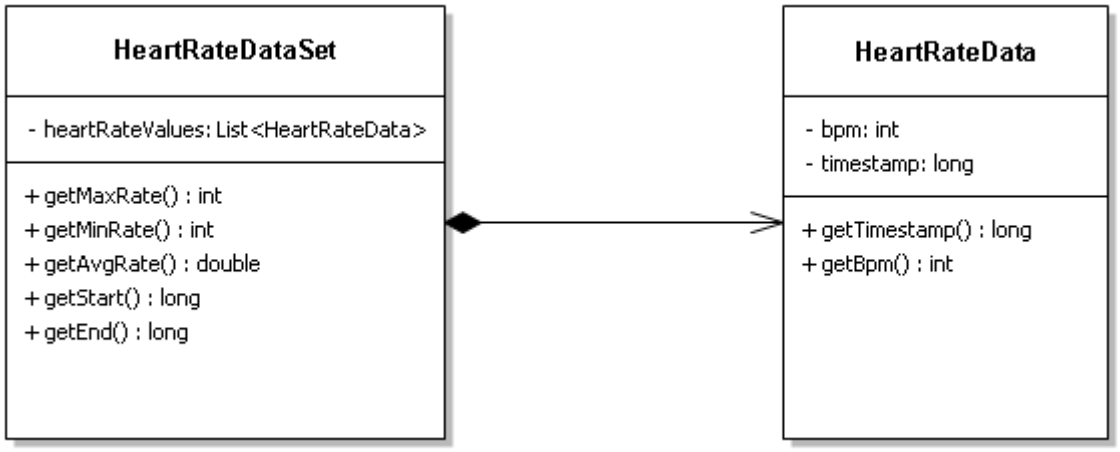


Abbildung 3.5: Datenmodell

3.3.3 Exceptions

Dem üblichen Java Coding Stil entsprechend wurde die HeartRateException, die von Exception abgeleitet ist, definiert um die Ausnahmebehandlung zu vereinfachen und unerwartetes Verhalten besser nachvollziehen zu können.

3.4 Testapplikation

Die Android-Testapplikation wurde entwickelt um das Rahmenwerk zu testen und die Evaluation durchzuführen. Sie ist abgestimmt an die Anforderungen der Evaluation und der Anforderungsanalyse. Mit der Testapplikation soll es möglich sein die angebotenen Funktionen des Rahmenwerks zu nutzen, die in Kapitel 3.2 beschrieben wurden. Unter anderem die Suche nach Geräten, die Verbindung mit Geräten und das Starten von Messungen.

3.4.1 Ziel

Die Testapplikation ist eine Android Applikation, die das beschriebene Rahmenwerk einsetzt. Mit der Testapplikation soll es möglich sein die Umgebung nach vorhandenen, BLE oder ANT+ fähigen, Geräten zu durchsuchen. Die gefundenen Geräte sollen aufgelistet werden und es soll die Möglichkeit bestehen, sich mit diesen zu Verbinden. Anschließend soll eine Herzfrequenzmessung mit dem erfolgreich verbundenen Gerät gestartet werden können. Die dabei produzierten Messdaten sollen persistiert werden und exportierbar sein.

3.4.2 Anforderungsanalyse Testapplikation

Bei der Anforderungsermittlung haben sich die folgenden Anforderungen an die Testapplikation herausgestellt. Die Auflistungen der Anforderungen sind aufgeteilt in funktionale und nicht funktionale Anforderungen.

Nicht funktionale Anforderungen

Nicht funktionale Anforderungen beschreiben Anforderungen an die Nutzbarkeit des Systems. Sie sind auch bekannt als Bedingungs- oder Qualitätsanforderungen und beschreiben wie die Software arbeiten soll [16].

Nummer	Titel	Beschreibung
1	Testen	Das Rahmenwerk soll in der Testapplikation genutzt werden um Herzfrequenzdaten zu erhalten.
2	Persistenz	Die Messdaten müssen in der Testapplikation persistiert werden.
3	Export	Die Messergebnisse müssen exportiert werden können.

Tabelle 3.3: Nicht funktionale Anforderungen der Testapplikation

Funktionale Anforderungen

Funktionale Anforderungen beschreiben die Aufgaben, die die Software lösen kann und das Verhalten der Software [16].

Nummer	Titel	Beschreibung
1	Scan	Die Scan Methode soll gestartet werden können.
2	Scanergebnisse	Die Ergebnisse des Scans sollen durch die Rückrufmethode in einer ListView sichtbar sein.
3	Connect	Die Verbindung mit einem kompatiblen Gerät soll hergestellt werden können.
4	Connection State	Der Connection State soll mithilfe der entsprechenden Rückrufmethode ersichtlich sein.
5	Messung 1	Drei Messungen auf Zeit a 2, 1, 1 Minuten sollen über Buttons gestartet werden können.
6	Stop	Eine Messung soll über einen Button gestoppt werden können.
7	Zeit	Die verbleibende Zeit der Messung soll durch der entsprechenden Rückrufmethode dargestellt werden.
8	Persistenz	Die Ergebnisse der Messung sollen in einer SQLite Datenbank persistiert werden.
9	Export	Die Ergebnisse der Messungen sollen durch eine Export-Methode per Mail exportiert werden können.

Tabelle 3.4: Funktionale Anforderungen der Testapplikation

3.4.3 Architektur

Um die Anforderungen zu realisieren ist eine Oberfläche mit sechs Buttons nötig um die jeweiligen Aktionen auszuführen. Im Fall des Scans wird eine ListView angezeigt, die die gefundenen Geräte verwaltet und auswählbar macht. Zusätzlich ist eine Anzeige des Verbindungsstatus notwendig. Beim Klick auf den Export Button soll die Datenbankdatei als E-Mail anhang versendet werden. Ein Textfeld wird eingefügt um die Testperson zu identifizieren. Das ist in einer Android Activity realisierbar.

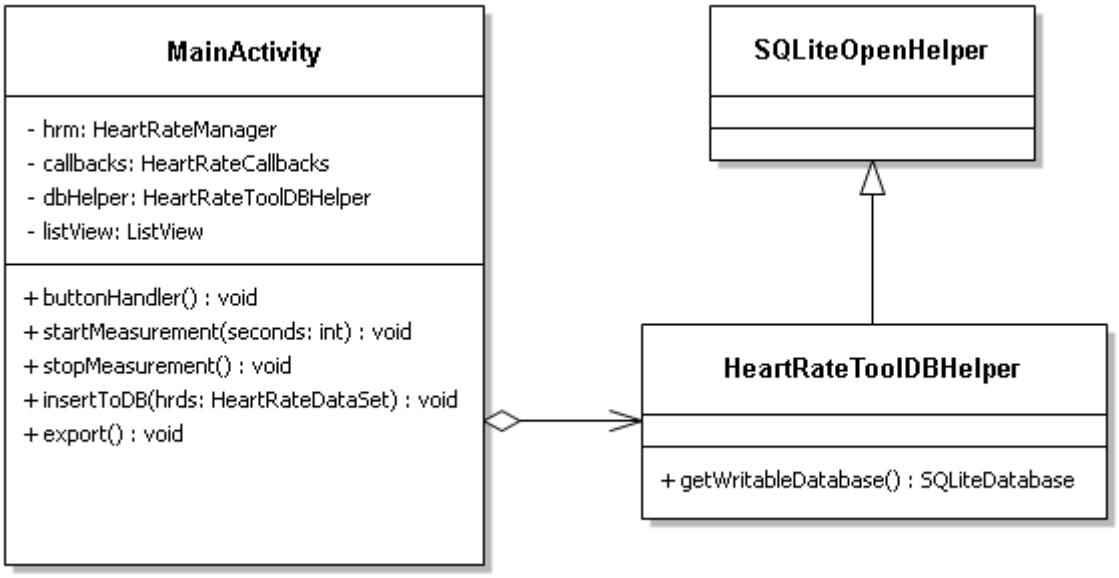


Abbildung 3.6: Architektur Testapplikation

3.5 Zusammenfassung

Das Ziel der Software ist es, ein modulares Rahmenwerk darzustellen durch das die Kommunikation mit gängigen Fitnesstrackern vereinfacht wird. Zudem soll sie einfach erweiterbar sein. Bei der Anforderungsanalyse werden die funktionalen und die nicht funktionalen Anforderungen separat betrachtet. Die Architektur setzt auf das Stragiemuster, um austauschbare Komponenten zu realisieren. Es wird ein geeignetes Datenmodell definiert, das die Messdaten abbildet. Zudem werden eigene Java Exceptions definiert. Die Kommunikation zur Anwendungsapplikation findet über klassische Rückrufmethoden statt. Es wird eine Android Applikation erstellt, um das Rahmenwerk zu testen und die Evaluati-

on durchzuführen. Diese ist sehr simpel aufgebaut und es ist möglich die Messdaten zu persistieren und exportieren.

Kapitel 4

Implementierung

In diesem Kapitel wird auf bestimmte Implementierungsaspekte eingegangen. Hauptsächlich wird die Kommunikation über die Protokolle BLE und ANT+ in den jeweiligen Adaptern beschrieben. Zudem werden Empfehlungen gegeben, wie das Rahmenwerk sinnvoll erweitert werden kann, wenn Bedarf nach neuen Sensoren beziehungsweise Protokollen besteht. Zuletzt wird auf die empfohlene Benutzung des Rahmenwerks eingegangen und Beispiele anhand der Testapplikation gezeigt. Dafür werden Codeausschnitte verwendet, die keinen Anspruch auf Vollständigkeit haben, da sie nur zu Demonstrationszwecken und eventuell gekürzt eingefügt werden.

Auf die Testapplikation wird nur in Bezug zur Anwendung des Rahmenwerks eingegangen, da die Implementierung ansonsten nicht relevant für diese Dokumentation ist.

4.1 Implementierungsdetails

4.1.1 Geräte suchen

Die startScanning() Methode ist eine abstrakte Methode in HeartRateManager und wird von den Adaptern mit Implementierungspflicht geerbt. Sie startet die Suche nach entsprechenden Geräten, die im Fall einer erfolgreichen Suche durch die Rückruffunktion deviceFound(HeartRateSensor device) an die aufrufende Software übergeben werden.

BLE

Bevor man die Systemmethoden der Android Bluetooth API nutzen kann, benötigt man eine Instanz des BluetoothManagers. Dieser verwaltet Bluetooth-Interaktionen auf einer höheren Ebene, wie zum Beispiel Informationen über alle verbundenen Geräte. Über den BluetoothManager bekommt man Zugriff auf eine Instanz des BluetoothAdapters, der für alle Aktionen auf niedriger Ebene zuständig ist. Der BluetoothAdapter ermöglicht es zum

Beispiel die Suche nach Geräten zu starten, Verbindungen zu Geräten herzustellen oder Sockets für eingehende Verbindungen zu erstellen.

```
1  public void initialize() throws HeartRateException {
2      if (mBluetoothManager == null) {
3          mBluetoothManager = (BluetoothManager) mParent.
               getSystemService(Context.BLUETOOTH_SERVICE);
4          if (mBluetoothManager == null) {
5              throw new HeartRateException("Could not get
                   Bluetooth Manager");
6          }
7      }
8
9      if(mBluetoothAdapter == null)
10         mBluetoothAdapter = mBluetoothManager.getAdapter();
11     if (mBluetoothAdapter == null)
12         throw new HeartRateException("Could not get Bluetooth
               Adapter");
13
14
15     if(!mParent.getPackageManager().hasSystemFeature(
           PackageManager.FEATURE_BLUETOOTH_LE))
16         throw new HeartRateException("Hardware does not
               support BLE");
17 }
```

Listing 4.1: init() Methode des BleAdapters des Rahmenwerks

In der Android API 18 wurde BLE eingeführt. Ab API 21 wurde das Vorgehen für einen Scan geändert, somit muss man zur Laufzeit zwischen verschiedenen Android Versionen unterscheiden.

API 18-20

Der Scan wird mit der Methode startLeScan() des BluetoothAdapters gestartet. Als Parameter wird ein Objekt der Klasse LeScanCallback übergeben, um über gefundene Geräte oder Fehler informiert zu werden. Im Fall eines gefundenen Geräts wird die Rückrufmethode onLeScan(...) aufgerufen, die das Gerät an sich, die Signalstärke und den ScanRecord als Byte-Array als Parameter enthält. Der ScanRecord enthält die Advertising Informationen die vom entsprechenden Gerät bereitgestellt werden. Um den ScanRecord einfacher auszuwerten wird die Hilfsklasse AlternateScanRecord, entwickelt der Firma Nordic Semiconductor, verwendet. Aus dem ScanRecord können dann Informationen über die an-

gebotenen Services entnommen werden. In diesem Fall werden Geräte gefiltert, die den Heart Rate Service anbieten [8].

```
if (Build.VERSION.SDK_INT < 21) {
    mBluetoothAdapter.startLeScan(mLeScanCallback  =
    new BluetoothAdapter.LeScanCallback() {
        @Override
        public void onLeScan(final BluetoothDevice device, int
            rssi,
        byte[] scanRecord) {

            BleSensor mdv = new BleSensor(device);
            AlternateScanRecord asr = AlternateScanRecord.
               parseFromBytes(scanRecord);

            if(asr.getServiceUuids().contains(new ParcelUuid(
               UUID.fromString(HEARTRATE_SERVICE_UUID))))
            {
                callBacks.deviceFound(mdv);
                if(!deviceMap.containsKey(mdv.getAdress()))
                    deviceMap.put(mdv.getAdress(), mdv);
            }
        }
    });
}
```

Listing 4.2: init() Ausschnitt startScan Methode des BleAdapters API 18-20

API $\geq$ 21

Ab API 21 wird die Scan Aktion um ein separates Scanner Objekt erweitert. Zudem ist der ScanRecord kein Byte-Array mehr sondern ein Objekt des Typs ScanRecord, somit ist es ohne Hilfsklassen möglich, einfach durch das Objekt zu navigieren. Das Starten des Scans geschieht über den vom BluetoothAdapter bezogenen Scanner mit der Methode startScan(). Auch hier wird ein Object für Rückruffunktionen übergeben, allerdings vom ebenfalls neuen Typ ScanCallback. Durch ein ebenfalls neues ScanSettings Objekt hat man noch die Möglichkeit den Scan zu beeinflussen. Im folgenden wird der Scan Mode SCAN_MODE_LOW_LATENCY verwendet um die bestmöglichste Suche zu gewährleisten.

```
1  ScanSettings.Builder builder = new ScanSettings.Builder();
2  builder.setScanMode(ScanSettings.SCAN_MODE_LOW_LATENCY);
3  if(mBluetoothAdapter != null && builder != null)
4  mBluetoothAdapter.getBluetoothLeScanner().startScan(null,
       builder.build(), mDeviceFoundCallback = new ScanCallback
       () {
5      public void onScanResult(int callbackType, ScanResult
           result) {
6          if (Build.VERSION.SDK_INT > 20)
7          {
8              BleSensor bleSensor = null;
9              bleSensor = new BleSensor(result.getDevice());
10
11             if(bleSensor != null &&  result.getScanRecord() !=
                   null && result.getScanRecord().getServiceUuids()
                    != null && result.getScanRecord().
                   getServiceUuids().contains(new ParcelUuid(UUID.
                   fromString(HEARTRATE_SERVICE_UUID))))
12             {
13                 callBacks.deviceFound(bleSensor);
14
15                 if(!deviceMap.containsKey(bleSensor.getAdress())
                       )
16                     deviceMap.put(bleSensor.getAdress(),
                           bleSensor);
17             }
18         }
19     }
20 }
```

Listing 4.3: init() Ausschnitt startScan Methode des BleAdapters API >= 21

ANT+

Eine ANT API im Android SDK existiert nicht. Die Firma Dynastream Innovations Inc. stellt ein SDK bereit um mit Android Geräten mit ANT+ Geräten zu kommunizieren. Für die Nutzung müssen auf dem ANT+ fähigen Android Smartphone die Applikationen ANT Radio Service und ANT+ Plugins Service installiert sein [4].

Um einen Scan zu starten wird ein ScanController benötigt, den man mit der Methode AntPlusHeartRatePcc.requestAsyncScanController(...) der API anfordern kann. Als Parameter wird auch hier eine Instanz der Callback Schnittstelle AsyncScanControl-

ler.IAsyncScanResultReceiver übergeben, in der die Rückrufmethoden ausimplementiert sind.

In der Rückrufmethode onSearchResult() wird dann das gefundene Gerät behandelt.

```
1  AntPlusHeartRatePcc.requestAsyncScanController(mParent, 0,
2  new AsyncScanController.IAsyncScanResultReceiver()
3  {
4      @Override
5      public void onSearchResult(final AsyncScanController.
           AsyncScanResultDeviceInfo deviceFound)
6      {
7          AntplusSensor gd = new AntplusSensor(deviceFound);
8          callBacks.deviceFound(gd);
9          if(!deviceMap.containsKey(gd.getAdress()))
10             deviceMap.put(gd.getAdress(), gd);
11     }
12 });
```

Listing 4.4: Ausschnitt der startScan Methode in AntplusManager

4.1.2 Verbindung herstellen

Werden entsprechende Geräte gefunden, ist es möglich eine Verbindung zu diesen herzustellen um später die Herzfrequenzdaten anzufordern. Um Informationen zum Verbindungsstatus zurückzugeben wird die Rückrufmethode connectionStateChanged() verwendet.

BLE

Das BluetoothDevice Objekt, das durch den erfolgreichen Scan erzeugt wurde, besitzt die Methode connectGatt(...), welche einen Verbindungsversuch zu dem damit asoziierten Bluetooth Gerät startet. Als Parameter wird der Methode connectGatt(...) ein Objekt des Typs BluetoothGattCallback übergeben, das die Rückrufmethoden realisiert. Der Rückgabewert ist ein BluetoothGatt Objekt, das Methoden bereitstellt um mit den BLE Geräten zu kommunizieren. Daher wird zuvor sichergestellt, ob zu dem entsprechenden Bluetooth Gerät schon ein BluetoothGatt Objekt existiert und wenn ja, wird statt connectGatt(...) ein simples connect() am BluetoothGatt Objekt aufgerufen, was einer erneuten Verbindungsherstellung zum entsprechenden Gerät entspricht.

```
if(mBluetoothGatt!=null)
    mBluetoothGatt.disconnect();
if(mBluetoothGatt != null && mBluetoothGatt.getDevice().
    getAddress().equals(s.getAdress()))
{
    // just reconnect
    if(!mBluetoothGatt.connect())
        throw new HeartRateException("Reconnect failed");
}

BluetoothDevice btDevice = mBluetoothAdapter.getRemoteDevice
    (s.getAdress());
mBluetoothGatt = btDevice.connectGatt(mParent, false,
    btleGattCallback);
if(mBluetoothGatt == null)
    throw new HeartRateException("Connect failed");
```

Listing 4.5: Ausschnitt der connect Methode in BleManager

ANT+

Mit der Methode requestDeviceAccess(...) des ScanControllers wird ein Verbindungsversuch zu einem ANT+ Gerät gestartet. Als Methodenparameter werden das AsyncScanController.AsyncScanResultDeviceInfo Objekt, welches beim erfolgreichen Scan erstellt wurde und zwei Objekte für die Rückruffunktionen, übergeben. Das erste Rückrufobjekt ist vom Typ AntPluginPcc.IPluginAccessResultReceiver und behandelt die Rückmeldung bezüglich dem Verbindungsvorgang. Das zweite Rückrufobjekt ist vom Typ AntPluginPcc.IPluginAccessResultReceiver und behandelt Rückmeldungen die Zustandsänderungen der Verbindung beinhalten. Bei einem erfolgreichen Verbindungsaufbau wird ein Objekt vom Typ AntPlusHeartRatePcc zurückgegeben, mit dem die Kommunikation zum ANT+ Gerät stattfindet.

```
hrScanCtrl.requestDeviceAccess(asyncScanResultDeviceInfo,
new AntPluginPcc.IPluginAccessResultReceiver<
    AntPlusHeartRatePcc>()
{
    @Override
    public void onResultReceived(AntPlusHeartRatePcc result,
        RequestAccessResult resultCode, DeviceState
            initialDeviceState)
        {
        if(resultCode == RequestAccessResult.SEARCH_TIMEOUT)
```

```
9         {
10            callBacks.connectionStateChanged(new AntplusSensor(
                  asyncScanResultDeviceInfo), CONNECTION.
                  DISCONNECTED);
11        }
12        else
13        {
14            callBacks.connectionStateChanged(new AntplusSensor(
                  asyncScanResultDeviceInfo), CONNECTION.CONNECTED
                  );
15            base_IPluginAccessResultReceiver.onResultReceived(
                  result, resultCode, initialDeviceState);
16            heartRatePcc = result;
17        }
18    }
19 }, iDeviceStateChangeReceiver);
```

Listing 4.6: Ausschnitt der connect Methode in AntplusManager

4.1.3 Messung starten

Wenn die Verbindung zu einem Fitnesstracker hergestellt ist, ist es möglich die Herzfrequenzmessung zu starten. Es gibt die Möglichkeit die Messung für eine beliebige oder für eine begrenze Zeit lang zu starten. Es gibt folgende Rückrufmethoden, die in diesem Zusammenhang aufgerufen werden können:

- **newHeartRateData**: neues Datum eingetroffen
- **secondDone**: eine Sekunde der Messung ist vorüber
- **measurementDone**: die Messung ist fertig

BLE

Anhand des BluetoothGatt Objekts werden die angebotenen Services des verbundenen Geräts ermittelt. Anschließend wird der die HeartrateMeasurement Characteristic herausgefiltert und der entsprechende Descriptor angefordert. Mit Hilfe der Characteristic und des Descriptors ist es möglich die Notifikation über die Herzfrequenz zu aktivieren. Dazu wird der Rückruf aktiviert, der bei Änderung einer Characteristic aufgerufen wird und anschließend der Descriptor mit der Option BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE an das Gerät gesendet.

Änderungen des Werts der Characteristic führen nun zum Aufruf der Rückrufmethode onCharacteristicChanged(...) im BluetoothGattCallback Objekt.

```
List<BluetoothGattService> services = mBluetoothGatt.
   getServices();
for (BluetoothGattService service : services) {
   List<BluetoothGattCharacteristic> characteristics =
      service.getCharacteristics();
   if(service.getUuid().toString().equals(
      HEARTRATE_SERVICE_UUID))
   {
      for(BluetoothGattCharacteristic c : characteristics) {
         if(c.getUuid().toString().equals(
            HEARTRATE_MEASUREMENT_CHARACTERISTIC_UUID))
         {
            List<BluetoothGattDescriptor> descriptors;
            if((descriptors = c.getDescriptors()) != null &&
                descriptors.size() > 0)
            {
               BluetoothGattDescriptor descriptor =
                  descriptors.get(0);
               if(descriptor.getUuid().toString().equals(
                  CLIENT_CHARACTERISTIC_CONFIGURATION_DESCRIPTOR_UUID
                  ))
               {
                  boolean success = mBluetoothGatt.
                     setCharacteristicNotification(c, true);
                  if(!success) {
                     throw new HeartRateException("Setting
                        proper notification status for
                        characteristic failed!");
                  }
                  descriptor.setValue(
                     BluetoothGattDescriptor.
                     ENABLE_NOTIFICATION_VALUE);
                  mBluetoothGatt.writeDescriptor(descriptor)
                     ;
               }
               throw new MeasurementException("GATT
                  Descriptor not available");
```

```
23                 }
24             }
25         }
26     }
27 }
```

Listing 4.7: Ausschnitt der startMeasurement Methode in BleManager

ANT+

Über das durch das Herstellen der Verbindung erstellte Objekt der Klasse AntPlusHeartRatePcc, mit dem die Kommunikation mit dem ANT+ Gerät realisiert wird, lässt sich das Senden der Herzfrequenz aktivieren. Dazu wird die subscribeHeartRateDataEvent(...) Methode aufgerufen. Als Parameter wird eine Instanz der Schnittstelle AntPlusHeartRatePcc.IHeartRateDataReceiver() für Rückruffunktionen übergeben. In der Rückrufmethode onNewHeartRateData() wird dann das eingetroffene Datum behandelt.

```
1  public void subscribeToHrEvents()
2  {
3
4      hrPcc.subscribeHeartRateDataEvent(new AntPlusHeartRatePcc
           .IHeartRateDataReceiver()
5      {
6          @Override
7          public void onNewHeartRateData(final long estTimestamp
               , EnumSet<EventFlag> eventFlags,
8          final int computedHeartRate, final long heartBeatCount
               ,
9          final BigDecimal heartBeatEventTime, final
               AntPlusHeartRatePcc.DataState dataState)
10         {
11             final String textHeartRate = String.valueOf(
                   computedHeartRate)
12             + ((AntPlusHeartRatePcc.DataState.ZERO_DETECTED.
                   equals(dataState)) ? "*" : "");
13
14             ...
15
16         }
17     }
18 }
```

Listing 4.8: Ausschnitt der startMeasurement Methode in AntplusManager

4.2 Erweiterungsmöglichkeit

Kann das Rahmenwerk die Anforderungen für einen bestimmten Typ eines Fitnesstrackers nicht erfüllen und bietet dieser eine Möglichkeit um mit der Android SDK zu kommunizieren, ist es möglich das Rahmenwerk zu erweitern. Im Folgenden wird erläutert, wie man hier praktisch vorgehen kann.

4.2.1 Sensor

Für die Geräte des neuen Protokolls wird stellvertretend eine neue Sensor Klasse erstellt. Um die Geräte mit dem Rahmenwerk kompatibel zu machen, implementiert diese Klasse die Schnittstelle HeartRateSensor. Neben den Getter und Setter Methoden kann, je nach zu implementierendem Protokoll, ein protokollspezifisches Objekt abgelegt werden, falls es zur internen Kommunikation nötig ist. Um den Austausch der Sensoren über verschiedene Activitys zu ermöglichen, erweitert die HeartRateSensor Schnittstelle die Parcelable Schnittstelle. Diese Methoden gilt es ebenfalls entsprechend zu implementieren.

4.2.2 Manager Klasse

Die neu erstellte Manager Klasse muss von der abstrakten Klasse HeartRateManager ableiten. Anschließend sind die abstrakten Methoden zu implementieren.

```
1  public class XManager extends HeartRateManager {
2      public XManager(HeartRateCallbacks callbacks) throws
           HeartRateException
3      {
4          super(callbacks);
5      }
6      @Override
7      public void startScanning() {
8      }
9
10     @Override
11     public void stopScanning() {
12     }
13
14     @Override
15     public void connect(HeartRateSensor s) throws
           HeartRateException{
```

```
16      }
17
18      @Override
19      public void close() {
20      }
21
22      @Override
23      public void startMeasurement() throws HeartRateException{
24      }
25      @Override
26      public HeartRateDataSet stopMeasurement() {
27          //code for stoping measurement
28          running = false;
29          return getHeartRateDataSet();
30      }
31  }
```

Listing 4.9: Beispiel einer Manager Klasse

Konstruktor

Im Konstruktor muss mindestens ein Objekt der Klasse HeartRateCallbacks übergeben werden. Dieses muss dann mit super an die Basisklasse übergeben werden.

startScanning

In dieser Klasse wird der Scan nach Geräten des neu zu implementierenden Protokolls realisiert. Bei neu gefundenen Geräten empfiehlt es sich, diese in der Hashmap deviceMap abzulegen, um den aktuellen Status der gefundenen Geräte zu halten. Gefundene Geräte sollten mit callBacks.deviceFound(HeartRateSensor) an die aufrufende Anwendung übergeben werden.

connect

Implementation des Verbindungsaufbaus, der Parameter ist das Gerät mit dem die Verbindung aufgebaut werden soll. Der zu nutzende Rückruf ist connectionStateChanged(...).

startMeasurement

Hier soll dem verbundenen Gerät signalisiert werden, dass Herzfrequenzdaten gesendet werden sollen. Dabei sollte man folgendes beachten. Um die Messung auf Zeit korrekt zu implementieren, sollte man beim Eintreffen des ersten Wertes folgendes Codegerüst verwenden. Im Falle einer Messung auf Zeit werden in der Basisklasse die Variablen timer_started und running auf true und counter auf die entsprechende Sekundenanzahl gesetzt. In diesem Fall sollte man beim ersten Herzfrequenzdatum den Timer starten.

Damit der Timer nur einmal gestartet wird, wird timer_started anschließend wieder auf false gesetzt. Im Timer-Thread wird dann der counter sekundenweise herunter gezählt und entsprechend der secondDone(...) Rückruf aufgerufen. Das eingetroffene Herzdatum wird mit der Rückruffunktion newHeartrateData(...) übergeben und sollte der ArrayList heartRateValues hinzugefügt werden.

```
1  //Neuer Wert eingetroffen
2   if(timer_started)
3  {
4      timer_started = false;
5      new Thread(new Runnable() {
6          public void run() {
7              while (counter > 0 && running) {
8                  try {
9                      Thread.sleep(1000);
10                 }
11                 catch (InterruptedException e) {
12                 }
13                 counter--;
14                 callBacks.secondDone(counter);
15             }
16             if(running)
17             {
18                 callBacks.measurementDone(stopMeasurement());
19                 running = false;
20             }
21         }
22     }).start();
23 }
24
25 if(running)
26 {
27     Date d = new Date();
28     int heartRate=0; //TODO set heart frequency value
29     HeartRateData tmp = new HeartRateData(heartRate, d.
           getTime());
30     if(callBacks != null)
31         callBacks.newHeartrateData(tmp);
32     getHeartRateValues().add(tmp);
```

```
33 }
```

Listing 4.10: Beispiel der Implementierung beim Eintreffen eines neuen Datums

stopMeasurement

Hier soll dem verbundenen Gerät signalisiert werden, dass Herzfrequenzdaten nicht mehr gesendet werden sollen. Der Rückgabewert sollte die heartRateValues ArrayList sein.

4.3 Benutzung

Die Benutzung des Rahmenwerks war in den vorherigen Abschnitten schon teilweise ersichtlich. In diesem Abschnitt wird genauer erläutert, wie man das Rahmenwerk in eine Android Applikation einbauen kann.

Rechte

In der AndroidManifest Datei müssen die BLUETOOTH und BLUETOOTH_ADMIN Rechte gesetzt werden. Für eine Nutzung unter Android 6.0 muss zusätzlich das Recht ACCESS_FINE_LOCATION oder ACCESS_COARSE_LOCATION gesetzt werden [1].

Rückruffunktionen

Ein Objekt einer Klasse die HeartRateCallbacks implementiert muss vor dem erstellen der Instanz des Rahmenwerks erstellt werden. Hierfür kann man die Null Klasse benutzen und die benötigten Rückrufmethoden überschreiben.

```
1  callbacks = new HeartRateCallbacks.Null() {
2  @Override
3  public void connectionStateChanged(HeartRateSensor s,
       HeartRateManager.CONNECTION state){
4  if(state == HeartRateManager.CONNECTION.CONNECTED)
5  ...
6  }
7  @Override
8  public void measurementDone(HeartRateDataSet data){
9  ...
10 }
11
12 @Override
13 public void deviceFound(final HeartRateSensor device){
14 ...
15 }
16 @Override
```

```
17 public void newHeartrateData(HeartRateData d){
18 ...
19 }
20
21 @Override public void secondDone(int counter)
22 {
23 ...
24 }
25 };
```

Listing 4.11: Beispiel Erstellung Rückrufobjekt

Managerobjekt erstellen

Nachdem der das Rückrufobjekt erstellt wurde, kann eine Instanz des entsprechenden Managers erstellt werden, der das Rückrufobjekt übergeben wird.

```
1  HeartRateManager hrm;
2  try{
3      if(DEVICE_TYPE == HeartrateEnum.DEVICE_TYPE_MIO)
4      {
5          hrm = new BleManager(this, callbacks);
6      }else
7      {
8          hrm = new AntplusManager(this, callbacks);
9      }
10 }catch(HeartRateException e)
11 {
12     ...
13 }
```

Listing 4.12: Erstellung einer Manager Instanz

Danach lassen sich die Methoden startScanning(), stopScanning(), connect(...), startMeasurement(...), stopMeasurement() in Kombination mit den Rückrufmethoden nutzen.

Kapitel 5

Anforderungsabgleich

In diesem Kapitel werden die Anforderungen aus Kapitel 3 erneut aufgegriffen und mit den Funktionen des implementierten Rahmenwerk verglichen.

5.1 Nicht funktionale Anforderungen

Nummer	Titel	Beschreibung
1	Herzfrequenzdaten	Anforderung erfüllt. Der Anwender des Frameworks kann Herzfrequenzdaten im festgelegten Datenformat von verschiedenen Fitnesstrackern beziehen, ohne sich mit den Eigenschaften der einzelnen Protokollen zu beschäftigen.
2	Modulare Architektur	Anforderung erfüllt. Durch die schnittstellenbasierte Architektur nach dem Strategiemuster ist eine Erweiterung problemlos möglich, siehe Kapitel 4.2, Erweiterungsmöglichkeit. Durch die Generalisierungen ziehen Änderungen im Rahmenwerk keine oder wenige Änderungen in der Anwendung nach sich.
3	Ab Android 4.3	Anforderung erfüllt. Android Smartphones ab Android Version 4.3 werden vom Rahmenwerk unterstützt. Siehe Kapitel 2.7.5, eingesetzte Hardware.
4	Ab Android 5.0	Anforderung erfüllt. Es werden ab Android API 21 die neuen SDK Methoden und Klassen verwendet. Siehe Kapitel 4

5.2 Funktionale Anforderungen

Nummer	Titel	Beschreibung
1	Typ	Anforderung erfüllt. Siehe Kapitel 4.3, Manager Objekt erstellen.
2	Scan	Anforderung erfüllt. Die Methode startScanning(...) erfüllt diese Anforderung. Siehe Kapitel 4.1.1.
3	Scan Callback	Anforderung erfüllt. Die Rückrufmethode deviceFound(...) wird aufgerufen, sobald ein Gerät gefunden wurde.
4	Connect	Anforderung erfüllt. Siehe Kapitel 4.1.2 Verbindung herstellen.
5	Connect to Adress	Anforderung erfüllt. Übergibt man der connect(...) Methode statt dem HeartRate-Sensor nur die Adresse, wird ebenfalls eine Verbindung hergestellt.
6	Connection State Callback	Anforderung erfüllt. Ändert sich der Zustand der Verbindung, wird dies über die Rückrufmethode connectionStateChanged(...) zurückgegeben.
7	Start Measurement	Anforderung erfüllt. Siehe Kapitel 4.1.3 Messung starten.
8	Start Measurement 2	Anforderung erfüllt. Die Messung auf Zeit wird durch einen Timer im Rahmenwerk realisiert.
9	New Value Callback	Anforderung erfüllt. Beim Eintreffen eines neuen Herzfrequenzdatums wird die Rückrufmethode newHeartrateData(...) aufgerufen.
10	Second Callback	Anforderung erfüllt. Nach jeder vergangenen Sekunde wird die Rückrufmethode secondDone(...) aufgerufen.
11	Finished Callback	Anforderung erfüllt. Nach dem erfolgreichen Beenden einer Messung wird die Rückrufmethode measurementDone(...) aufgerufen.
12	Stop Measurement	Anforderung erfüllt. Mit der Methode stopMeasurement(...) kann die Messung gestoppt werden und die Messergebnisse können bezogen werden.

Kapitel 6

Evaluation

In diesem Kapitel folgt die Auswertung der durchgeführten Evaluation. Da es in der Evaluierung hauptsächlich um das Testen des Rahmenwerks geht, werden die Ergebnisse der Evaluation lediglich deskriptiv aufgezeigt. In der Evaluation wird das entwickelte Rahmenwerk getestet. Dazu wird die Testapplikation, die in Kapitel 3.4 beschrieben wird, verwendet. Mit der Testapplikation werden an 20 Testpersonen Herzfrequenzmessungen durchgeführt. Die Testpersonen werden vorab und anschließend durch einen Fragebogen im Zusammenhang mit der Evaluation befragt.

6.1 Ziel

Das Ziel der Evaluation ist zum Einen das Testen des Rahmenwerks im Sinne eines Systemtests.

> "*System testing is concerned with testing the behavior of an entire system. Effective unit and integration testing will have identified many of the software defects. System testing is usually considered appropriate for assessing the nonfunctional system requirements—such as security, speed, accuracy, and reliability (see Functional and Non-Functional Requirements in the Software Requirements KA and Software Quality Requirements in the Software Quality KA). External interfaces to other applications, utilities, hardware devices, or the operating environments are also usually evaluated at this level*.„ [16]

Außerdem wird durch die Auswertung der Messergebnisse ein grobes Bild der Genauigkeit der eingesetzten Fitnesstracker generiert. Durch die Befragung der Testpersonen soll eine Einschätzung der Praxistauglichkeit gegeben werden.

6.2 Szenario

In Zusammenarbeit mit der Tinnitus Research Initiative wurde ein Szenario entwickelt, das die Anforderungen an die Evaluation erfüllt. Für die Messung werden zwei Fitnesstracker des Herstellers Mio Global verwendet, die vom DBIS Institut bereitgestellt werden. Die Testpersonen tragen beide Fitnesstracker parallel. Am rechten Arm befindet sich das Mio Alpha 2 und am linken Arm das Mio Alpha Armband. An den Geräten wird anschließend der Bluetooth Herzfrequenz-Sendemodus aktiviert. Danach wird folgendes Testprotokoll angewandt.

Tabelle 6.1: Testprotokoll Evaluation

Aktivität	**Dauer**	**Zweck**
Sitzen	1 Minute	finden des Pulses
Sitzen	2 Minute	Messen des Ruhepulses
Gehen	1 Minute	Messen des Pulses beim Gehen
Gehen und Fragebogen beantworten	1 Minute	Messen des Pulses beim Gehen, während Fragebogen ausgefüllt wird

6.3 Fragebögen

Testfragebogen

Es wurde ein Fragebogen erstellt, der Informationen zu den Testpersonen aufnimmt. Zudem wird vor der Messung eine Einschätzung des aktuellen Pulses verlangt. Im Anschluss der Messung wird der subjektive Aufwand der Messung abgefragt. Außerdem wird noch das Verhältnis zur Durchführung von Pulsmessungen und Nutzung von Fitnesstrackern abgefragt. Der Fragebogen ist im Anhang A.1 ersichtlich .

Fragebogen während der Messung

Als Fragebogen während der Messung wurde eine geeignete Fragebogen Smartphone Applikation gesucht, um die Umstände der späteren Benutzung des Rahmenwerks zu simulieren. Es wurde eine Applikation verwendet die kostenlos ist, offline benutzbar ist und relativ wenig komplexe Fragen beinhaltet. Die Android Applikation “Führerschein App 2018 - Fahrschule Theorie“ lässt sich im PlayStore finden und über diesen installieren. Für den Testlauf wurden zufällige Fragen ausgesucht.

6.4 Testpersonen

Bei der Evaluation nahmen 20 Testpersonen teil. Davon waren 10 Frauen und 10 Männer. Das Durchschnittsalter betrug 26,9 Jahre. Die älteste Person war 61 Jahre alt, die jüngste Person war 17 Jahre alt. Die Hautfarbe der getesteten Personen ist weiß.

Tabelle 6.2: Testpersonen

Nr.	Geschlecht	Alter	Jahre der Ausbildung ab Grundschule
1	w	26	16
2	m	26	15
3	m	26	13
4	m	26	18
5	m	26	20
6	w	27	17
7	m	25	18
8	w	61	11
9	m	17	12
10	w	24	13
11	m	28	19
12	m	26	18
13	w	28	16
14	w	24	17
15	w	28	18
16	w	25	18
17	w	25	18
18	m	25	19
19	m	23	16
20	w	22	16

6.5 Ergebnisse

Die Messdaten die bei der Evaluation entstanden sind, werden im Folgenden aufbereitet. Hierbei werden Durchschnittswerte beider Sensoren betrachtet und die Ergebnisse beider Sensoren anhand ihrer Korrelation verglichen.

Der durchschnittliche Ruhepuls über beide Sensoren bei der “Baseline“ Messung lag bei 69,36 Schlägen pro Minute. Der durchschnittliche Puls über beide Sensoren bei der “Gehen“ Messung lag bei 89,87 Schläge pro Minute. Der durchschnittliche Puls über beide Sensoren bei der “Gehen+Fragebogen“ Messung lag bei 85,16 Schlägen pro Minute.

Der durchschnittliche Ruhepuls der “Baseline“ Messung des Mio Alpha 2 beträgt 69,03 Schläge pro Minute. Der durchschnittliche Ruhepuls der Messung des Mio Alpha 1 beträgt 69,7 Schläge pro Minute. Somit liegt die durchschnittliche Abweichung beider Sensoren bei der “Baseline“ Messung bei 0,67 Schlägen pro Minute. Der Korrelationskoeffizient beider Sensoren liegt bei der “Baseline“ Messung bei 0,799.

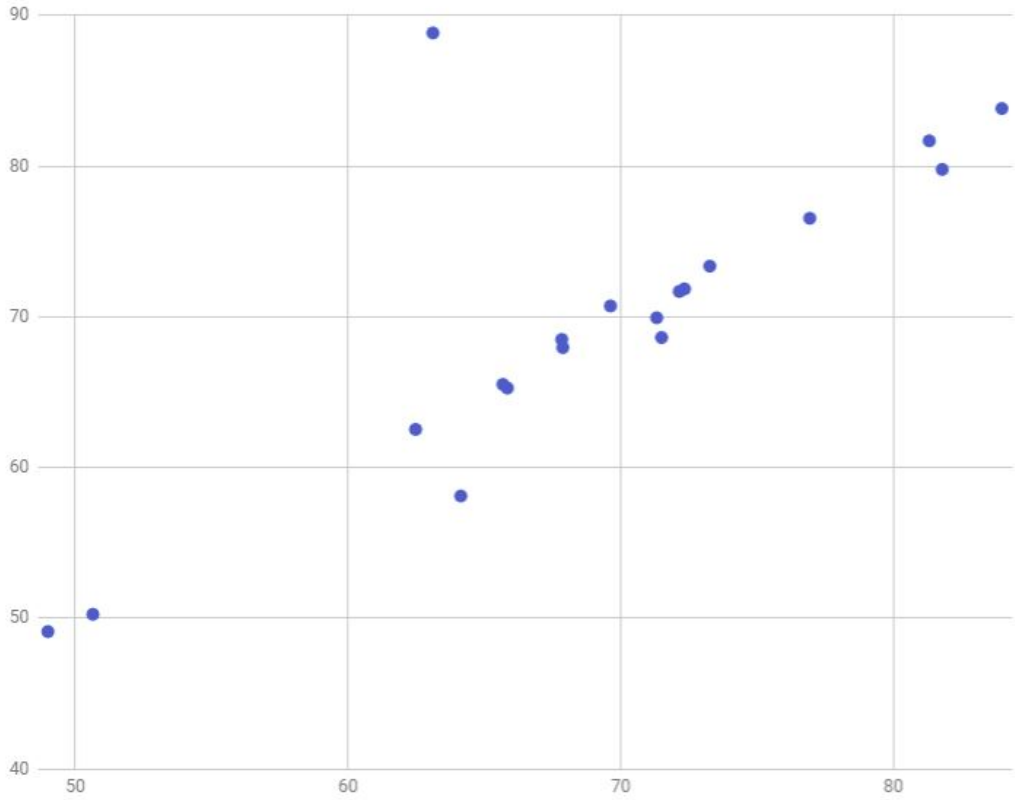


Abbildung 6.1: Bildliche Darstellung der Korrelation bei der “Baseline“ Messung

Der durchschnittliche Puls der “Gehen“ Messung des Mio Alpha 2 beträgt 88,75 Schläge pro Minute. Der durchschnittliche Puls der “Gehen“ Messung des Mio Alpha 1 beträgt

90,99 Schläge pro Minute. Somit liegt die durchschnittliche Abweichung beider Sensoren bei der “Gehen“ Messung bei 2,24 Schlägen pro Minute. Der Korrelationskoeffizient beider Sensoren liegt bei der “Gehen“ Messung bei 0,49.

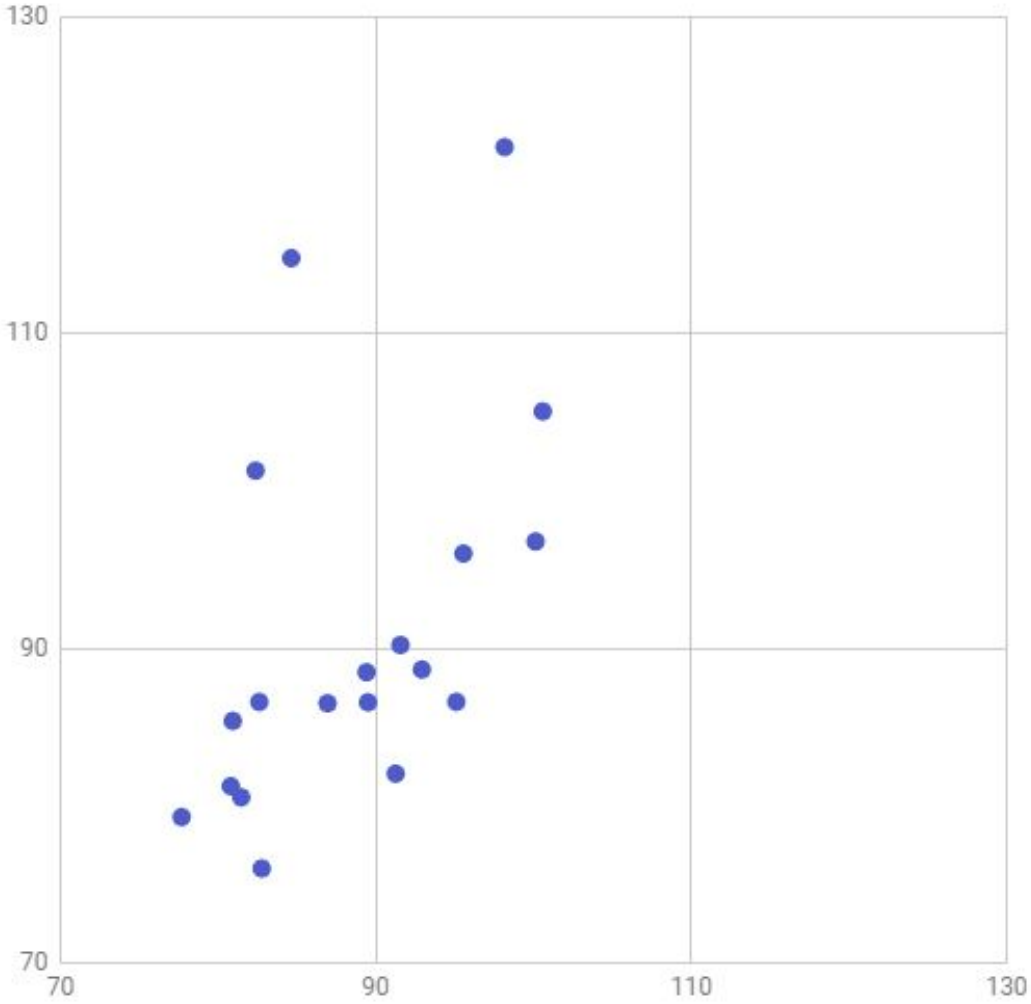


Abbildung 6.2: Bildliche Darstellung der Korrelation bei der “Gehen“ Messung

Der durchschnittliche Puls der “Gehen+Fragebogen“ Messung des Mio Alpha 2 beträgt 87,59 Schläge pro Minute. Der durchschnittliche Puls der “Gehen+Fragebogen“ Messung des Mio Alpha 1 beträgt 82,74 Schläge pro Minute. Somit liegt die durchschnittliche Abweichung beider Sensoren bei der “Gehen+Fragebogen“ Messung bei 4,85 Schlägen pro Minute. Der Korrelationskoeffizient beider Sensoren liegt bei der “Gehen+Fragebogen“ Messung bei 0,69.

Im Schnitt war der Puls bei der “Gehen+Fragebogen“ Messung 5 Schläge pro Minute **niedriger** als bei der “Gehen“ Messung. Beim Mio Alpha 2 war der Puls bei der “Gehen+Fragebogen“ Messung 1,16 Schläge pro Minute **niedriger** als bei der “Gehen“ Mes-

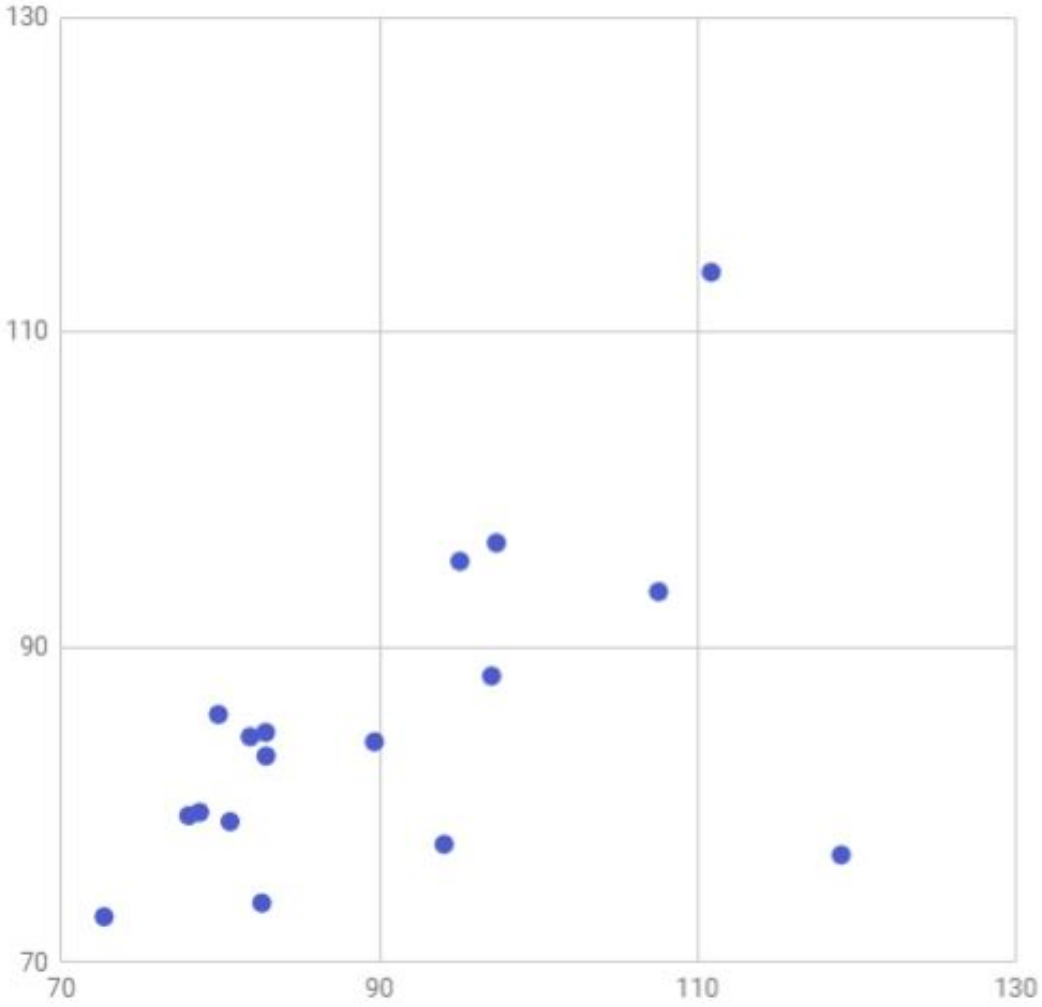


Abbildung 6.3: Bildliche Darstellung der Korrelation bei der “Gehen+Fragebogen“ Messung

sung. Beim Mio Alpha 1 war der Puls bei der “Gehen+Fragebogen“ Messung 8,25 Schläge pro Minute **niedriger** als bei der “Gehen“ Messung.

Die durchschnittliche Schätzung der Testpersonen ihres Ruhepules lag bei 70,45 Schlägen pro Minute. Somit ergibt sich eine Abweichung zum durchschnittlich gemessenen Ruhepuls über beide Sensoren von 1,09 Schlägen pro Minute, über den Mio Alpha 2 von 1,45 Schlägen pro Minute und über den Mio Alpha 1 von 0,75 Schlägen pro Minute.

Die anschließende Befragung zur Evaluation kam zu folgendem Ergebnis.

Tabelle 6.3: Ergebnis der Befragung

Nr.	Geschätzter Ruhepuls	War die Messung aufwändig?	Regelmäßig Fitness-tracker im Einsatz?	Regelmäßige Puls-messung?
1	68	nein	nein	ja
2	60	nein	nein	nein
3	72	nein	nein	nein
4	55	nein	nein	ja
5	60	ja	ja	neutral
6	90	nein	nein	nein
7	60	ja	nein	nein
8	60	neutral	nein	nein
9	70	ja	nein	nein
10	60	neutral	nein	nein
11	60	ja	nein	nein
12	80	nein	nein	nein
13	82	nein	nein	nein
14	70	nein	nein	nein
15	80	neutral	nein	neutral
16	60	nein	nein	nein
17	85	nein	nein	nein
18	70	nein	nein	nein
19	105	nein	nein	neutral
20	62	nein	nein	nein
Gesamt	70,45	4 ja | 3 neutral | 13 nein	1 ja | 0 neutral | 19 nein	2 ja | 3 neutral | 15 nein

Tabelle 6.4: Mittelwerte der einzelnen Messungen

Nr.	S1 Baseline	S1 Gehen	S1 Gehen + Fragebogen	S2 Baseline	S2 Gehen	S2 Gehen + Fragebogen
1	69,78813559	90,42105263	90,38983051	70,07563025	86,16393443	86,44262295
2	62,47663551	80,81034483	78,69090909	62,5210084	81,24137931	79,54237288
3	65,84158416	77,69387755	79,87037037	65,26086957	79,28070175	85,75409836
4	48,99137931	86,94545455	82,86666667	49,10526316	86,5	83,11666667
5	67,83653846	91,56862745	82,84210526	68,49137931	90,19672131	84,61403509
6	72,33628319	100,5909091	107,5535714	71,84482759	105,0172414	93,55
7	69,62184874	95,55932203	97,35294118	70,71428571	96	96,64814815
8	63,1092437	89,50877193	78	88,83050847	86,55932203	79,33333333
9	64,13392857	81,47368421	72,68333333	58,10169492	80,53333333	72,91071429
10	76,92982456	91,26415094	94,06896552	76,52991453	82,03278689	77,50909091
11	72,14529915	80,94	80,60655738	71,6754386	85,37931034	78,94915254
12	67,87394958	82,38333333	81,86666667	67,94827586	101,2586207	84,33928571
13	65,6779661	89,41666667	82,60344828	65,50909091	88,46808511	73,77966102
14	83,97368421	84,64583333	95,05769231	83,81355932	114,7413793	95,48333333
15	50,64102564	95,11111111	119,0327869	50,25210084	86,59016393	76,83606557
16	81,31092437	100,1355932	97,04918033	81,67226891	96,76666667	88,19672131
17	71,4957265	82,61818182	62,2	68,61538462	86,58333333	54,45762712
18	71,31683168	82,77777778	68,5	69,92792793	76,01886792	65,4137931
19	73,26315789	92,92727273	89,68518519	73,35344828	88,64150943	84,01923077
20	81,78632479	98,16949153	110,862069	79,77391304	121,7868852	113,8363636
Schnitt	69,02751459	88,74807284	87,58911397	69,70083951	90,98801212	82,73661584

Auffälligkeiten

Die Messungen Mio Alpha 2 konnten nicht immer auf Anhieb korrekt durchgeführt werden. Aus bisher nicht bekannten Gründen wurde während der Messung die Verbindung zwischen dem Smartphone und dem Fitnesstracker unterbrochen. Daraufhin wurde die Messung wiederholt.

6.6 Fazit

Ziel dieser Evaluation war es, das Rahmenwerk auf Funktion zu prüfen. Dieser Test wurde erfolgreich anhand 20 Testpersonen mit jeweils 6 Messungen aufgeteilt auf zwei Fitnesstracker durchgeführt. Das geschilderte Evaluationsszenario wurde entsprechend durchgeführt, die Ergebnisse wurden detailliert verglichen und beschrieben. Die Ergebnisse sind für eine weitere Betrachtung und eine Auswertung bereit.

6.7 Zusammenfassung

Ziel der Evaluation ist das Testen des Rahmenwerks im Sinne eines Systemtests und die Vermittlung eines groben Eindrucks der Messergebnisse der ausgewählten Fitnesstracker. Mit der Tinnitus Research Initiative wurde ein Szenario entwickelt, das die Anforderungen der Evaluation erfüllt. Dieses Szenario beinhaltet 3 Messungen pro Testperson mit jeweils zwei Fitnesstrackern (ein Gerät pro Arm) und einen vorherigen und anschließenden Fragebogen. Es nahmen 20 Testpersonen an der Evaluation teil.

Die Korrelationen der Messungen zwischen dem Alpha Mio 2 und dem Alpha Mio 1 sind zusammengefasst wie folgt:

- **“Baseline“ Messung:** 0,799
- **“Gehen“ Messung:** 0,49
- **“Gehen+Fragebogen“ Messung:** 0,69

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die Zielsetzung dieser Arbeit entstand durch die Anforderungen des Track Your Tinnitus Systems. Das Track Your Tinnitus System beinhaltet eine Android Applikation. Eine der noch nicht implementierten Anforderungen dieser Applikation ist das Auslesen von Herzfrequenzdaten der Patienten beziehungsweise deren Fitnesstrackern, falls diese entsprechende, benötigte Funktionen bereitstellen. Das Ziel dieser Arbeit war das Design und die Implementierung eines entsprechenden flexiblen Rahmenwerks zur Auslesung der Herzrate von Fitnesstrackern. Es wurden zwei Verbindungsprotokolle implementiert: Bluetooth Low Energy und ANT+. Voraussetzung ist, dass die Fitnesstracker die Herzfrequenz nicht verschlüsselt und dem standard folgend über das jeweilige Protokoll senden. Als Testgeräte dienten der Mio Alpha 1, der Mio Alpha 2 und der Garmin vivosmart HR+ und 8 Android Smartphones mit 3 verschiedenen Haupt-Versionen des Betriebssystems. Die Architektur wurde anhand der Anforderungsanalyse modular gestaltet, sodass die Implementierung generalisiert werden und das Rahmenwerk einfach erweiterbar ist. Nachdem das Rahmenwerk in den Grundfunktionen stabil implementiert wurde, fand ein Abgleich der Anforderungen statt, der positiv ausfiel. Anschließend wurde das Rahmenwerk in einem ausgiebigen Test mit 20 Testpersonen evaluiert und die Ergebnisse zur detaillierten Bewertung präsentiert.

7.2 Ausblick

In erster Linie diente die Entwicklung des Rahmenwerks der Erweiterung der Track Your Tinnitus Android Applikation. Diese Arbeit erläutert die Nutzung des Rahmenwerks und ermöglicht diesen Schritt. Desweiteren wäre es möglich das Rahmenwerk um ein Verbindungsprotokoll oder anders funktionierenden Tracker zu erweitern. Das nicht deterministische Verhalten des Mio Alpha 2, beschrieben in Kapitel 6.5, sollte genauer (gegebenen-

falls auf der Transportschicht) untersucht werden. Da der Garmin vivosmart HR+ Tracker nur während der Entwicklung getestet wurde, ist es auch angebracht für diesen Typ Tracker eine entsprechende Evaluation durchzuführen. Eine weitere interessante Evaluation wäre der Vergleich der Messung mit einem geeichten, medizinischen Messgerät.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sinngemäße Übernahmen aus anderen Werken sind als solche kenntlich gemacht und mit genauer Quellenangabe (auch aus elektronischen Medien) versehen.

Ulm, den 22.01.2018 Florian Malsam

Anhang A

Anhang

Fragebogen Pulsmessung

Fragen zur Person

Datum: _____ Ort: _____

Name: _____ Email: _____

Geschlecht: ☐ m ☐ w Alter: ____

Jahre der Ausbildung ab Grundschule: __

Smartphone: _____

Wie schätzen Sie Ihren Puls ein? _____ Schläge pro Minute.

Messung bitte jetzt durchführen.
Danach weiter im Fragebogen.

Bitte beantworten Sie die folgenden Fragen:

Die Pulsmessung war aufwändig für mich.

☐ stimme zu ☐ neutral ☐ stimme nicht zu

Ich setze regelmäßig einen Fitnesstracker ein.

☐ stimme zu ☐ neutral ☐ stimme nicht zu

Ich messe regelmäßig meinen Puls, technisch oder manuell.

☐ stimme zu ☐ neutral ☐ stimme nicht zu

Abbildung A.1: Fragebogen der Evaluation

Literaturverzeichnis

[1] *Android 6.0 changes*. : *Android 6.0 changes*, https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html#behavior-hardware-id

[2] *ANT Message Protocol and Usage*. : *ANT Message Protocol and Usage*. Rev 5.1

[3] *Bluetooth Webseite*. : *Bluetooth Webseite*, www.bluetooth.com

[4] *Creating ANT+ Android Applications*. : *Creating ANT+ Android Applications*, www.thisisant.com

[5] *Fitbit Webseite*. : *Fitbit Webseite*, https://www.fitbit.com/

[6] *Garmin vivosmart HR+ manual*. : *Garmin vivosmart HR+ manual*, http://www8.garmin.com/manuals/webhelp/vivosmarthr/EN-US/

[7] *Garmin Webseite*. : *Garmin Webseite*, https://www.garmin.com/

[8] *Heart Rate Service*. : *Heart Rate Service*, https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.service.heart_rate.xml

[9] *How Bluetooth Works*. : *How Bluetooth Works*, https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-p2p

[10] *IEEE 802.15 Working Group for Wireless Specialty Networks*. : *IEEE 802.15 Working Group for Wireless Specialty Networks*, http://grouper.ieee.org/groups/802/15/

[11] *Mio 1 Manual*. : *Mio 1 Manual*, https://www.mioglobal.com/docs/mio_alpha_userguide_eng.pdf

[12] *Mio 2 Manual*. : *Mio 2 Manual*, https://www.mioglobal.com/docs/mio_alpha2_complete-user-guide_en.pdf

[13] *PowerStream Li-ion Coin Cell Lir2032 Data Sheet*. : *PowerStream Li-ion Coin Cell Lir2032 Data Sheet*

[14] *Android Source*. : *Android Source*, Nov. 2017. https://source.android.com/setup/

[15] *Strategy Analytics: Android Captures Record 88 Percent Share of Global Smartphone Shipments in Q3 2016*. : *Strategy Analytics: Android Captures Record 88 Percent Share of Global Smartphone Shipments in Q3 2016*, Nov. 2017. https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2016/11/02/#.WmHB-Kjia70

[16] ABRAN, A. ; BOURQUE, P. ; DUPUIS, R. ; MOORE, J. W.: *Guide to the software engineering body of knowledge-SWEBOK*. IEEE Press, 2001

[17] BATRA, J. P. J. B. M.: NIST Special Publication 800-121. In: *csrc.nist.gov* (2017)

[18] DBIS: DBIS Universität Ulm. In: *Homepage der Universität Ulm* (2017). https://www.uni-ulm.de/in/iui-dbis/startseite/]

[19] DBIS: Track Your Tinnitus DBIS Seite. In: *DBIS* (2017). https://www.uni-ulm.de/in/iui-dbis/forschung/laufende-projekte/trackyourtinnitus/

[20] DÜKING, P. ; HOTHO, A. ; HOLMBERG, H.-C. ; FUSS, F. K. ; SPERLICH, B. : Comparison of Non-Invasive Individual Monitoring of the Training and Health of Athletes with Commercially Available Wearable Technologies. In: *Frontiers in Physiology* 7 (2016), S. 71. – ISSN 1664–042X

[21] DYNASTREAM: ANT history. In: *ANT Webseite* (2017). https://www.thisisant.com/company/d1/history/

[22] FARAGHER, H. : An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications. In: *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation* (2014)

[23] GERLACH, H.-E. : *Praktische Phlebologie - Empfehlungen zur differenzierten Diagnostik und Therapie phlebologischer Krankheitsbilder ; 69 Tabellen*. Stuttgart : Georg Thieme Verlag, 2006. – ISBN 978–3–131–19232–5

[24] GORELICK, D. M.: Mio Heart Rate Accuracy vs EKG. In: *Mio Heart Rate Accuracy Study Brief* (2013)

[25] HERRMANN, J. : *Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten*, Universtität Ulm, Diplomarbeit, 2014

[26] KARGL: Mobile Communication and Bluetooth. In: *Advanced Concepts of Computer Networks* (2015)

[27] KHSSIBI, B. V. S. Idoudi: Presentation and analysis of a new technology for low-power wireless sensor network. In: *International Journal of Digital Information and Wireless Communications* (2013)

[28] MAAS, R. : Fitness-Tracker und Datenschutz / Bitkom. 2016. – Forschungsbericht

[29] PARAK, J. ; KORHONEN, I. : Evaluation of wearable consumer heart rate monitors based on photopletysmography. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014. – ISSN 1094–687X, S. 3670–3673

[30] PROBST, T. ; PRYSS, R. ; LANGGUTH, B. ; RAUSCHECKER, J. ; SCHOBEL, J. ; REICHERT, M. ; SPILIOPOULOU, M. ; SCHLEE, W. ; ZIMMERMANN, J. : Does tinnitus depend on time-of-day? An ecological momentary assessment study with the "TrackYourTinnitus" application. In: *Frontiers in Aging Neuroscience* 9 (2017), S. 253–253

[31] PROBST, T. ; PRYSS, R. ; LANGGUTH, B. ; SCHLEE, W. : Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the "TrackYourTinnitus" application. In: *Scientific Reports* 6 (2016), February

[32] PROBST, T. ; PRYSS, R. ; LANGGUTH, B. ; SPILIOPOULOU, M. ; LANDGREBE, M. ; VESALA, M. ; HARRISON, S. ; SCHOBEL, J. ; REICHERT, M. ; STACH, M. ; SCHLEE, W. : Outpatient Tinnitus Clinic, Self-Help Web Platform, or Mobile Application to Recruit Tinnitus Study Samples? In: *Frontiers in Aging Neuroscience* 9 (2017), April, S. 113–113

[33] PRYSS, R. ; PROBST, T. ; SCHLEE, W. ; SCHOBEL, J. ; LANGGUTH, B. ; NEFF, P. ; SPILIOPOULOU, M. ; REICHERT, M. : Mobile Crowdsensing for the Juxtaposition of Realtime Assessments and Retrospective Reporting for Neuropsychiatric Symptoms. In: *30th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2017)*, IEEE Computer Society Press, June 2017

[34] PRYSS, R. ; SCHLEE, W. ; LANGGUTH, B. ; REICHERT, M. : Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback. In: *6th IEEE International Conference on AI & Mobile Services (IEEE AIMS 2017)*, IEEE Computer Society Press, June 2017

[35] R, P. ; M, R. ; B, L. ; W, S. : Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research. In: *Mobile Services (MS), 2015 IEEE International Conference* (2014)

[36] SCHLEE, W. ; PRYSS, R. ; PROBST, T. ; SCHOBEL, J. ; BACHMEIER, A. ; REICHERT, M. ; LANGGUTH, B. : Measuring the Moment-to-Moment Variability of Tinnitus: The TrackYourTinnitus Smart Phone App. In: *Frontiers in Aging Neuroscience* 8 (2016), December, S. 294–294

[37] SHCHERBINA, A. ; MATTSSON, C. M. ; WAGGOTT, D. ; SALISBURY, H. ; CHRISTLE, J. W. ; HASTIE, T. ; WHEELER, M. T. ; ASHLEY, E. A.: Accuracy in Wrist-Worn, Sensor-Based Measurements of Heart Rate and Energy Expenditure in a Diverse Cohort. In:

Journal of Personalized Medicine (2017)

[38] STAHL, S. E. ; AN, H.-S. ; DINKEL, D. M. ; NOBLE, J. M. ; LEE, J.-M. : How accurate are the wrist-based heart rate monitors during walking and running activities? Are they accurate enough? In: *BMJ Open Sport & Exercise Medicine* 2 (2016), Nr. 1

[39] T, P. ; R, P. ; B, L. ; W, S. : Emotion dynamics and tinnitus: Daily life data from the "TrackYourTinnitus" application. In: *Scientific Reports 6* (2016)

[40] TALASILA, B. Curtmola: Mobile Crowd Sensing. In: *Department of Computer Science New Jersey Institute of Technology Newark, NJ, USA* (2014)

Abbildungsverzeichnis

Tabellenverzeichnis