

FHIR in Digitalen Gesundheitsanwendungen

Automatisierte Erfassung, Ablage und Retrieval von
Vitaldaten auf einem FHIR-Testserver

FHIR in Digital Health Applications

Automated Collection, Storage and Retrieval of Vital
Signs on a FHIR Test Server

Bachelorarbeit

zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

Autor: Marius Müller

Matrikelnummer: 968739

Studiengang: Medizintechnik (B.Sc.)

Betreuer: Prof. Dr. Stefan Benzschawel

Trier, 21.05.2021

Vorwort

Mein Dank richtet sich an alle, die mich beim Verfassen dieser Bachelorarbeit unterstützt haben. Vor allem richtet sich mein Dank an Herrn Prof.Dr.rer.nat. Benzschawel für die fachkundige und unkomplizierte Betreuung von der Themenfindung bis zur Abgabe der Arbeit. Zudem gilt Dank an das Technische Komitee FHIR HL7 Deutschland, mit deren Erlaubnis das FHIR®-Logo in die, im Rahmen dieser Abschlussarbeit erstellte Web-Applikation, integriert werden durfte. Bedanken möchte ich mich außerdem bei Méline Köster, die mir mit Rat und Tat bei der Formatierung der Arbeit zur Seite stand.

Zur besseren Lesbarkeit wird in der vorliegenden Arbeit auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Es wird das generische Maskulinum verwendet, wobei beide Geschlechter gleichermaßen gemeint sind.

Ich wünsche Ihnen viel Freude beim Lesen dieser Bachelorarbeit.

Marius Müller

Trier, 21.05.2021

Kurzfassung

Die Abschlussarbeit mit dem Titel „FHIR in Digitalen Gesundheitsanwendungen - Automatisierte Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Testserver“ befasst sich mit dem Einsatz des internationalen Datenstandards FHIR® zur Kommunikation von Vitalparametern. Dazu wird die Entwicklung von Standards und Terminologien im Gesundheitswesen mit besonderem Fokus auf FHIR wissenschaftlich erarbeitet.

Um den Einsatz von FHIR zu demonstrieren, wird prototypisch ein Software-System erstellt, mit dem die mit Consumer-Geräten aufgenommenen Vitalparameter auf einen Raspberry Pi 3B+ übertragen und dort mit Hilfe des Open-Source Kommunikationsservers Mirth Connect ins FHIR-Format konvertiert werden können. Die so erstellten FHIR-Ressourcen werden auf einen FHIR-Testserver geladen. Mit der im Rahmen dieser Arbeit erstellten Web-Applikation „GetFHIRData“ können die Messergebnisse vom FHIR-Testserver abgegriffen und visualisiert werden.

Abstract

The Bachelor Thesis entitled “FHIR in Digital Health Applications – Automated Collection, Storage and Retrieval of Vital Signs on a FHIR Test Server” deals with the use of the international data standard FHIR® for the communication of vital parameters. To this end, the development of standards and terminologies in the healthcare sector with a particular focus on FHIR is scientifically elaborated.

In order to demonstrate the use of FHIR, a prototype software system will be created with which the vital parameters recorded with consumer devices can be transferred to a Raspberry Pi 3B+ and converted to FHIR format using the open-source communication server Mirth Connect. The FHIR resources created in this way are loaded to a FHIR test server. With the web-application “GetFHIRData” created as part of this work, the measurement results can be accessed and visualized from the FHIR test server.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	1
1.2	Zielsetzung	3
1.3	Inhaltlicher Aufbau der Arbeit	4
2	Related Work	5
3	Grundlagen	7
3.1	Health Level 7 (HL7)	7
3.2	Entwicklung von Standards und Terminologien im Gesundheitswesen	9
4	FHIR – Fast Healthcare Interoperability Resources	16
4.1	Zusammensetzung des FHIR-Standards	17
4.2	Aufbau einer FHIR-Ressource	18
4.2.1	Profiling	20
4.3	Kommunikation von FHIR-Ressourcen	21
4.4	Anwendung von FHIR	24
4.4.1	Medizinische Informationsobjekte (MIOs)	25
4.4.2	Entwicklung von MIOs	27
4.4.3	Digitale Gesundheitsanwendungen (DiGA)	29
5	Konzept	30
5.1	Verwendete Hard- und Software	30
5.2	Kommunikationsserver Mirth Connect	34
5.3	Entwurf des Gesamtsystems	37
5.4	Darstellung der Vitalparameter als FHIR-Ressource	39
6	Realisierung	41
6.1	Durchführung der Messung	41
6.2	Kopieren der Messdaten auf den Raspberry Pi	44
6.3	Konvertieren der Messdaten	46
6.4	Upload der Daten auf FHIR-Testserver mit Mirth Connect	48
6.4.1	Einrichten des Channels	49
6.4.2	Einrichten der Filter und Transformer	51

7	Visualisierung	53
7.1	Das Login-System.....	53
7.2	Darstellung der Messergebnisse	55
8	Fazit.....	57
8.1	Zusammenfassung der Ergebnisse	57
8.2	Ausblick.....	59
Quellen	VIII
Index.....		VIII
Anhang.....		VIII

Abbildungsverzeichnis

Abbildung 1: Chronologische Entwicklung von Standards und Terminologien im Gesundheitswesen	9
Abbildung 2: Referenz-Informations-Modell (RIM)	14
Abbildung 3: Beispielhafte FHIR Ressource des Typs „Patient“	19
Abbildung 4: KBV-Basis-Profil: PatientIn	26
Abbildung 5: Withings BPM Core: Blood Pressure Monitor	31
Abbildung 6: Withings Body+	31
Abbildung 7: Withings Thermo	32
Abbildung 8: Samsung Galaxy J5	32
Abbildung 9: Raspberry Pi 3B+	33
Abbildung 10: Funktionsweise eines Mirth Connect Channels.....	36
Abbildung 11: Grafische Darstellung des umzusetzenden Konzepts.....	38
Abbildung 12: Observation Ressource - Body Temperature.....	40
Abbildung 13: Übersichtsseite der App „Health Mate“	41
Abbildung 14: Ausschnitt der Tabelle „measure“ aus der Datenbank „WithingsWiScale“	43
Abbildung 15: Visualisierung des Einsatzes der verwendeten Hardware.....	44
Abbildung 16: Terminal-Ausgabe des Raspberry Pis (adb devices/root).....	45
Abbildung 17: Terminal-Ausgabe des Raspberry Pis (adb chmod/pull)	45
Abbildung 18: Beispielhafter Ausschnitt des Ordners „XML_Output“	46
Abbildung 19: Beispielhafte XML-Datei mit den Messergebnissen einer Temperaturmessung	47
Abbildung 20: Beispielhafte XML-Datei mit den Ergebnissen einer Blutdruckmessung	47
Abbildung 21: Admin Launcher des Kommunikationsservers Mirth Connect.....	48
Abbildung 22: Source Connector des Mirth Connect Channels „XML_to_FHIR“	49
Abbildung 23: Destination Connector des Mirth Connect Channels „XML_to_FHIR“	50
Abbildung 24: Message Templates für den Mirth-Connect Channel „XML_to_FHIR“	50
Abbildung 25: Destination-Filter des Mirth-Connect Channels „XML_to_FHIR“	51
Abbildung 26: 1. Destination-Transformer (Message Builder) des Mirth Connect Channels „XML_to_FHIR“	51

Abbildung 27: 2. Destination-Transformer (JavaScript) des Mirth Connect Channels „XML_to_FHIR“	52
Abbildung 28: 3. Destination-Transformer (JavaScript) des Mirth Connect Channels „XML_to_FHIR“	52
Abbildung 29: Abbildung der Filter Optionen, die dem Benutzer der Web-Applikation „GetFHIRData“ zur Verfügung stehen	56

Tabellenverzeichnis

Tabelle 1: Die wichtigsten Nachrichtentypen des HL7 v2 Standards.....	10
Tabelle 2: Komponenten eines LOINC-Eintrags	12
Tabelle 3: Interaktion auf Instanz-Ebene	23
Tabelle 4: Interaktion auf Typ-Ebene	23
Tabelle 5: Interaktion auf Gesamtes System	23
Tabelle 6: Source Connectoren des Kommunikationsservers Mirth Connect	34
Tabelle 7: Destination Connectoren des Kommunikationsservers Mirth Connect	35
Tabelle 8: Für dieses Projekt relevante Messtypen der Withings-Datenbank.....	42

1 Einleitung

„Per App ins digitale Krankenhaus, [1]“

schreibt die Lausitzer Rundschau über das Carl-Thiem-Klinikum in Cottbus, das seit Anfang des Jahres 2020 einen digitalen Express Check-In anbietet, mit dem sich Patienten schnell und bequem „selbst aufnehmen“ können [57]. Was vor einigen Jahren noch als Science-Fiction abgetan wurde, könnte bald Realität werden. Projekte wie diese sind heutzutage keine Seltenheit mehr – spätestens seit der Einführung der elektronischen Patientenakte (ePA) für gesetzlich Versicherte der Bundesrepublik Deutschland am 1. Januar 2021 [40], nimmt die Gesundheits-IT eine zentrale Rolle in der Gesundheitsversorgung und der Gesellschaft ein. Mit dem Inkrafttreten des Digitale-Versorgungs-Gesetzes (DVG) am 19. Dezember 2019 wurden der Bevölkerung zudem „Apps auf Rezept“, also kleine digitale Helfer zur Erkennung und Behandlung von Krankheiten, an die Hand gegeben [47].

Mit jedem Schritt in Richtung „digitales Krankenhaus“ wird die Menge an Informationen, die aufgenommen, verarbeitet und ausgetauscht werden muss, größer und größer. Kein einzelnes Informationssystem kann diese Fülle an Daten verarbeiten und gleichzeitig die Belange in allen Bereichen des Gesundheitssektors abdecken. Aus diesem Grund muss die elektronische Datenverarbeitung auf verschiedene Anbieter und Systeme verteilt werden. Damit das so zusammengesetzte Gesamtsystem einwandfrei funktioniert, müssen alle Einzelsysteme funktional gekoppelt sein und die Software jedes Anbieters muss problemlos in die Kommunikationslandschaft integriert werden können. Das Ziel ist es also, dass bei der Einführung eines neuen Systems, die Kommunikation mit den bereits vorhandenen Systemen schlüsselfertig vorliegt und sofort mit dem Datenaustausch begonnen werden kann. [6]

1.1 Aufgabenstellung

Um die notwendigen Schnittstellen zwischen verschiedenen Systemen im Gesundheitswesen zu standardisieren, wurde FHIR ins Leben gerufen. FHIR steht für „Fast Healthcare Interoperability Resources“ und ist ein Standard für den Datenaustausch zwischen Softwaresystemen im Gesundheitswesen. Im Rahmen dieser Bachelorarbeit wird der Datenstandard FHIR vorgestellt und erläutert, wie und warum FHIR für die Kommunikation von Gesundheitsdaten genutzt wird. Dabei wird ebenfalls darauf eingegangen, welche Vorteile FHIR gegenüber bereits etablierten Standards bietet.

Um die Anwendung von FHIR zu demonstrieren, wird im praktischen Teil der Arbeit eine prototypische „digitale Gesundheitsanwendung“ entwickelt. Das Ziel dabei ist es, die Erfassung der Vitalparameter Blutdruck, Herzfrequenz, Körpergewicht, BMI und Körpertemperatur mit Consumer-Geräten umzusetzen. Die so erfassten Daten sollen jedoch nicht in einer proprietären Cloud, sondern auf einem frei gewählten FHIR-Server in Form einer FHIR-Ressource gespeichert werden. Aufgenommen werden die Vitaldaten mit Hilfe dreier Messgeräte der Firma Withings: eine Körperwaage, ein Blutdruckmessgerät und ein Schläfenthermometer. Die Daten sollen von einem Smartphone, das über Bluetooth mit den

Messgeräten verbunden ist, auf einen Raspberry Pi geladen werden. Dort sollen sie mit einem Kommunikationsserver in den FHIR-Standard konvertiert und anschließend auf einem FHIR-Testserver gespeichert werden. Um die Messergebnisse zu visualisieren, sollen die FHIR-Daten vom FHIR-Testserver abgefragt und anschließend in Tabellen und Diagrammen grafisch aufbereitet werden. Zusätzlich zur Dokumentation der praktischen Arbeit, soll der prototypische Ablauf einer Messung inklusive der technischen Aspekte des Abgreifens der Daten, deren Konvertierung und Ablage auf dem FHIR-Testserver, sowie das Retrieval der Daten in Form eines Kurzfilms dargestellt werden.

1.2 Zielsetzung

Diese Arbeit wird nicht mit der Intention verfasst, ein verkaufsfertiges Produkt mit den oben genannten Funktionalitäten zu entwickeln. Das Ziel der Arbeit ist es, die Grundlagen des Datenaustauschs im Gesundheitswesen, mit besonderem Fokus auf dem Standard FHIR, wissenschaftlich zu erarbeiten und den Einsatz dieses Standards anhand eines Real-World-Problems zu demonstrieren. Dabei wird besonderer Wert auf einen möglichst hohen Lerneffekt gelegt, weshalb versucht wird, möglichst viele Facetten der Kommunikation von Gesundheitsdaten darzustellen.

Am Ende dieser Arbeit wird der Leser verstehen, warum Standards für den Datenaustausch im Gesundheitswesen unabdingbar sind, wie Kommunikationsserver verwendet werden können, um diesen Datenaustausch zu vereinfachen und wie FHIR und FHIR-Server in Software-Projekte integriert werden können.

Zudem soll diese Abschlussarbeit als Grundlage für weiterführende Tätigkeiten im Fachbereich Informatik der Hochschule Trier dienen. Unter der Aufsicht von Herrn Professor Benzschawel soll ein hochschuleigener FHIR-Server implementiert werden, der von einem Team der Hochschule verwaltet wird und auf dem selbst aufgenommene Gesundheitsdaten abgelegt werden können. Die im Rahmen dieser Arbeit beschriebenen Funktionalitäten können dann genutzt werden, um eigenständige Projekte mit Einbindung des FHIR-Servers der Hochschule zu entwickeln, die unabhängig von externen Anbietern sind.

1.3 Inhaltlicher Aufbau der Arbeit

Die Arbeit gliedert sich in sechs Teile, wobei der erste Teil die Einleitung darstellt, in der dem Leser ein erster Einblick in die Thematik und den Aufbau der Arbeit gegeben wird.

Anschließend folgt das Kapitel „Related Work“ in dem verwandte Arbeiten vorgestellt werden, um diese Arbeit in den Kontext der aktuellen Forschung zu setzen.

Im dritten Kapitel wird die Entwicklung von Standards und Terminologien im Gesundheitswesen aufgezeigt und es wird darauf eingegangen, wie diese in FHIR integriert wurden, bzw. wie sie die Entstehung von FHIR beeinflusst haben. Daraufhin wird ein detaillierter Einblick in den Aufbau und die Funktionsweise von FHIR gegeben und der Einsatz des Standards wird anhand von Medizinischen Informationsobjekten (MIOs) der Kassenärztlichen Bundesvereinigung (KBV) und digitalen Gesundheitsanwendungen (DiGAs) des Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM) erläutert. Dabei wird besonders auf die Frage eingegangen, warum FHIR für die Umsetzung der MIOs und DiGAs genutzt wird. Nachdem dieser Komplex beschrieben wurde, wird in den praktischen Teil der Abschlussarbeit übergeleitet.

Hier wird zu Beginn das Konzept, das im Rahmen dieser Arbeit umgesetzt werden soll, präsentiert. Dazu wird die verwendete Hard- und Software vorgestellt und die Rahmenbedingungen für die Umsetzung werden definiert.

Im darauffolgenden Kapitel wird detailliert dargestellt, wie das zuvor beschriebene Konzept realisiert wurde. Hier wird auf die technischen Einzelheiten in Bezug auf die Beschaffung und Konvertierung der Daten ins FHIR Format und den Upload auf den FHIR-Testserver eingegangen. An dieser Stelle sei darauf hingewiesen, dass Leser, die nicht an den technischen Details interessiert sind, das Kapitel „Realisierung“ ohne weiteres überspringen können. Da zusätzlich zur Abschlussarbeit ein Kurzfilm gedreht wurde, in dem alle Funktionalitäten dargestellt werden, bekommt der Leser auch ohne detailliertes Wissen über die technische Umsetzung, einen guten Überblick über die Funktionsweise der im Projekt implementierten Komponenten.

Anschließend folgt das Kapitel „Visualisierung“, in dem erläutert wird, wie die Vital-Parameter grafisch aufbereitet werden können, nachdem die Messergebnisse auf dem FHIR-Server abgelegt wurden.

Im letzten Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben, wie die hier dargestellten Funktionalitäten erweitert und in zukünftigen Projekten integriert werden können.

2 Related Work

Vitalparameter gehören zu den wichtigsten allgemeinen Beobachtungskriterien, da sie Veränderung im physischen und psychischen Befinden des Menschen signalisieren können und so Rückschlüsse auf bestimmte Krankheiten zulassen [63]. Menschen, die an chronischen Krankheiten leiden, müssen ihre Vitalparameter wie Blutdruck, Herzfrequenz, Blutzucker und Gewicht regelmäßig messen, da sie eine wichtige Ergänzung der schulmedizinischen Behandlung geworden sind, um den Fortschritt einer Therapie zu sichern [67]. Auch im Bereich des „Ambient Assisted Living“ nimmt die Vitalparametererfassung eine zentrale Rolle ein und soll den Anwendern zu einer Steigerung der Lebensqualität und einem längeren selbständigen Leben im häuslichen Umfeld verhelfen [69].

Viele der bisher verfügbaren Lösungen zur Erfassung von Vitaldaten stellen die Messergebnisse jedoch nur der Person zur Verfügung, die die Messung durchführt, was dazu führt, dass Gesundheitsdienstleister oft keinen Zugriff auf die Daten haben.

Eine Lösung für dieses Problem bietet der in [64] vorgestellte Gesundheitsdienst zur Vitalparametererfassung im täglichen Leben. Zusätzlich zu den Standardfunktionalitäten ist hier ein Clinical Decision Support System (CDSS) integriert, um dem Patienten personalisierte Berichte über seinen Gesundheitszustand anzeigen lassen zu können. Die gesundheitsbezogenen Daten der Benutzer werden auf einem Server gespeichert, auf dem Dienste zur Generierung von Echtzeit-Feedback, Empfehlungen, Erinnerungen und Alarmen implementiert sind. Diese Daten können von medizinischem Fachpersonal abgegriffen und ausgewertet werden.

Systeme wie diese sind heutzutage keine Seltenheit mehr [65][66]. Dennoch stellt sich bei der Implementierung immer die Frage wie die Daten kommuniziert werden können, damit Interoperabilität zwischen verschiedenen Systemen mit möglichst effizienter Übertragung erreicht werden kann.

In [67] wird ein System zur Beobachtung und Überwachung von Vitalparametern der *University of Applied Sciences Upper Austria* vorgestellt, bei dem die Vitaldaten mit Consumer-Geräten vom Patienten selbst aufgenommen und mit Hilfe einer eigens konzipierten App auf das Smartphone des Patienten geladen werden. Um die Daten auswerten zu können, wurden diese als HL7 ORU Nachricht (*siehe Abschnitt 3.2*) an ein Telehealth Service Center übertragen und dort unter Zuhilfenahme von entsprechenden IHE-Profilen (*siehe Abschnitt 3.2*) zu einer elektronischen Patientenakte zusammengefasst. Im Rahmen der in [67] vorgestellten Fallstudie, wurde die Schnittstelle zur Datenübertragung so erweitert, dass der HL7 FHIR-Standard für die Datenkommunikation eingesetzt werden konnte. Anschließend wurden beide Implementierungen verglichen und verschiedenen Patientengruppen zum Test bereitgestellt. Dabei konnte in einem Auswertungszeitraum von fünf Monaten festgestellt werden, dass die Menge an Daten, die an das Telehealth Service Center übertragen wurde, im Falle der Übertragung mit HL7 ORU, 3,2-mal höher waren als bei der Übertragung mit FHIR. Dies hatte eine Steigerung der Attraktivität des Gesamtsystems für die Probanden zur Folge, da die benötigte Rechenleistung der FHIR-Lösung wesentlich geringer ausfiel und somit die Akkulaufzeit der verwendeten Smartphones wesentlich weniger belastet wurde.

FHIR bietet also eine leichtgewichtige Lösung für die Kommunikation von Patientendaten, die die Performance der Übertragung positiv beeinflussen kann [67]. Um die Patientendaten jedoch sinnvoll interpretieren und jedem Patienten individuelle Informationen bereitstellen zu können, müssen neben den Vitalparametern weitere Daten über den Patienten, wie der allgemeine Gesundheitsstatus oder Vorerkrankungen, erhoben werden. Damit die Daten einwandfrei miteinander in Relation gesetzt werden können, muss eine semantische und syntaktische Interoperabilität zwischen den Daten aller Quellen gewährleistet sein. Auch hier kann FHIR Abhilfe schaffen:

In [68] wird von *El-Sappagh et al.* ein Prototyp für ein cloud-basiertes System vorgestellt, das Typ-1-Diabetes Mellitus Patienten überwacht. Das System sammelt Patientendaten aus verschiedenen Quellen, um diese zu formalisieren, zu analysieren und in das Gesamtsystem zu integrieren. Basierend auf dem so erstellten Patientenprofil, können jedem Patienten vollständig personalisierte Pflegepläne, Insulinschemata, Diäten und Übungen vorgeschlagen werden. Zusätzlich findet eine Echtzeit-Überwachung auf Basis von Vitalparametern statt, die mit drahtlosen Sensoren am Patienten aufgenommen werden. Alle aufgenommenen und gesammelten Daten werden in Form von FHIR Ressource dargestellt und auf einem zentralen FHIR-Server gespeichert.

FHIR bietet also die Möglichkeit, die notwendigen Schnittstellen zwischen verschiedenen Systemen im Gesundheitswesen zu standardisieren und ermöglicht so einen möglichst effektiven Datenaustausch.

Diese Arbeit schließt in gewisser Weise an die oben genannten wissenschaftlichen Arbeiten an, indem auch hier ein Prototyp für ein System zur automatisierten Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Server implementiert wird. Da die Kommunikation der Gesundheitsdaten mit Hilfe des Open-Source Kommunikationsservers Mirth Connect umgesetzt wird und auch die FHIR-Spezifikation frei zugänglich ist, können die hier vorgestellten Funktionalitäten ohne weiteres nachgestellt und in weiterführenden Arbeiten umgesetzt und erweitert werden.

Um verstehen zu können, wie die Standardisierung in FHIR umgesetzt und wie die Kommunikation von Gesundheitsdaten implementiert wurde, wird im folgenden Kapitel auf die Grundlagen und die Entwicklung von Datenstandards im Gesundheitswesen eingegangen

3 Grundlagen

Als in den 1970er Jahren die ersten Informationssysteme für Krankenhäuser und Arztpraxen entwickelt wurde, war jede Schnittstelle zwischen Systemen im Gesundheitswesen individuell gestaltet. Da es zu diesem Zeitpunkt keine Standards für die Kommunikation von Patientendaten und Untersuchungsergebnissen gab, war die Implementierung solcher Schnittstellen sehr aufwendig und dementsprechend kostenintensiv. [29]

Um dieses Problem zu lösen, wurden seitdem mehrere Standards und Organisationen ins Leben gerufen, die sich zur Aufgabe gemacht haben, den Datenaustausch im Gesundheitswesen zu standardisieren. Bis zum heutigen Tage wird stetig daran gearbeitet, die bestehenden Standards zu verbessern und neue Standards wie FHIR zu entwickeln.

Im Folgenden wird ein kurzer Überblick über die Entwicklung von Standards und Terminologien im Gesundheitswesen gegeben. Dabei wird besonders auf die Organisation HL7 (die FHIR erschaffen hat) und diejenigen Standards und Terminologien eingegangen, die in FHIR angewendet werden, bzw. die Entwicklung von FHIR beeinflusst haben.

3.1 Health Level 7 (HL7)

Die Organisation Health Level 7 (HL7) wurde 1987 mit dem Ziel gegründet, den Datenaustausch im Gesundheitswesen zu standardisieren. HL7 ist eine nicht-gewinnorientierte Organisation zur Entwicklung von Standards zum Austausch, zur Integration und zum Abruf von elektrischen Gesundheitsinformationen. Die Organisation wird von mehr als 1.600 Mitgliedern aus über 50 Ländern unterstützt, darunter mehr als 500 Firmenmitglieder, Gesundheitsdienstleister, staatliche Stellen, pharmazeutische Unternehmen und Beratungsunternehmen. [2]

*„The **HL7 Mission** is to provide standards that empower global health data interoperability.”*
[3]

HL7 ist eine ganze Familie von Standards, die die Mission verfolgen, die Interoperabilität von globalen Gesundheitsdaten zu ermöglichen und die notwendigen Schnittstellen zwischen verschiedenen Systemen zu standardisieren. Ergänzend zu diesen Standards, sind passende Terminologien bedeutungsvoll, die von anderen Organisationen verwaltet werden. [6] Auf diese Standards und Terminologien wird im folgenden Kapitel genauer eingegangen.

*“The **HL7 Vision** is a world in which everyone can securely access and use the right health data when and where they need it.”* [3]

HL7 verfolgt die Vision einer Welt, in der jeder Mensch an jedem Ort und zu jeder Zeit auf seine Gesundheitsdaten zugreifen kann. Gesundheitsdaten sollen über Einrichtungs-, Fachrichtungs- und Landesgrenzen hinweg kommuniziert werden können, um so ein globales Gesundheitsnetz aufbauen zu können. [4]

In Deutschland wird HL7 durch den eingetragenen Verein „HL7 Deutschland“ repräsentiert, mit dem Ziel der Förderung des Datenaustauschs und der Interoperabilität im Gesundheitswesen, der Gesundheitswirtschaft sowie der Wissenschaft und Forschung. „HL7 Deutschland“ koordiniert die Interessen der deutschen HL7-Anwender und arbeitet aktiv in europäischen und internationalen HL7-Organisationen mit, die sich mit der Anwendung, der Weiterentwicklung und der Nutzung der HL7 Standards befassen. [10]

3.2 Entwicklung von Standards und Terminologien im Gesundheitswesen

Um die Entstehung, die Funktionsweise und den Einsatz von FHIR besser verstehen zu können, wird im Folgenden ein kurzer Überblick über die Entwicklung von Standards und Terminologien im Gesundheitswesen gegeben. Dabei wird ebenfalls auf die Klassifikationssysteme ICD und SNOMED eingegangen, die beide im Zusammenhang mit FHIR verwendet werden. In Abbildung 1 sind die hier behandelten Standards chronologisch dargestellt.

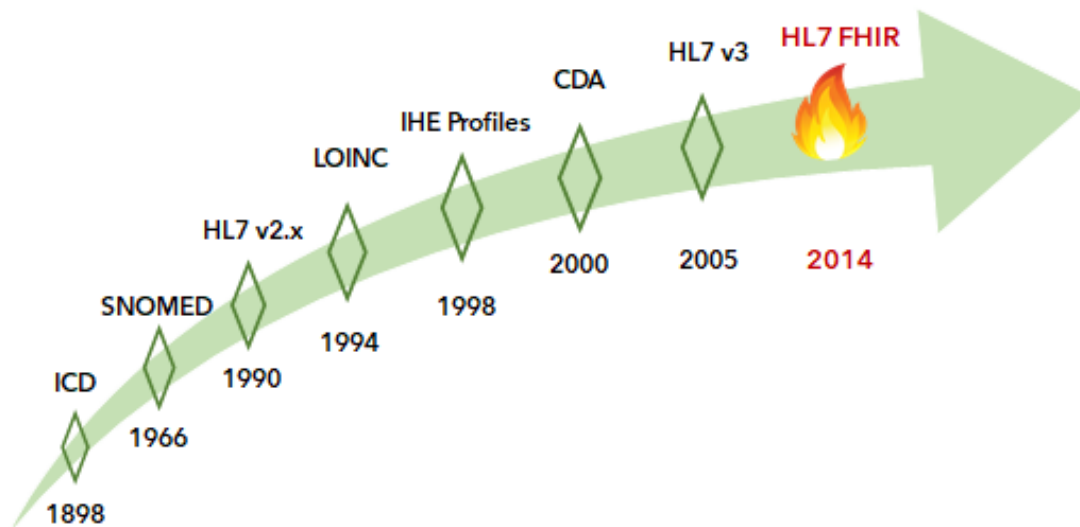


Abbildung 1: Chronologische Entwicklung von Standards und Terminologien im Gesundheitswesen

Quelle: In Anlehnung an den Vortrag „SMART on FHIR“ von Renee Cannon – Senior Sales Engineer, Intersystems(2018)

Digitale Aufzeichnung: <https://www.youtube.com/watch?v=MMfe181tmwU&t=1705s>

ICD – Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme

Die Ursprünge der Klassifikationssysteme gehen auf die 1850er Jahre zurück, in denen eine einheitliche, internationale Nomenklatur für Todesursachen angestrebt wurde. Im Jahr 1899 wurde im Zuge der Tagung des internationalen Statistischen Instituts in Oslo ein erster Entwurf vorgelegt, der in den folgenden Jahren in 10 Revisionskonferenzen überarbeitet und verbessert wurde. Die aktuelle Fassung ICD-10 wird von der Weltgesundheitsorganisation (WHO) verwaltet. Am 01. Januar 2022 tritt die zukünftige Version ICD-11 in Kraft, die bereits von der Weltgesundheitsversammlung verabschiedet wurde. [13]

Im ICD-10 Klassifikationssystem ist jeder Erkrankung weltweit ein eindeutiger Diagnoseschlüssel zugeordnet, sodass internationale Gesundheitsstatistiken systematisch miteinander verglichen werden können. ICD-10 besteht aus 22 Kapiteln, die thematisch den Bereich aller Krankheiten abdecken. In Deutschland sind Ärzte dazu verpflichtet ihre Diagnosen nach ICD-10-GM (German Modification) zu verschlüsseln. [13][14]

SNOMED – Systematisierte Nomenklatur der Medizin

SNOMED ist eine Familie medizinischer Terminologie-Systeme, wobei die erste Version auf das Jahr 1974 zurückgeht. Die aktuelle Version SNOMED CT ist derzeit eine der umfassendsten und wichtigsten medizinischen Terminologien weltweit und enthält über 300.000 Begriffe, die eindeutig identifiziert, logisch definiert und hierarchisch angeordnet sind. [15]

Im Gegensatz zu ICD-10 klassifiziert SNOMED CT nicht nur Diagnosen, sondern bildet medizinische Fachausdrücke ab und setzt diese zueinander in Beziehung, indem Symptomen, Befunden und Anwendungen weltweit gültige Codes zugeordnet werden. Auf diese Weise wird eine einheitliche Sprache geschaffen, die eine semantische Interoperabilität schafft, mit der medizinische Daten einrichtungs- und sektorübergreifend kommuniziert werden können. [15] [16]

SNOMED CT genießt weltweit ein sehr hohes Ansehen und wird seit Beginn 2021 vom Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM) auch in Deutschland zur Verfügung gestellt. Alle im deutschen Gesundheitswesen beteiligten Institutionen, Fachkreise, Softwarehersteller und Kliniken können beim BfArM eine kostenfreie Lizenz und ebenfalls kostenlose Schulungen beantragen, um einen reibungslosen Start der Implementierung zu gewährleisten. Auf diese Weise soll die Digitalisierung des deutschen Gesundheitswesens weiter vorangetrieben werden. [17]

HL7 v2 – Health Level 7 Version 2

HL7 v2 ist der weltweit meistgenutzte Standard für den Informationsaustausch im Gesundheitswesen und dient der Systemintegration innerhalb von Krankenhäusern. Nahezu 100% aller deutschen Krankenhäuser nutzen HL7 Version 2 für die interne Kommunikation, da er die wichtigsten Anwendungsgebiete im Krankenhaus wie Patienten-Administration, Befundkommunikation, Leistungsanforderung und -übermittlung, Dokumenten- und Stammdatenaustausch und Mitarbeiterdaten abdeckt. [7]

Im HL7 v2 Standard sind verschiedene Nachrichtentypen definiert, die in mehrere Segmente und Felder gegliedert sind. In Tabelle 1 folgt eine Auflistung der wichtigsten Nachrichtentypen:

Tabelle 1: Die wichtigsten Nachrichtentypen des HL7 v2 Standards

Nachrichtentyp	Bedeutung	Deutsche Übersetzung
ADT	A dmission, D ischarge, T ransfer	Aufnahme, Verlegung, Entlassung
ORM	O rders M essage	Anforderung einer Untersuchung
ORR	O rders R esponse	Antwort auf ORM Nachricht
ORU	O bservation R esult U nsolicited	Befundübermittlung
MDM	M edical D ocument M anagement	Übermittlung med. Dokumente
DFT	D etailed F inancial T ransactions	Übermittlung von Leistungsdaten zur Abrechnung
BAR	B illing A ccount R ecord	Übermittlung von Leistungsdaten nach ICD-10 Standard

Das Austauschformat für HL7 Version 2 ist einfach und textbasiert, jedoch benötigt man zum Verarbeiten der Informationen einen Parser. Die Verarbeitung von HL7-Nachrichten, Segmenten und Feldern ist somit applikationsspezifisch und muss vom Entwickler der jeweiligen Anwendung implementiert werden. [7]

LOINC – Logical Observation Identifiers Names and Codes

LOINC steht für *Logical Observation Identifiers Names and Codes* und ist der internationale Standard zur Kennzeichnung von Laborwerten, Vitalwerten und weiteren klinischen Messungen und Beobachtungen. Er wurde 1994 vom LOINC-Komitee am Regenstrief Institute in Indianapolis (USA) erstellt und wird seitdem in Zusammenarbeit mit Vertretern aus der Forschung und der US-Regierung weiterentwickelt und verbessert. LOINC wird insbesondere von HL7 als Kodiersystem zur Ergebnisübermittlung empfohlen. [21]

Das Regenstrief Institut stellt außerdem die Datenbank RELMA (Regenstrief LOINC Mapping Assistant) zur Verfügung, die neben den englischsprachigen LOINC-Tabellen, viele in weitere Sprachen übersetzte LOINC-Bezeichner enthält – darunter ca. 11.000 qualitätsgesicherte, deutschsprachige Bezeichnungen. In Deutschland wird LOINC vom BfArM verwaltet, welches die Funktion der zentralen Datenhaltung übernimmt und den Funktionsaustausch mit den zuständigen internationalen Institutionen und Projektgruppen koordiniert. [22]

Informationssysteme im Gesundheitswesen verwenden überwiegend proprietäre Systeme, um Daten zu kommunizieren, weshalb es besonders bei einrichtungsübergreifendem Datenaustausch aufwendig und fehleranfällig ist, die Daten vollständig und richtig ein- und zuzuordnen. Ziel von LOINC ist es, mit standardisierten Bezeichnungen den Austausch und das Zusammenführen von Untersuchungsergebnissen zu fördern und mit weltweit einheitlichen Identifikatoren für jede Art der Bestimmung, eine problemlose Kommunikation zu ermöglichen. [22]

Die LOINC-Datenbank besteht aus zwei Teilen: Den *Laboruntersuchungen* mit den gängigen Kategorien der klinischen Chemie, Hämatologie, Serologie und Toxikologie und den *klinischen Untersuchungen* mit Einträgen zu Vitalzeichen, hämodynamischen Bestimmungen, EKG und vielen weiteren klinischen Untersuchungen. Jeder dieser Untersuchungen wird ein LOINC-Code zugewiesen, der als eindeutiger Identifikator dient und aus einer mehrstelligen Nummer und einer einstelligen Kontrollziffer besteht. [20]

Als Beispiel hierfür wird im Folgenden die Messung des systolischen Blutdrucks betrachtet:

8480 – 6 : *Systolic blood pressure*

Es ist zu beachten, dass der LOINC-Code keinerlei Informationen über die Messung wiedergibt, sondern lediglich ein eindeutiger Identifikator des eigentlichen LOINC-Eintrags darstellt. Der LOINC-Eintrag wird durch ein 6-achsiges Klassifizierungssystem mit den folgenden Komponenten beschrieben: [20]

Tabelle 2: Komponenten eines LOINC-Eintrags

Component	Substanz oder Entität, die gemessen, bewertet oder beobachtet wird
Property	Merkmal oder Attribut, das gemessen, bewertet oder beobachtet wird
Time	Zeitintervall, in dem die Beobachtung oder Messung durchgeführt wurde
System	Kontext, in dem die Messung oder Beobachtung durchgeführt wurde
Scale	Skalentyp, in dem Ergebnis angegeben wird
Method	Optional; Besondere Art der Messmethode kann angegeben werden, falls sie vom Standard abweicht

Der vollständig spezifizierte LOINC-Name setzt sich folgendermaßen zusammen:

< component >: < property >: < time >: < system >: < scale >: < method >

Im Falle der Messung des systolischen Blutdrucks mit dem LOINC-Code 8480-6 ergibt sich der LOINC-Name zu:

Intravascular systolic: Pres: Pt: Arterial system: Qn

(Pres: Pressure; Pt: point in time; Qn: Quantitative)

IHE – Integrating the Healthcare Enterprise

IHE steht für *Integrating the Healthcare Enterprise* und ist eine 1998 gegründete Initiative zur Verbesserung und Standardisierung des Informationsaustauschs im Gesundheitswesen. IHE unterstützt die koordinierte Nutzung bereits etablierter Standards wie HL7 und versucht mit Hilfe von Profilen und Frameworks einen Großteil der spezifischen medizinischen Anforderungen abzudecken und so eine optimale Patientenversorgung zu ermöglichen. [23] Auf der offiziellen IHE-Homepage heißt es: „Systeme, die in Übereinstimmung mit IHE entwickelt wurden, kommunizieren besser miteinander, sind einfacher zu implementieren und ermöglichen es den Leistungserbringern, Informationen effektiver zu nutzen.“[23]

In den IHE-Profilen werden die relevanten Standards identifiziert und technische Leitfäden entwickelt. Sie stellen so für den Anwender eine einfache und zuverlässige Möglichkeit dar, ein ausreichendes Maß an Standardkonformität und eine effiziente Interoperabilität zu erreichen. Die Profile bieten Entwicklern einen klaren Implementierungspfad für Kommunikationsstandards und geben ihnen ein Werkzeug an die Hand, welches die Komplexität, die Kosten und das Risiko bei der Implementierung interoperabler Systeme reduziert. [23]

IHE ist nach klinischen und operativen Bereichen (Domänen) organisiert. Diese Domänen erstellen technische Frameworks, die jedes Jahr von den Komitees der entsprechenden Domäne überprüft, mit Ergänzungen erweitert und neu veröffentlicht werden. Aus den Frameworks werden die IHE-Profile abgeleitet, die zunächst zur öffentlichen Kommentierung und anschließend zu Zwecken der Testimplementierung veröffentlicht werden. Wenn alle Kriterien für eine erfolgreiche Implementierung erfüllt sind, wird das Profil als endgültiger Text veröffentlicht und in das technische Framework der Domäne aufgenommen. [23]

Ein Beispiel für ein IHE-Profil ist das „Cross-Enterprise Document Sharing“ Profil (XDS.b), in dem die Komponenten beschrieben werden, die zum Austausch von medizinischen Patientendokumenten notwendig sind. Hier werden Systeme definiert, die die Dokumente erzeugen („Document Sources“), Systeme, die diese Dokumente speichern („Repositories“) und eine Datenbank („Registry“), die die Metadaten über die gespeicherten Dokumente sammelt und verwaltet. [24]

CDA – Clinical Document Architecture

Bei der Dokumentation im Gesundheitswesen ist es wichtig, dass der Kontext dauerhaft erhalten bleibt, da er wichtige medizinische Zusammenhänge zwischen Einzelinformationen aufzeigt. Daher ist man im Gesundheitswesen daran gewöhnt, eine Dokumentenansicht von klinischen Beobachtungen zu verfassen, die reich an Text ist und den Zusammenhang des Geschehens aufzeigt. Der Inhalt der Dokumente muss dabei leicht verfügbar sein und ohne große technische Barrieren sichtbar gemacht werden können. [25]

Zu diesem Zweck wurde die *Clinical Document Architecture (CDA)* von HL7 entwickelt, welche einen Standard darstellt, der die Struktur und die Semantik von klinischen Dokumenten zum Zwecke des Austauschs spezifiziert. CDA wurde 2000 von ANSI als Standard genehmigt und ist heute ein Teil der HL7 Version 3 Familie. CDA wurde in der Extensible Markup Language (XML) implementiert und hat sich als Ziel gesetzt, den Austausch von menschenlesbaren Dokumenten zwischen Anwendern, unabhängig von deren technischen Fähigkeiten zu ermöglichen. Außerdem soll die Langlebigkeit aller kodierten Informationen und die Kompatibilität mit einer Vielzahl von Anwendungen zur Dokumentenerstellung gefördert werden. [26]

Ein CDA-Dokument ist ein definiertes und vollständiges Informationsobjekt, das Text, Bilder, Töne und andere multimediale Inhalte enthalten kann. Dabei wird gefordert, dass das Dokument menschenlesbar ist und vom Empfänger in einem Standard-Webbrowser angezeigt werden kann. Wichtig ist hierbei, dass in CDA keine Regeln zur Übertragung von Dokumenten enthalten sind und es sich somit nicht um einen Nachrichtenstandard handelt - um ein CDA-Dokument mit einem Empfänger auszutauschen, muss es mit einem anderen Standard wie HL7 V2 oder IHE XDS kombiniert werden. [26]

Ein CDA-Dokument besteht aus einem *Header* und einem *Body*. Der Header enthält die Metadaten mit Informationen zum Ersteller, zu Ort und Zeit der Erstellung und zu dem Inhalt des Dokuments, sodass das Dokument eindeutig identifiziert werden kann. Der Body enthält den eigentlichen Inhalt des Dokuments, der in menschenlesbarer Form angegeben wird. Auch wenn der Body lediglich aus einem PDF-Dokument besteht, ist der große Vorteil, dass er mit

einem Header verknüpft ist, der dem Empfänger zusätzliche, strukturierte Informationen bereitstellt. [26]

HL7 V3 – Health Level 7 Version 3

Auch wenn HL7 v2 einen ersten Schritt in Richtung Interoperabilität getan hat, müssen immer noch einige Herausforderung bewältigt werden: Bei Version 2 steht ein pragmatisches Vorgehen beim Nachrichtenaustausch im Vordergrund, sodass einfache Lösungen für die Kommunikation von Gesundheitsdaten erarbeitet werden können. Es mangelt dem HL7 v2 Standard jedoch an formalen Methoden zur Modellierung von Nachrichten und Datenelementen und an einem konsistenten Datenmodell. Außerdem fehlen klar definierte Benutzerrollen. [29]

Um diese und weitere Probleme zu lösen, wurde im Jahr 2005 HL7 v3 eingeführt, womit das Ziel verfolgt wurde, einen präzisen Standard zu implementieren, der international verwendet werden kann. Der Hauptgedanke bei der Umstellung von HL7 v2 auf HL7 v3 war es, eine modellgesteuerte Methodik zu etablieren, die alle Arbeitsabläufe im Gesundheitswesen durch Modellierung unterstützt. Dadurch sollte die Optionalität reduziert und die Wiederverwendbarkeit maximiert werden, um die Konsistenz der HL7 Nachrichten zu erhöhen. [29]

HL7 v3 dient also der umfassenden Integration aller Einrichtungen des Gesundheitswesens und stellt eine ganze Familie von Kommunikationsstandards dar. Diese Standards basieren auf XML und dem Referenz-Informations-Modell (RIM), das sich aus vier Basisklassen, zwei Verbindungsklassen und zahlreichen abgeleiteten Klassen zusammensetzt (Abb. 2). [27]

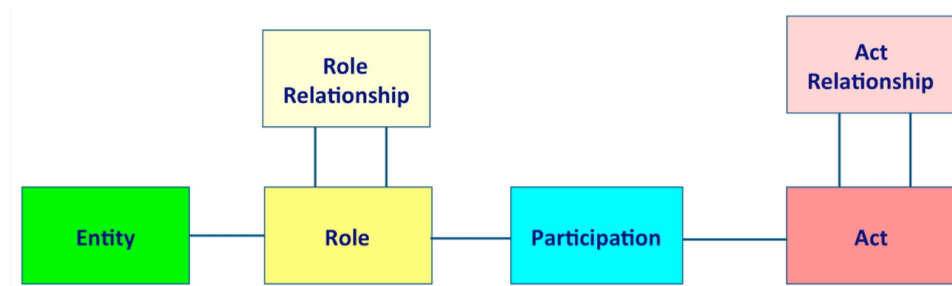


Abbildung 2: Referenz-Informations-Modell (RIM)

Quelle: <https://hl7.de/themen/hl7-v3-rim-das-referenzinformationsmodell/>

Bei den Basisklassen handelt es sich um die Entität des Bezugsobjekts (Entity), die Rolle (Role) die es einnimmt, die Partizipation (Participation), also eine nähere Beschreibung der Teilnahme an einer Aktivität und die Aktivität (Act) im Rahmen der medizinischen Untersuchung. Ergänzt werden die Basisklassen durch zwei weitere Verbindungsklassen: der Beziehung von Rollen untereinander (Role Relationship) und der Beziehung von Aktivitäten untereinander (Act Relationship). [27]

Ein stark vereinfachtes Beispielszenario zur Anwendung des Referenz-Informations-Modell ist ein stationärer Aufenthalt für eine Operation in einem Krankenhaus: Der Patient selbst stellt die Entität einer Person dar und besitzt die Rolle „Patient“. Die Aktivität, die durchgeführt werden soll, ist die Operation, an der der Chirurg als „Operateur“ teilnimmt (Partizipation). [28]

HL7 V3 hat viele Probleme von V2 behoben, aber auch viele neue Probleme geschaffen. Ein Hauptziel von V3 war es, für Konsistenz zu sorgen, jedoch mangelt es dem Standard stark an Flexibilität. Es besteht zwar die Möglichkeit V3 bis zu einem gewissen Maße anzupassen, aber dazu sind Experten notwendig, die den umfangreichen v3 Standard und das Referenz-Informations-Modell verstehen und die notwendigen Tools anwenden können. Das Ziel von v3 war es, eine Lösung zu entwickeln, die einfach und kostengünstig zu implementieren ist, jedoch ist Zahl der v3-Experten sehr klein. Die Implementierung erfordert erhebliche Ressourcen und ein außergewöhnliches Maß an Schulungen und Wissensaufbau, um erfolgreich zu sein, weshalb die Implementierung teilweise mit enormen Kosten verbunden ist. [30][31]

V3 wird von vielen IT-Experten im Gesundheitswesen als zu komplex angesehen [31]. Es wurde versucht jeden noch so kleinen Spezialfall mit dem V3-Modell zu erfassen, sodass die Anzahl und die Größe der möglichen XML-Nachrichten ein enormes Ausmaß angenommen haben. Auch wenn HL7 V3 nicht allgemein als Misserfolg gewertet werden kann, blieb die große Akzeptanz die v2 einige Jahre zuvor erfahren hatte, dennoch aus. Dem Streben nach einem einfach zu implementierenden Standard, der Interoperabilität im Gesundheitswesen ermöglicht, wurde also immer noch nicht gerecht. [30][31]

Mit diesem Ziel vor Augen, wurde im Jahr 2014 der Standard FHIR veröffentlicht, der im Folgenden detailliert beschrieben wird.

4 FHIR – Fast Healthcare Interoperability Resources

FHIR steht für „Fast Healthcare Interoperability Resources“ und ist ein von HL7 ins Leben gerufener Standard für den Datenaustausch zwischen Softwaresystemen im Gesundheitswesen. Auf der offiziellen FHIR-Homepage schreibt HL7 dazu: „FHIR kombiniert die besten Eigenschaften der HL7 v2, HL7 v3 und CDA Produktlinien, während es die neuesten Web-Standards nutzt und einen engen Fokus auf die Implementierbarkeit legt.“ [32]

FHIR wurde erstmals im Februar 2014 veröffentlicht und wird seitdem stetig weiterentwickelt. Im Rahmen dieser Arbeit wird sich auf die aktuell gültige Version v4.0.1 (Stand 10.05.2021) bezogen. Das Upgrade auf FHIR Version R5 ist bereits in Arbeit, jedoch werden dabei nur kleinere Änderungen vorgenommen, sodass die Arbeit auch in Zukunft ihre Gültigkeit behält. [33]

Ein treibender Faktor bei der Entwicklung von FHIR war der Paradigmenwechsel im Gesundheitswesen, der dazu führt, dass die Nachfrage danach, Patientendaten einrichtungs- und fachrichtungsübergreifend kommunizieren zu können, stetig steigt und der Patient selbst zunehmend Kontrolle über seine medizinischen Daten fordert. Außerdem geht der Trend von Offline zu Online, von Desktop zu Tablet und von Software zu App, weshalb FHIR agil sein und den Anwender ortsunabhängig mit seinen Daten verbinden können muss. [4]

HL7 FHIR bietet wesentliche Verbesserungen gegenüber bestehenden Standards: Es wird ein starker Fokus auf einfache Implementierbarkeit gelegt, weshalb mehrere Implementierungs-bibliotheken mit vielen Beispielen zur Verfügung gestellt werden, um eine Entwicklung schnell in Gang setzen zu können. FHIR verfolgt das Ziel, Interoperabilität „out-of-the-box“ bereitzustellen. Dazu werden Basisressourcen erzeugt, die wiederverwendet, aber auch bei Bedarf auf die lokalen Anforderungen angepasst werden können.

Bei der Entwicklung von FHIR wurde darauf geachtet, aus den Fehlern von HL7 v3 zu lernen und neue Ideologien einzuführen. FHIR kann im Gegensatz zu HL7 v3 neben bereits etablierten Standards wie HL7 v2 und CDA koexistieren und so die Vorteile aller Standards nutzen. Außerdem ist FHIR ohne Einschränkungen frei verfügbar und schließt nicht mehr aus, dass Nicht-HL7-Mitglieder den Standard anwenden. Auf diese Weise konnte sich bereits jetzt eine große Community an FHIR-Experten etablieren, die ihr Wissen mit anderen teilen und Feedback zur Anwendung in realen Einsatzszenarien geben können. [32]

4.1 Zusammensetzung des FHIR-Standards

Der Schlüsselbaustein des FHIR-Standards sind Ressourcen. Ressourcen sind kompakte, diskrete Einheiten des Datenaustauschs mit wohldefiniertem Verhalten und eindeutiger Semantik. Die insgesamt 145 spezifizierten Ressourcen sind die kleinste Einheit der Datenübermittlung und decken das gesamte Spektrum im Gesundheitswesen ab. [4].

Bei der Entwicklung wurde darauf geachtet, dass die Größe einer Ressource nicht zu umfangreich ist, da sie sonst auf Grund der komplexen Datenstruktur schwer zu handhaben wäre. Sie sollte jedoch auch nicht zu klein sein, da die Daten genügend Metainformationen und Kontext benötigen, um eine Bedeutung zu erhalten. Somit gibt es beispielsweise keinen eigenen Ressourcen-Typ für das Geschlecht, sondern den Ressourcen-Typ „Patient“, in dem das Geschlecht als Attribut angegeben wird. Gleichmaßen existiert kein Ressourcen-Typ „elektronische Patientenakte“, da dies viel zu umfangreich wäre. Stattdessen kann eine elektronische Patientenakte aus einer Kollektion von verschiedenen, spezifizierten Ressourcen zusammengefügt werden. [4]

Zur besseren Übersicht lassen sich die in FHIR implementierten Ressourcen-Typen in drei Gruppen einteilen:

1. **Administrative Ressourcen**, wie „Patient“, „Practitioner“ und „Organization“:

Hierbei handelt es sich um Ressourcen-Typen, in denen demografische und administrative Informationen über Personen, die gesundheitsbezogenen Dienstleistungen erhalten oder erbringen, festgehalten werden. Außerdem können formell oder informell anerkannte Unternehmen, Institutionen und Körperschaften erfasst werden.

2. **Klinische Resources**, wie „Medication“, „Observation“ und „FamilyMemberHistory“:

Hierbei handelt es sich um Ressourcen-Typen, die Befunde und Interpretationen von diagnostischen Tests, die an Patienten oder Patientengruppen durchgeführt wurden, enthalten. Außerdem können Informationen zur Verschreibung, Abgabe und Einnahme von Medikamenten und Gesundheitszustände von verwandten Personen, die im Zusammenhang mit der Pflege des Patienten relevant sind, erfasst werden.

3. **Infrastrukturellen Resources**, wie „CapabilityStatement“ und „StructureDefinition“:

Hierbei handelt es sich um Ressourcen-Typen, die zur Beschreibung der zugrundeliegenden Ressourcen, den in FHIR definierten Datentypen und von Erweiterungen und Einschränkungen genutzt werden können. Außerdem können Aussagen über die tatsächliche Serverfunktionalität oder erforderliche, bzw. gewünschte Serverimplementierungen dokumentiert werden.

Eine List mit detaillierten Informationen über alle momentan spezifizierten FHR-Ressourcen-Typen, lässt sich unter <https://hl7.org/fhir/resourcelist.html> finden.

4.2 Aufbau einer FHIR-Ressource

Eine FHIR-Ressource kann in den Formaten JSON, XML und Turtle dargestellt werden. Jede FHIR Ressource besteht aus vier Teilen: Der Identität und Metadaten der Ressource, der Narrative, Extensions und den Strukturierten Daten, die den eigentlichen Inhalt der Ressource repräsentieren [32]. Der Aufbau von FHIR-Ressourcen wird im folgendem anhand der Ressource „Patient“ erläutert, die in Abbildung 3 exemplarisch dargestellt ist.

1. Identität und Metadaten der Ressource

Dieser Teil wird nicht vom Erzeuger der Ressource, sondern vom FHIR-Server bereitgestellt. In den Meta-Daten können Informationen über die Versionsverwaltung und die Quelle der Ressource angegeben sein - außerdem wird der erzeugten FHIR-Ressource eine eindeutige ID zugewiesen. An dieser Stelle sei auf den Unterschied zwischen der ID und dem Identifier hingewiesen: Die ID wird vom jeweiligen FHIR-Server selbst vergeben und dient der Identifizierung der Ressource auf dem entsprechenden Server. Der Identifier hingegen ist ein Teil der strukturierten Daten des Ressourcen-Typs „Patient“ und kann vom Erzeuger der Ressource angegeben werden, um den Patienten intern zu identifizieren. [32]

2. Narrative: Menschenlesbare Zusammenfassung der Ressource

Jede Ressource kann einen menschenlesbaren Text enthalten, der dazu verwendet werden kann, den Inhalt der Ressource darzustellen. Diese Zusammenfassung muss alle Informationen enthalten, die benötigt werden, um die wesentlichen klinischen und geschäftlichen Informationen der Ressource zu verstehen. Die Narrative enthält ein Status-Element, mit dem angegeben wird, ob die Zusammenfassung vollständig aus dem Inhalt der Ressource verfasst wurde oder ob zusätzliche Informationen angegeben werden, die nicht in den strukturierten Daten der Ressource enthalten sind. Der Inhalt der Zusammenfassung wird als XHTML-Fragment angegeben. [32]

3. Extensions

Bei der Entwicklung der FHIR-Standards wurde das 80/20-Prinzip verfolgt: Attribute wurden nur als Teil der Ressource übernommen, wenn davon ausgegangen werden konnte, dass mindestens 80% aller Implementierungen, die die Ressource verwenden, auch das entsprechende Attribut benutzen [4]. Werden für gewisse Einsatzszenarien Erweiterungen außerhalb des üblichen Use-Cases benötigt, können Extensions verwendet werden. Jede Extension benötigt eine formale öffentliche Definition, auf die in der Extension selbst verwiesen werden muss [34]. Im Fall der in Abbildung 3 erzeugten Ressource des Typs *Patient*, wurde beispielhaft die Extension „*patient-birthPlace*“ verwendet. Mit Hilfe dieser Extension kann der genaue Ort angegeben werden, an dem der Patient geboren wurde, falls diese Information für ein bestimmtes Anwendungsszenario relevant ist. In dem „*URL-Element*“ wird auf die Definition der Extension verwiesen. Eine Auflistung aller bisher definierten Extensions lässt sich unter <https://www.hl7.org/fhir/extensibility-registry.html> finden.

4. Strukturierte Daten

Der eigentliche Inhalt der Resource wird in Form von strukturierten Attributen repräsentiert. Im Falle des Ressourcen-Typs *Patient* werden Informationen zum Namen, Geschlecht, Geburtsdatum und Kontaktdaten erfasst, die in fest-definierten Datentypen angegeben werden müssen. Genaue Angaben über den Aufbau aller bereits definierten Ressourcen-Typen, sind in der offiziellen FHIR-Spezifikation unter <https://hl7.org/fhir> zu finden. Dort ist auch angegeben, welche Attribute der jeweiligen Resource-Typen vorhanden sein müssen und welche optional sind. Beispielsweise ist bei einer Resource des Typs „Observation“ (die eine zentrale Rolle im weiteren Verlauf der Arbeit einnehmen wird) die Angabe eines Status-Codes verpflichtend, wohingegen die Angabe einer Interpretation der Messergebnisse optional ist.

<pre>{ "resourceType": "Patient", "id": "1933612", "meta": { "versionId": "1", "lastUpdated": "2021-03-08T19:31:10.160+00:00", "source": "#3TfFgCL7rhHYdAPc" }, }</pre>	Identität der Resource und Meta-Daten
<pre>"text": { "status": "generated", "div": "<div xmlns='http://www.w3.org/1999/xhtml'> <p>name: Mueller, Marius
identifier: 12345
gender: male
 birthDate: Sep 30, 1998</p></div>" },</pre>	Narrative
<pre>"extension": [{ "url": "http://hl7.org/fhir/StructureDefinition/patient-birthPlace", "valueAddress": { "text": "Feldstraße 16, 54290 Trier", "line": ["Feldstraße 16"], "city": "Trier", "postalCode": "54290", "country": "DEU" } },]</pre>	Extensions mit URL
<pre>"identifier": [{ "system": "http://fhirtutorial.example/patient", "value": "123456" }], "active": true, "name": [{ "family": "Müller", "given": ["Marius"] }], "telecom": [{ "system": "phone", "value": "(+49) 651 507066", "use": "home" }, { "system": "email", "value": "muelmari@hochschule-trier.de", "use": "work" }], "gender": "male", "birthDate": "1998-09-30", "address": [{ "use": "home", "text": "Hohenzollernstraße 78, 54290 Trier", "line": ["Hohenzollernstraße 78"], "city": "Trier", "postalCode": "54290", "country": "DEU" }] }</pre>	Strukturierte Daten

Abbildung 3: Beispielhafte FHIR Ressource des Typs „Patient“

Quelle: Eigene Darstellung

4.2.1 Profiling

In der FHIR-Spezifikation wird eine Reihe von Ressourcen und Frameworks definiert, jedoch sind die Anforderungen an diese von vielen Faktoren abhängig. So hat beispielsweise die Rechtslage im jeweiligen Land oder das Einsatzszenario einen großen Einfluss darauf, welche Elemente in einer Ressource enthalten sein müssen und welche nicht. FHIR versteht sich als Plattform-Spezifikation, die eine gemeinsame Plattform, bzw. Grundlage bildet, auf der eine Vielzahl unterschiedlicher Lösungen implementiert werden können. Um den Entwicklern die Möglichkeit zu geben, einzelne Resources nach ihren Bedürfnissen anzupassen, wurde das „Profiling“ in FHIR integriert. Mit Hilfe von Profilen lassen sich Regeln darüber einführen, welche Ressourcenelemente verwendet oder nicht verwendet werden und welche Elemente hinzugefügt werden, die nicht in der Basisspezifikation enthalten sind. [38]

Um ein FHIR-Profil zu erstellen, muss eine Ressource des Typs *StructureDefinition* erstellt werden. Im Body dieser Ressource können Ressourcen und Datentypen und die entsprechenden Erweiterungen und Einschränkungen zu diesen definiert werden. Jeder Ressource des Typs *StructureDefinition* wird eine URL zugewiesen, über die das Profil adressiert werden kann.

4.3 Kommunikation von FHIR-Ressourcen

FHIR ist als Schnittstellenspezifikation konzipiert, die den Inhalt der Daten, die zwischen Anwendungen im Gesundheitswesen ausgetauscht werden und die Art und Weise, wie der Austausch implementiert und verwaltet werden soll, spezifiziert. FHIR definiert mehrere Methoden zum Austausch von Daten zwischen Systemen – welcher dieser Ansätze verfolgt werden soll, hängt vom Verwendungszweck ab. Die meisten Implementierer konzentrieren sich jedoch auf den Ansatz der RESTful API, weshalb dieser im weiteren Verlauf der Arbeit detailliert beschrieben und angewendet wird. [35]

REST steht für Representational State Transfer und bezeichnet ein Programmierparadigma für Web-Services. Programmierparadigmen sind verschiedene grundlegende Ansätze und Techniken für die Programmierung von Software und beschreiben den grundlegenden Stil, in dem ein Programm entworfen wird. REST wurde entworfen, um den Anforderungen des modernen World Wide Web besser gerecht zu werden. Die Kernidee bei REST ist das Konzept der Ressourcen: Alles was in REST adressierbar ist, ist eine Ressource. [36]

Für eine im REST-Architekturstil verwendbare Ressource gelten folgenden Prinzipien: Jede Ressource muss über einen Unique Resource Identifier (URI) adressierbar sein, sodass sie eindeutig identifiziert werden kann. Außerdem muss die Kommunikation stets zustandslos sein, d.h. dass jede Anfrage (auch vom selben Auftraggeber), als unabhängige Transaktion betrachtet wird. Daher müssen bei jeder Anfrage sämtliche erforderlichen Informationen vollständig mitgesendet werden. Bei REST können verschiedene Repräsentationen einer Ressource existieren. Somit kann ein Client eine Ressource beispielsweise im JSON-, aber auch im XML-Format anfordern. [36]

API steht für Application Programming Interface (deutsch: Programmierschnittstelle) und dient dazu Informationen zwischen einer Anwendung und einzelnen Programmteilen standardisiert auszutauschen. Die API definiert in welcher Form die Informationen und Daten vom Anwendungsmodul entgegengenommen und wieder zurückgesendet werden. Die Übergabe von Daten und Befehlen erfolgt somit strukturiert und nach einer zuvor definierten Syntax. [37]

Die FHIR-RESTful-API ist eine Allzweckschnittstelle, die für Push- und Pull-Daten zwischen Systemen verwendet werden kann. Da FHIR ein Standard ist, setzt man auf die Standardisierung von Ressourcenstrukturen und Schnittstellen, was der Schlüssel zur Sicherstellung einer konsistenten Interoperabilität über verschiedene Systeme ist. In FHIR sind verschiedene Interaktionen mit FHIR Ressourcen definiert, die über HTTP-Operatoren realisiert werden:

- GET: lesender Zugriff auf eine Ressource
- POST: neue Ressource wird erstellt
- PUT: Ressource wird erstellt oder bearbeitet, falls die URI bereits bekannt ist
- DELETE: Ressource wird gelöscht

In der offiziellen FHIR-Spezifikation wird ein Style-Guide angegeben, in dem der Aufbau einer solchen Interaktion definiert ist.

```
VERB [base]/[type]/[id] {?_format=[mime-type]}
```

Das erste Wort einer Interaktion ist das HTTP-Verb, das für die Interaktion genutzt wird, also GET, POST, PUT oder DELETE. Darauf folgte die Base-URL des Service, also die Adresse, unter der alle von dieser Schnittstelle definierten Ressourcen zu finden sind, also beispielsweise die URL <https://fhir-server.org>. Darauf folgend wird der Ressourcen-Typ und die ID der Ressource angegeben. So könnte beispielsweise der in Abschnitt 4.2 erzeugte Patient über folgende Interaktion angesprochen werden:

```
GET https://fhir-server.org/Patient/1933612
```

Über den Mime-Typ kann angegeben werden, in welchem Format die vom FHIR-Server empfangenen Daten repräsentiert werden sollen (also JSON, XML oder Turtle). Die geschweiften Klammern signalisieren, dass die Angabe des gewünschten Formats optional ist. Sollen die Daten im JSON-Format empfangen werden, wird die Interaktion dem Style-Guide entsprechend um den Mime-Type erweitert:

```
GET https://fhir-server.org/Patient/1933612?_format=json
```

Der Host eines FHIR-Servers kann selbst entscheiden welche Interaktionen und Ressourcen-Typen auf dem entsprechenden Server unterstützt werden. Daher muss jeder FHIR-Server ein CapabilityStatement bereitstellen, das angibt, welche Operationen und Ressourcen unterstützt werden.

In der offiziellen FHIR-Spezifikation werden die zu implementierenden HTTP-Operationen als „Interaktionen“ bezeichnet und in drei Gruppen aufgeteilt:

- Interaktionen auf Instanz-Ebene: Eine Interaktion auf eine bestimmte Instanz eines Ressourcen-Typen – beispielsweise ein lesender Zugriff auf eine Ressource des Typs „Patient“
- Interaktion auf Typ-Ebene: Eine Interaktion auf einen bestimmten Ressourcen-Typen – beispielsweise die Erzeugung einer neuen Ressource des Typs Patient
- Interaktion auf Gesamtes System: Eine Interaktion auf das gesamte System – beispielsweise die Abfrage des CapabilityStatements.

Alle in der FHIR-Spezifikation aufgeführten Interaktionen sind in Tabelle 3-5 zusammengefasst.

Tabelle 3: Interaktion auf Instanz-Ebene

Interaktion	Erklärung	Definition
read	Lesen des aktuellen Zustands der Ressource	GET [base]/[type]/[id] {?_format=[mime-type]}
vread	Lesen des Zustands einer bestimmten Version der Ressource; „vid“ ist der Bezeichner für die Versions-ID der Ressource	GET [base]/[type]/[id]/_history/[vid] {?_format=[mime-type]}
update	Aktualisiert eine vorhandene Ressource anhand ihrer ID (oder erstellt sie, wenn sie neu ist)	PUT [base]/[type]/[id] {?_format=[mime-type]}
patch	Aktualisiert eine bestehende Ressource durch einen Satz von Änderungen an ihr	PATCH [base]/[type]/[id] {?_format=[mime-type]}
delete	Löschen einer Ressource	DELETE [base]/[type]/[id]
history	Ruft die Änderungshistorie für eine bestimmte Ressource ab	GET [base]/[type]/[id]/_history {?_format=[mime-type]}

Tabelle 4: Interaktion auf Typ-Ebene

create	Erstellen einer neuen Ressource mit einer vom Server zugewiesenen ID	POST [base]/[type] {?_format=[mime-type]}
search	Durchsuchen des Ressourcen-Typs basierend auf einigen Filterkriterien	GET [base]/[type]{?[parameters]}&_format=[mime-type]}
history	Abrufen der Änderungshistorie für einen bestimmten Ressourcen-Typ	GET [base]/[type]/_history {?_format=[mime-type]}

Tabelle 5: Interaktion auf Gesamtes System

capabilities	Abrufen einer Fähigkeitsanweisung für das System	GET [base]/metadata{?_format=[mime-type]}
batch	Aktualisieren, Erstellen oder Löschen einer Gruppe von Ressourcen in einer einzigen Interaktion	POST [base] {?_format=[mime-type]}
history	Ruft die Änderungshistorie für alle Ressourcen ab	GET [base]/_history{?_format=[mime-type]}
search	Suche über alle Ressourcen-Typen basierend auf einigen Filterkriterien	GET [base]/_history{?[parameters]}&_format=[mime-type]}

4.4 Anwendung von FHIR

Deutschland arbeitet immer noch an den Grundlagen zur Schaffung einer digitalen Infrastruktur im Gesundheitswesen. Besonders beim einrichtungsübergreifenden Datenaustausch sind große Defizite vorhanden: Laut einer Studie der Unternehmensberatung McKinsey&Company aus dem Jahr 2020, kommunizieren 93 Prozent der niedergelassenen Ärzte in Deutschland noch überwiegend in Papierform mit Krankenhäusern [58]. Anders sieht die Situation bei Krankenkassen aus, die ihren Versicherten bereits jetzt einige digitalen Services zur Verfügung stellen, die stetig an Beliebtheit gewinnen: Die Anzahl der Downloads von digitalen Gesundheitsanwendung hat sich vom Jahr 2019 zum Jahr 202 auf 2 Millionen verdoppelt. [58]

Das Bundesministerium für Gesundheit erkannte in den letzten Jahren, welche Chancen sich durch die Digitalisierung des Gesundheitswesens ergeben und hat daraufhin den rechtlichen Rahmen geschaffen, um diese Digitalisierung voranzutreiben. Mit der Einführung des Digitale-Versorgung-Gesetz (DVG) am 19. Dezember 2019, wurde die Rechtsgrundlage für den Anspruch von Versicherten auf die Versorgung mit digitalen Gesundheitsanwendungen geschaffen. Außerdem wurden Krankenhäuser und Apotheken dazu verpflichtet, sich an die Telematik-Infrastruktur anzubinden – weiteren Leistungserbringern wurde der Anschluss auf freiwilliger Basis ermöglicht. [59]

Besonders das Interesse an FHIR ist derzeit enorm und das aus verschiedenen Gründen: Einerseits erkennen Hersteller zunehmend den Nutzen des Standards. FHIR wird immer häufiger an Stelle von proprietären Schnittstellen verwendet, wenn Hersteller beispielsweise eigene Web-Applikationen oder mobile Apps an ihre Back-End Infrastruktur anbinden. Hersteller erkennen zunehmend, dass der Einsatz von FHIR mittelfristig von großem Nutzen sein könnte, da die Entwicklung neuer Software durch die vordefinierten, wiederverwendbare Komponenten, die FHIR bietet, stark vereinfacht wird. Andererseits steigt die Zahl der Software-Entwickler, die FHIR-Schnittstellen umsetzen müssen stetig an, da FHIR in immer mehr Spezifikationen und Leitfäden vorgeschrieben wird. [60]

Zwei Beispiele hierfür sind die medizinischen Informationsobjekte der KBV und die digitalen Gesundheitsanwendungen des Bundesinstituts für Arzneimittel und Medizinprodukte (BfArM), die im Folgenden vorgestellt werden.

4.4.1 Medizinische Informationsobjekte (MIOs)

Als Grund für den mangelhaften Datenaustausch im deutschen Gesundheitswesen, werden von der Kassenärztlichen Bundesvereinigung (KBV) die verschiedenen Datenaustauschformate angegeben, die untereinander nicht kompatibel sind und somit eine einfache Kommunikation kaum möglich machen [39]. Um diesem Problem entgegenzuwirken, entwickelt die KBV Medizinische Informationsobjekte (MIOs), die digitale Informationsbausteine mit medizinischen Daten darstellen und interoperabel von jedem System im Gesundheitswesen lesbar sein sollen. Somit soll ein Austausch und eine Verarbeitung von Gesundheitsdaten unabhängig vom verwendeten Softwaresystem ermöglicht werden. [41]

Ein Beispiel für ein medizinisches Informationsobjekt ist der elektronische Impfpass, der Daten zum Patienten, zum Impfstoff und zu impfrelevanten Erkrankungen enthält. Hierbei ist jedoch nicht nur der Impfpass selbst ein MIO: auch die einzelnen medizinischen Informationen, die im Impfpass enthalten sind, stellen ihrerseits ebenfalls ein medizinisches Informationsobjekt dar. [40]

Die Entwicklung der MIOs wurde auch in Hinblick auf die von der gematik entwickelte elektronische Patientenakte (ePA) vorangetrieben. Gesetzlich Versicherte in Deutschland haben seit dem 01.01.2021 ein Anrecht auf die Nutzung der elektronischen Patientenakte, die von der jeweiligen Krankenkasse als kostenlose App bereitgestellt wird. Die ePA funktioniert dabei Hand in Hand mit MIOs: In der ePA werden medizinische Daten eines Patienten gespeichert, die über ePA-Funktionalitäten hinzugefügt, aktualisiert und entfernt werden können. MIOs fungieren dabei als Standard, der die Lesbarkeit, die Speicherung und die Verarbeitung der Daten für verschiedene Personen ermöglicht und somit die Struktur der Daten bestimmt. [40]

Auf der offiziellen Website der KBV wird die elektronische Patientenakte mit einem Schrank verglichen, in dem die Gesundheitsinformationen in Schubladen gelagert werden. Die Medizinischen Informationsobjekte stellen in dieser Metapher die Bauanleitung für die Fächer der Schubladen dar. Sie definieren welche Informationen in welches Fach gehören und ermöglichen es zugriffsberechtigten Akteuren des Gesundheitswesens und den Versicherten selbst, die Schubladen zu öffnen und auf die Informationen zuzugreifen. [40]

Um die gewünschte syntaktische Interoperabilität zu erreichen, wird FHIR in Kombination mit internationalen Standards zur Dokumentation der Gesundheitsdaten (ICD 10 GM und SNOMED CT) verwendet. Dabei wird ausgenutzt, dass FHIR ein plattformunabhängiger, moderner Webstandard ist und sich für Desktopanwendungen, sowie für Apps verwenden lässt. Der größte Vorteil den FHIR für MIOs bereithält, ist, dass Ressourcen nicht wie in einem Dokument jedes Mal von neuem erzeugt werden müssen, sondern erneut Verwendung finden. Somit können gewisse Daten, die für mehrere MIOs relevant sind in sogenannten KBV-Basisprofilen festgehalten werden, die wiederverwendet werden können – gleiche Information soll auch an gleicher Stelle wiederzufinden sein. [39]

In den KBV-Basisprofilen wird neben der Wiederverwendbarkeit auch die Anpassungsfähigkeit (Profilierung) einzelner FHIR-Ressourcen ausgenutzt, indem die Ressourcen genau auf die Bedürfnisse der KBV angepasst werden. Somit kann sichergestellt werden, dass alle Ressourcen, die für die MIOs verwendet werden, einheitlich aufgebaut sind und alle geforderten Informationen enthalten. Die Basisprofile sind also eine stabile Grundlage für die Entwicklung der MIOs und der FHIR-Schnittstellen.

Ein Beispiel für ein KBV-Basisprofil ist das Profil „KBV-Basis-PatientIn“, dass in Abbildung 4 dargestellt ist. In diesem Profil wird festgelegt, wie Ressourcen des Typs *Patient* aufgebaut sein müssen, damit sie mit den Regeln der KBV konform sind und in Medizinischen Informationsobjekten genutzt werden können. Dabei wird festgelegt, welche Attribute in der Ressource enthalten sein müssen, welche Attribute enthalten sein können und von welchem Datentyp diese sein müssen.

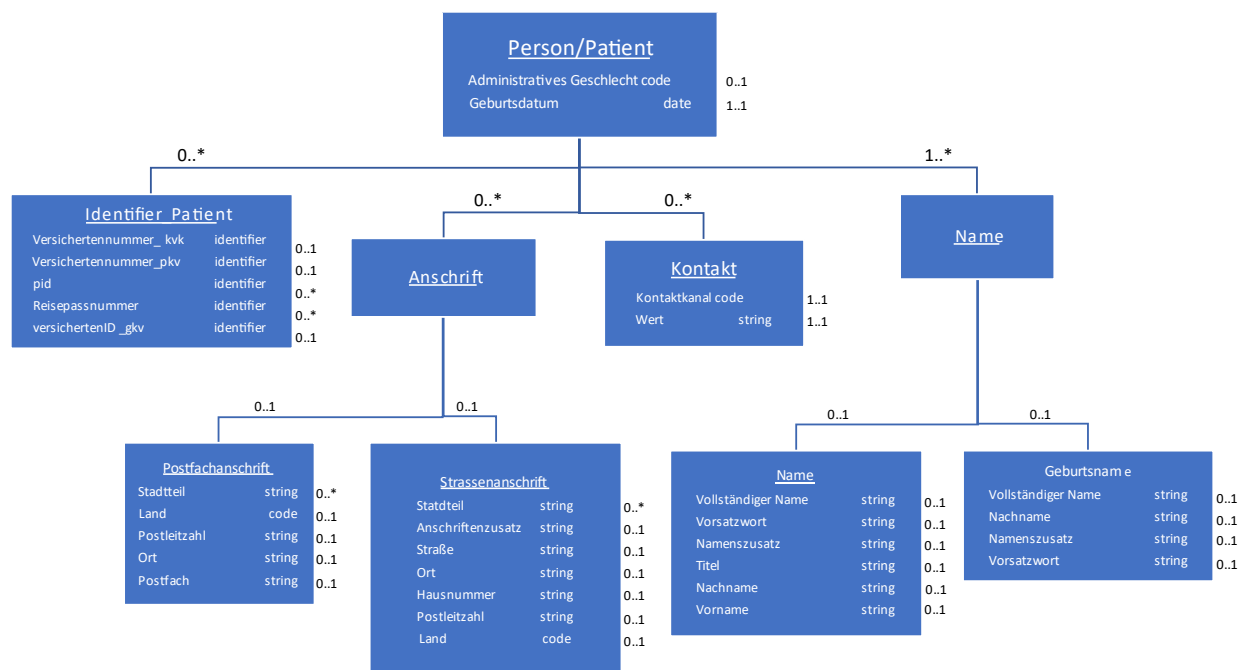


Abbildung 4: KBV-Basis-Profil: PatientIn

Quelle: In Anlehnung an <https://mio.kbv.de/display/BASE1X0/PatientIn>

Stand Mai 2021 sind sieben KBV-Basisprofile definiert: [42]

- KBV-Basis-PatientIn
- KBV-Basis-Behandelnde-Person
- KBV-Basis-Einrichtung
- KBV-Basis-Diagnose
- KBV-Basis-Apgar-Score
- KBV-Basis-Vitalzeichen & Körpermaße
- KBV-Basis-Terminologie

4.4.2 Entwicklung von MIOs

Zu Beginn des Entstehungsprozesses eines medizinischen Informationsobjekts steht eine sorgfältige Analyse in Zusammenarbeit mit beteiligten Verbänden und Organisationen. Auf Grundlage der ersten Analyse wird das MIO inhaltlich modelliert und die Spezifikationen werden erarbeitet. Darauf folgt die erste, 6-wöchige Feedback-Phase. Um der Öffentlichkeit die Möglichkeit zu geben, aktiv zu kommentieren und an der Entwicklung teilzuhaben, wurde die Plattform <https://mio.kbv.de> erstellt, auf der Interessierte Hinweise geben, Verbesserungsvorschläge machen und Fragen stellen können. Außerdem veranstaltet die KBV in dieser Phase regelmäßig, für die Öffentlichkeit zugängliche Websessions, in denen der neuste Stand der Dinge aufgezeigt wird (nähere Informationen und Termine für Interessierte lassen sich ebenfalls unter <https://mio.kbv.de> finden). Anschließend werden die Kommentare bearbeitet, Änderungen vorgenommen und die finale Form des Medizinischen Informationsobjekts erstellt. Die Festlegung des MIOs erfolgt durch den Vorstand der KBV. Stand Mai 2021 sind vier MIOs definiert: [41]

1. Elektronischer Impfpass

Impfungen sind die wichtigste Maßnahme zur Vermeidung von Infektionskrankheiten und basieren auf dem §22 des Infektionsschutzgesetzes, indem auch die Dokumentationsinhalte konkret vorgegeben werden. Der Impfpass ist historisch bedingt ein papierbasiertes Dokument, in dem die Impfungen handschriftlich festgehalten werden. Dies birgt die Gefahr, dass Impfungen nicht vollständig dokumentiert werden und so nicht-erforderliche Impfungen erfolgen könnten. Der elektronische Impfpass hingegen liefert eine zentral gespeicherte, elektronische Impfdokumentation, die stets eingesehen und ergänzt werden kann. Außerdem ist der Verlust der Daten weniger leicht möglich und die Daten können als Basis für ein Impfmanagement genutzt werden, dass Impfeempfehlungen ausspricht und Impflücken aufdeckt. Die erste Version des Impfpasses ist bereits festgelegt - Version 1.1.0 befindet sich momentan in der Festlegungsphase. [43]

2. Elektronisches Zahnärztliches Bonusheft

Mit dem 1989 durch das „Gesundheitsreform-Gesetz“ eingeführten Zahnärztlichen Bonusheft haben Versicherte der GKV in Deutschland die Möglichkeit, die Durchführung von zahnärztlichen Vorsorgeuntersuchungen schriftlich dokumentieren zu lassen. Damit wurde das Ziel verfolgt, den Versicherten regelmäßig an zahnärztliche Kontrolluntersuchungen zu erinnern und so Erkrankungen der Zähne und des Mundraums vorzubeugen, bzw. sie frühzeitig zu erkennen. Das Zahnärztliche Bonusheft soll in Form eines Medizinischen Informationsobjekts in die elektronische Patientenakte integriert werden. Gemäß der ePA-Struktur wird dazu für jeden elektronischen Bonusheftnachweis eine Datei im XML-Format erzeugt. Die elektronische Fassung des Bonusheftes wird von der KBV als „sinnvoll und zeitgemäß“ beschrieben [44]. Die erste Version des elektronischen Zahnärztlichen Bonushefts ist bereits festgelegt - Version 1.1.0 befindet sich momentan in der Kommentierungsphase. [44]

3. Elektronischer Mutterpass

Um die Ergebnisse der Vorsorgeuntersuchungen während der Schwangerschaft und nach der Entbindung festzuhalten, wurde 1961 der Mutterpass in Deutschland eingeführt. Die werdende Mutter erhält diesen Pass in papiergebundener Form zum Zeitpunkt der offiziellen Feststellung der Schwangerschaft und sollte diesen stets bei sich tragen, damit alle wichtigen Informationen für die Geburt und in medizinischen Notfällen bereitliegen. Außerdem sollte er nach der Schwangerschaft aufbewahrt werden, da er für weitere Schwangerschaften relevant sein könnte. Mit Hilfe des elektronischen Mutterpasses, wären diese Informationen jederzeit verfügbar und könnten fortlaufend ergänzt werden. Zudem erleichtert die elektronische Form des Mutterpasses den Austausch von Informationen und macht einen Verlust der Daten wesentlich unwahrscheinlicher. Schwangere könnten darüber hinaus durch Managementsysteme an bevorstehende Untersuchungen erinnert werden und Hinweise zu medizinischen Befunden und Maßnahmen erhalten. Die erste Version des elektronischen Mutterpasses ist bereits festgelegt. [45]

4. Elektronisches U-Heft

Das U-Heft (Kinderuntersuchungsheft) dient zur Dokumentation der Früherkennungsuntersuchungen und liegt seit 1971 in Papierform in deutscher Sprache vor. Es umfasst insgesamt 10 U-Untersuchungen, die von der Geburt an bis etwa zum Alter von 5 Jahren durchgeführt werden. Mit Hilfe der U-Untersuchungen können körperliche, geistige und psychosoziale Auffälligkeiten frühzeitig erkannt und Präventions- und Behandlungsempfehlungen abgeleitet werden. Jede U-Untersuchung ist in sich geschlossen und soll immer nur den Entwicklungsstand zum gegenwärtigen Zeitpunkt dokumentieren. Ziel des MIOs ist es, eine möglichst inhalt- und strukturerhaltene, elektronische Version des analogen Dokuments zu erstellen, damit sich die bisherigen Nutzer nicht auf eine völlig neue Art der Dokumentation einstellen müssen. Langfristig soll das elektronische U-Heft in die ePA eingegliedert werden und von der erleichterten Verfügbarkeit und Handhabung profitieren. Die erste Version des elektronischen U-Heft ist bereits festgelegt. [46]

4.4.3 Digitale Gesundheitsanwendungen (DiGA)

Digitale Gesundheitsanwendungen (DiGAs) sind Medizinprodukte mit gesundheitsbezogener Zweckbestimmung. Die Hauptfunktion von DiGAs beruht auf digitalen Technologien und ist dazu bestimmt, die Förderung der Gesundheit sowie die Erkennung, Überwachung und Behandlung von Krankheiten zu unterstützen. Das Verzeichnis aller erstattungsfähiger digitalen Gesundheitsanwendungen wird vom Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM) erstellt und geführt. [47]

Eine DiGA ist ein CE-gekennzeichnetes Medizinprodukt der Risikoklasse I oder IIa und wird vom Patienten oder vom Leistungserbringer und dem Patienten gemeinsam genutzt, weshalb man auch von „digitalen Helfern“ in der Hand des Patienten spricht. Seit dem 19.12.2019 haben gesetzlich Versicherte in Deutschland auf Grund des Digitale-Versorgungs-Gesetzes (DVG) Anspruch auf DiGAs, die auch als „Apps auf Rezept“ bezeichnet werden. Diese Apps auf Rezept können von Ärzten und Psychotherapeuten verordnet und von der Krankenkasse erstattet werden. Unter <https://diga.bfarm.de/de/verzeichnis> sind alle bisher zugelassenen Digitalen Gesundheitsanwendungen aufgelistet. [47]

Jede zugelassene DiGA wurde vom BfArM intensiv geprüft. Das Bewertungsverfahren ist dabei als zügiger „Fast-Track“ konzipiert: Nach dem Eingang des Antrags beträgt die Bewertungszeit durch das BfArM höchstens drei Monate. In diesem Zeitraum werden die Herstellerangaben zu den geforderten Produkteigenschaften, von Datenschutz bis hin zu Benutzerfreundlichkeit, geprüft. Außerdem muss vom Hersteller ein Nachweis für die mit der DiGA realisierbaren positiven Versorgungseffekte erbracht werden. Kann dieser zum Zeitpunkt der Einführung noch nicht erbracht werden, kann die DiGA probeweise für ein Jahr ins DiGA-Verzeichnis aufgenommen werden. In dieser Zeit muss der Hersteller den Nachweis eines Effekts, durch den sich der gesundheitliche Zustand eines Patienten, bzw. der Umgang eines Patienten mit seiner Erkrankung verbessert, erbringen. [47]

Auch im Falle von DiGAs wird großer Wert auf Interoperabilität gesetzt. Im Leitfaden für Entwickler, schreibt das BfArM vor, dass (sofern vorhanden) ein von der KBV definiertes MIO (oder ein im vesta-Verzeichnis empfohlener Standard) zur Umsetzung der interoperablen Schnittstelle verwendet werden soll. Sofern kein MIO zur Umsetzung vorhanden ist, soll ein international anerkannter Standard verwendet werden. Hier wird im Leitfaden explizit der FHIR-Standard angesprochen. [48]

Abschließend lässt sich festhalten, dass auch wenn die digitale Gesundheitsversorgung in Deutschland noch in den Startlöschern steckt, ein eindeutiger Trend in der Bevölkerung, sowie in der Regierung zu erkennen ist. Durch entsprechende Gesetzesentwürfe konnte der Grundstein für digitale Angebote wie die elektronische Patientenakte, das elektronische Rezept und digitale Gesundheitsanwendungen gelegt werden, die in Deutschland immer mehr an Beliebtheit gewinnen. Im Rahmen dieser Arbeit soll nun eine fiktive Digitale Gesundheitsanwendung erstellt werden, die prototypisch die Erfassung von Vitaldaten mit Consumer-Geräten umsetzt. Im weiteren Verlauf der Arbeit wird zuerst das Konzept und anschließend die Realisierung der DiGA detailliert beschrieben.

5 Konzept

Das Ziel der Arbeit ist es, eine fiktive digitale Gesundheitsanwendung zu erstellen. Die so erstellte DiGA soll prototypisch die Erfassung von Vitaldaten mit Consumer-Geräten umsetzen und diese mit Hilfe eines Kommunikationsservers auf einem FHIR-Testserver ablegen. Dazu wird verschiedene Hard- und Software verwendet, die in den folgenden Abschnitten vorgestellt wird.

5.1 Verwendete Hard- und Software

Zur Erfassung der Vitaldaten wurden Consumer-Geräte des Unternehmens Withings verwendet, die sich durch einfache Handhabung und Bedienung mit Hilfe einer kostenlosen App auszeichnen. Die App wird auf einem Samsung Galaxy J5 installiert, sodass alle Messergebnisse automatisch über WLAN mit dem Smartphone synchronisiert werden. Um die Messergebnisse zu verarbeiten, werden diese auf einen Raspberry Pi 3B+ geladen, auf dem der Kommunikationsserver Mirth Connect installiert ist. Mit Hilfe des Mirth Connects werden die Vitalparameter in FHIR-Ressourcen konvertiert und auf einen FHIR-Testserver hochgeladen. Im Folgenden werden alle verwendeten Komponenten vorgestellt:

Withings

Withings ist ein französisches Unternehmen, das sich auf die Entwicklung von Gesundheitsgeräten und Apps spezialisiert hat, die dazu bestimmt sind, die Gesundheit der Kunden nachhaltig zu verbessern und das tägliche Wohlbefinden zu steigern. Im Sortiment befinden sich Körperanalysewaagen, Fitnessuhren, Blutdruckmessgeräte und vieles mehr. Withings wurde 2016 von Nokia übernommen und wurde Teil von „Nokia Health“. Im Jahr 2018 wurde das Unternehmen wieder unabhängig unter dem Namen „Withings“. [49]

Health Mate App

Die Health Mate App ist eine von Withings entwickelte Tracking-App für Fitness, Aktivitäten und Gesundheitsdaten. Alle von den Gesundheitsgeräten erfassten Daten werden mit der Health Mate App synchronisiert und dort visuell aufbereitet. Mit Hilfe der Health Mate App lässt sich der Blutdruck kontrollieren, das Gewicht managen oder die Körpertemperatur tracken. [49]

BPM Core – Blood Pressure Monitor

BPM Core ist ein kabelloses Blutdruckmessgerät, mit dem Blutdruck- und Herzfrequenz-Messwerte erfasst werden können (Abb. 5). Die Daten werden automatisch über Bluetooth oder WLAN mit der Health Mate App synchronisiert, wo sie dokumentiert werden und mit einem Arzt geteilt werden können. Von Withings selbst wird das BPM Core als „erstes WLAN-Blutdruckmessgerät“ angepriesen. [49]



Abbildung 5: Withings BPM Core: Blood Pressure Monitor
Quelle: <https://www.withings.com/de/de/press#products-resources>

Body+

Body+ ist eine WLAN-Waage, die Messungen der kompletten Körperzusammensetzung, also Gewicht, Körperfett- und Wasseranteil, sowie Knochen- und Muskelmasse, durchführt (Abb. 6). Die Waage zeigt die Messergebnisse sofort auf dem Display an und synchronisiert die Daten mit der Health Mate App. Dort werden die Daten visualisiert und aufgearbeitet, um Trends in der Gewichtsentwicklung und Bilder der Körperzusammensetzung zu erstellen. [49]

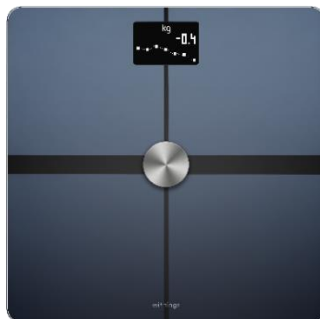


Abbildung 6: Withings Body+
Quelle: <https://www.withings.com/de/de/press#products-resources>

Thermo – Das intelligente Schläfenthermometer

Thermo ist ein WLAN-Schläfenthermometer, das durch einen Scan der Stirnoberfläche die Temperatur an der Schläfenarterie misst (Abb. 7). Somit kann die Körpertemperatur auf hygienische Art und Weise bestimmt werden, da kein Kontakt mit Speichel, Ohrenschmalz oder anderen Körperflüssigkeiten notwendig ist. Die Messwerte werden automatisch mit der Health Mate App synchronisiert, mit deren Hilfe Temperaturmesswerte verfolgt, Erinnerungen erstellt und Symptome und Medikamente zur Beratung eingegeben werden können. [49]



Abbildung 7: Withings Thermo

Quelle: <https://www.withings.com/de/de/press#products-resources>

Samsung Galaxy J5

Das Samsung Galaxy J5 ist ein Smartphone der Galaxy-Reihe des Herstellers Samsung. Das Gerät hat einen 5,2 Zoll Touch-Display, 16 GB Speicher und läuft unter dem Betriebssystem Android 6.0.1.



Abbildung 8: Samsung Galaxy J5

Quelle: <https://www.amazon.de/Samsung-Smartphone-Touch-Display-Speicher-Android-wei%C3%9F/dp/B012CC7RSG>

Raspberry Pi 3B +

Ein Raspberry Pi ist ein kreditkartengroßer Computer, der an einen Monitor angeschlossen wird und eine Standardtastatur und -maus verwendet. Der Raspberry Pi 3 Model B+ ist das neueste Produkt der Raspberry Pi 3 Reihe und ist mit einem 64-Bit-Quad-Core Prozessor ausgestattet. Die Raspberry Pis werden von der in Großbritannien ansässigen Wohltätigkeitsorganisation „Raspberry Pi Foundation“ entwickelt, die sich dafür einsetzt, die Bildung von Erwachsenen und Kindern im Bereich Computer, Informatik und verwandten Themen zu fördern. [50] Auf dem Raspberry wird der Kommunikationsserver MIRTH Connect und die entwickelte Software installiert.



Abbildung 9: Raspberry Pi 3B+

Quelle: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

HAPI FHIR-Testserver

HAPI FHIR ist eine vollständige Open-Source Implementation des FHIR-Standards in Java. HAPI stellt einen kostenlosen FHIR-Testserver zur Verfügung, der einen Großteil der in der FHIR-Spezifikation aufgeführten Ressourcen und Interaktionen unterstützt. In dieser Arbeit wird der Endpoint R4 verwendet: <http://hapi.fhir.org/baseR4>. [61]

5.2 Kommunikationsserver Mirth Connect

Mirth Connect ist ein Open-Source Kommunikationsserver, der im Jahr 2006 von der Mirth Corporation entwickelt wurde. 2013 wurde Mirth Connect von NextGen Healthcare erworben, weshalb der Kommunikationsserver teilweise auch unter dem Namen „NextGen Connect Integration Engine“ zu finden ist. Mit dem Mirth Connect hat sich NextGen Healthcare das Ziel gesetzt, ein sicheres, effizientes und kosteneffektives Mittel zum Austausch von Gesundheitsinformationen zur Verfügung zu stellen. In diesem Prozess fungiert der Kommunikationsserver als eine Art Dolmetscher, der eine eingehende Nachricht analysiert und in eine für den Benutzer verständliche Sprache übersetzt. [51]

Die Schnittstelle, die eine solche Aufgabe übernimmt, wird „Channel“ genannt. Jeder Channel hat mehrere Connectoren: Einen Source-Connector, der die Aufgabe hat, Daten vom Source-System aufzunehmen und einen oder mehrere Destination-Connectoren, die die verarbeiteten Daten an das Destination-Systeme senden (Abb. 10). Folgende Connector-Typen werden in der Basisversion des Mirth Connects unterstützt [51]:

Tabelle 6: Source Connectoren des Kommunikationsservers Mirth Connect

Channel Reader	Wartet darauf, dass ihm andere Channels oder Prozesse Daten senden
DICOM Listener	Empfängt und verarbeitet Nachrichten im DICOM Datentyp
Database Reader	Stellt eine Verbindung zu einer externen Datenbank her, führt Abfragen durch und liest ausgewählte Zeilen
File Reader	Liest Dateien aus einem lokalen oder externen Verzeichnis in einem festgelegten Intervall
HTTP Listener	Agiert als HTTP-Server und wartet auf Anfragen von einem oder mehreren Clients
JMS Listener	Stellt Verbindung zu einem externen JMS-Provider her und liest Nachrichten aus einer Warteschlange oder einem Topic
JavaScript Reader	Führt ein benutzerdefiniertes JavaScript Script nach einem festgelegten Zeitplan aus
TCP Listener	Empfängt Nachrichten, die über eine TCP-Verbindung eingehen
Web Service Listener	Verwendet einen einfachen Service, der Nachrichten Strings aufnimmt und Response-Strings zurücksendet

Tabelle 7: Destination Connectoren des Kommunikationsservers Mirth Connect

Channel Writer	Versendet Nachrichten an andere interne Channels
DICOM Sender	Versendet Nachrichten im DICOM Datentyp
Database Writer	Stellt Verbindung zu einer externen Datenbank her und verändert diese auf gewünschte Weise
Document Writer	Nimmt eine HTML-Vorlage und konvertiert diese entweder in ein PDF- oder ein RTF-Dokument
File Writer	Schreibt Dateien in das lokale Dateisystem oder in ein externes Verzeichnis
HTTP Sender	Sendet HTTP-Anfrage an einen externen Webserver
JMS Sender	Stellt Verbindung zu einem externen JMS-Provider her und schreibt Nachrichten in eine Warteschlange oder einem Topic
JavaScript Writer	Führt ein benutzerdefiniertes JavaScript Script aus
SMTP Sender	Sendet eine E-Mail an ein oder mehrere angegebene Adressen
TCP Sender	Baut neue TCP Client-Verbindung auf und sendet darüber Nachrichten
Web Service Sender	Verbindet sich mit einem Endpoint und ruft definierte Operationen auf

Mit Hilfe einer Erweiterung, die unter <https://www.nextgen.com/products-and-services/integration-engine> kostenlos heruntergeladen werden kann, lässt sich ein FHIR Source- und Destination Connector im Mirth Connect integrieren. In der Erweiterung ist ein FHIR-Listener enthalten, der sich ähnlich wie der HTTP-Listener verhält und einige FHIR-spezifischen Einstellungen ermöglicht. Außerdem ist ein FHIR-Sender und ein Resource-Builder enthalten, mit denen Ressourcen an einen FHIR-Server gesendet werden können. [52]

Wird eine Nachricht von einem Source-Connector aufgenommen, kann diese mit Filtern und Transformern bearbeitet werden. Ein Filter ist ein Teil des Connectors, der entscheidet, ob eine Nachricht weiterverarbeitet werden soll, oder nicht. Ein Filter gibt „true“ oder „false“ zurück: Bei „true“ wird die Nachricht als „akzeptiert“ und bei „false“ als „gefiltert“ bezeichnet. Ein Transformer ist der Teil des Connectors, der eine Nachricht modifiziert, eine Nachricht von einem Format in ein anderes konvertiert oder Teile der Nachricht zur späteren Verwendung extrahiert. Jeder Transformator hat einen Eingangs- und einen Ausgangsdattentyp (die auch gleich sein können). [51]

Abhängig von der Wahl des Destination Connector-Typs, wird die Nachricht verarbeitet und an das ausgewählte Ziel gesendet. Jeder Channel hat mindestens einen Destination Connector – es können jedoch auch mehrere Destination Connectoren implementiert werden. Man könnte beispielsweise über einen HTTP-Listener Daten empfangen, diese Daten in eine Datei im lokalen Dateisystem ablegen und gleichzeitig einen Teil der Daten in eine benutzerdefinierte Datenbank einfügen. Auf jeden Destination Connector können zusätzliche Filter und Transformer angewendet werden, um die Ausgangsnachricht nach Wünschen des Anwenders zu konfigurieren. [51]

Der gesamte Prozess der Datenverarbeitung mit einem Mirth Connect Channel ist in Abbildung 10 grafisch dargestellt:

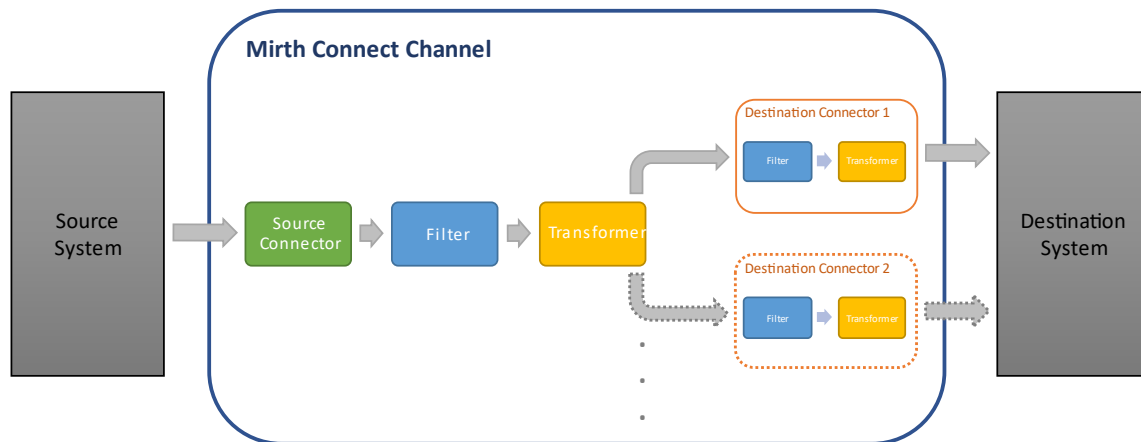


Abbildung 10: Funktionsweise eines Mirth Connect Channels
Quelle: eigene Darstellung

5.3 Entwurf des Gesamtsystems

Die in Kapitel 5.1 und 5.2 vorgestellte Hard- und Software soll genutzt werden, um das in Abbildung 11 als Flussdiagramm dargestellte Konzept umzusetzen.

Den Anfang des Prozesses stellt die Messung der Gesundheitsdaten mit den in Abschnitt 5.1 vorgestellten Messgeräten dar. Dazu werden das Blutdruckmessgeräte BPM, die Waage zur Messung der Körperzusammensetzung Body+ und das Schläfenthermometer Thermo mit der Health Mate App verbunden. Anschließend kann eine Messung durchgeführt werden, die von der Health Mate App automatisch erfasst und in proprietärer Darstellung auf dem lokalen Speicher des Smartphones abgelegt wird. Die Health Mate App fügt alle erfassten Messungen in eine SQL-Datenbank ein, in der Informationen über Art und Zeit der Messungen und über den Benutzer selbst gespeichert werden.

Um diese Messwerte zu verarbeiten, soll die Messdatei, in der alle bisher aufgenommenen Messungen enthalten sind, auf den Raspberry Pi kopiert werden. Dort sollen die Datenbankinformationen ausgelesen und in einem passendem Format in einem lokalen Verzeichnis des Raspberry Pis gespeichert werden. Wichtig ist hierbei, dass die Messwerte in einer Form dargestellt werden, die vom Kommunikationsserver Mirth Connect verarbeitet und in eine FHIR-Ressource konvertiert werden kann. Da die Health Mate App die SQL-Datenbank mit jeder Messung erweitert, muss eine Funktionalität integriert sein, die überprüft, ob es sich um eine neue Messung handelt, oder ob sie zu einem früheren Zeitpunkt bereits aufgenommen wurde.

Der Kommunikationsserver Mirth Connect läuft auf dem Raspberry Pi und überwacht das lokale Verzeichnis auf Änderung. Wird eine neue Datei im Verzeichnis abgelegt, wird diese vom Mirth Connect gelesen und in eine FHIR-Ressource konvertiert. Diese FHIR-Ressource wird auf einem FHIR-Testserver abgelegt.

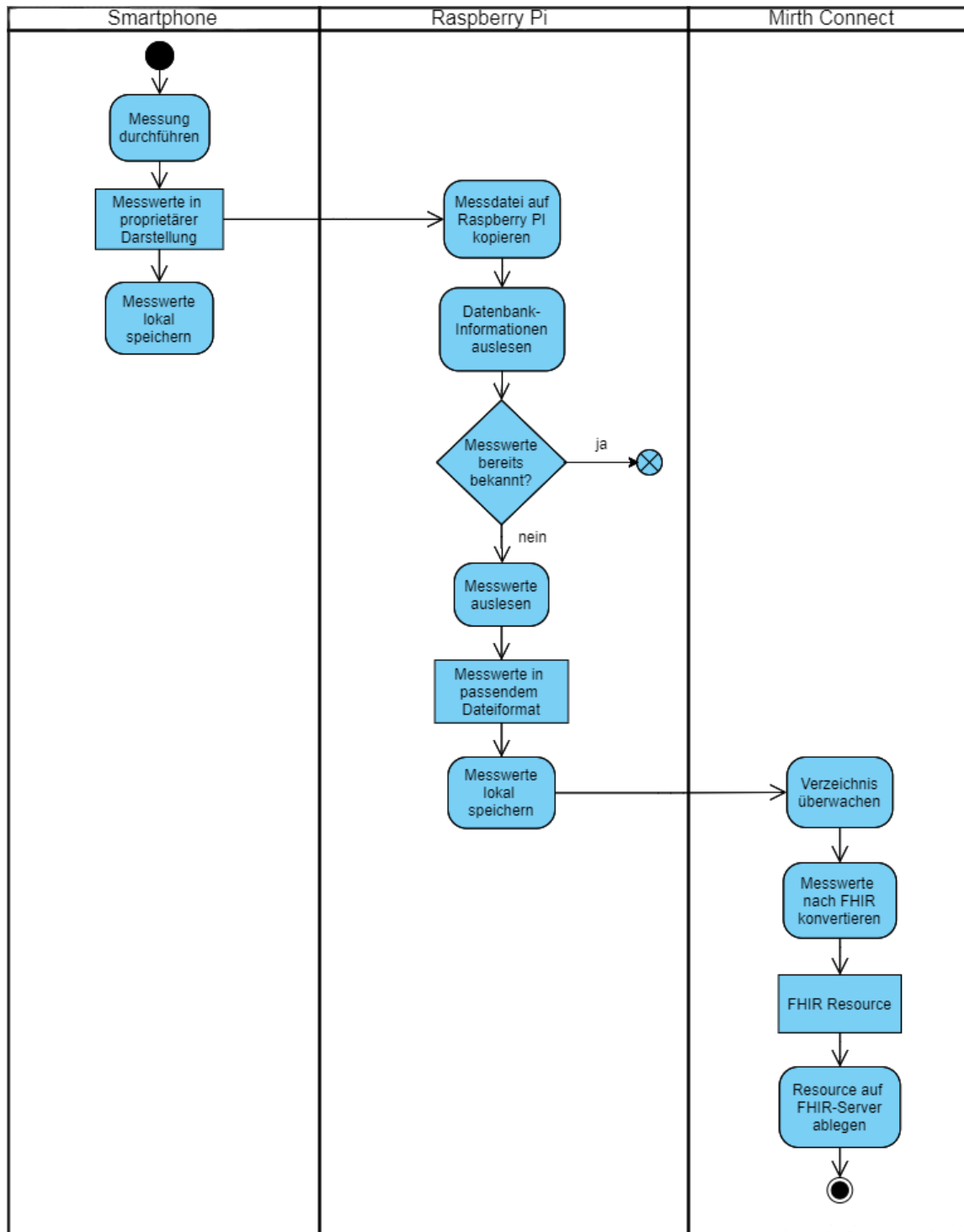


Abbildung 11: Grafische Darstellung des umzusetzenden Konzepts
Quelle: Eigene Darstellung

5.4 Darstellung der Vitalparameter als FHIR-Ressource

Mit den Withings-Messgeräten werden also die Vitalparameter Körpergewicht, BMI, Herzfrequenz, systolischer/diastolischer Blutdruck und Körpertemperatur aufgenommen. Um diese Vitalparameter auf dem FHIR-Testserver speichern zu können, werden sie in FHIR-Ressourcen des Typs *Observation* konvertiert. Observation-Ressourcen sind ein elementarer Bestandteil des FHIR-Standards und werden verwendet, um Labordaten, medizinische Messergebnisse oder klinische Befunde zu dokumentieren.

Besonders bei der Erfassung von Vitalparametern ist es von großer Wichtigkeit, ein einheitliches Vokabular und eine gemeinsame Syntax zu verwenden, um eine uneingeschränkte Wiederverwendung der Ergebnisse zu gewährleisten. Nur so kann die von FHIR angestrebte semantische-Interoperabilität erreicht werden. [56]

Zu diesem Zweck wurde in der FHIR-Spezifikation das Profil *observation-vitalsigns* definiert, das den Aufbau der Observation-Ressourcen für die Dokumentation der Vitalparameter Atemfrequenz, Herzfrequenz, Sauerstoffsättigung, Körpertemperatur, Körpergröße, Körpergewicht, BMI und Blutdruck vorgibt. Hierbei ist in der Spezifikation vorgeschrieben, dass jede FHIR-Implementierung der oben genannten Vitalparameter, zwingend diesem Profil entsprechen muss. [56]

Nach dem Profil *observation-vitalsigns* muss jede Observation-Ressource folgende Elemente enthalten:

1. Ein **Status-Code**, der angibt, ob die Messung bereits vollständig abgeschlossen und validiert ist, oder ob eventuell noch Änderungen vorgenommen werden können.
2. Das Element Category muss den **Category-Code 'vitals-signs'** und die Referenz zur FHIR-Spezifikation enthalten (<http://terminology.hl7.org/CodeSystem/observation-category>).
3. Ein **'Magic Value'**, der angibt welcher Vitalparameter aufgenommen wurde. In der FHIR-Spezifikation wurde hierfür der internationale Standard LOINC gewählt, der in Kapitel 3.2 vorgestellt wurde.
4. Das Ressourcen-Element *subject* muss eine **Referenz zu einer FHIR-Ressource des Typs Patient** enthalten.
5. Das Element *effectiveDateTime* muss den **Zeitpunkt** enthalten, an dem die Messung durchgeführt wurde. Dieser Zeitpunkt muss als Datentyp *dateTime* angegeben werden.
6. Ein **numerischer Ergebniswert**, der zusammen mit einem im Profil definierten Code für Einheiten angegeben werden muss. Wenn kein numerischer Ergebniswert vorhanden ist, muss im Element *dataAbsentReason* eine Begründung für die Abwesenheit angegeben werden.

Für jeden im Rahmen dieser Arbeit aufgenommenen Vitalparameter wurde ein Template im FHIR/JSON-Format erstellt. Dieses kann vom Kommunikationsserver Mirth-Connect als Grundlage genutzt und mit den aufgenommenen Messergebnissen zu einer vollständigen Ressource des Typs *Observation* ergänzt werden. In Abbildung 12 ist beispielhaft das Template für die Temperaturmessungen abgebildet – alle anderen Templates sind ähnlich aufgebaut und sind im Abgabeordner im Verzeichnis „FHIR Templates“ zu finden.

```
{
  "resourceType": "Observation",
  "status": "final",
  "category": [{
    "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/observation-category",
      "code": "vital-signs",
      "display": "Vital Signs"
    }]
  }],
  "code": {
    "coding": [{
      "system": "http://loinc.org",
      "code": "8310-5",
      "display": "Body temperature"
    }]
  },
  "subject": {
    "reference": "Patient/[patient_id]",
    "display": "[patient_name]"
  },
  "effectiveDateTime": "[dateTime]",
  "valueQuantity": {
    "value": [result_value],
    "unit": "°C",
    "system": "http://unitsofmeasure.org",
    "code": "Cel"
  }
}
```

Abbildung 12: Observation Ressource - Body Temperature

Quelle: Eigene Darstellung

Im folgenden Kapitel wird im Detail beschrieben, wie das in diesem Kapitel vorgestellte Konzept umgesetzt wurde. Falls der Leser nicht an den technischen Einzelheiten interessiert ist, kann er das Kapitel überspringen, da zusätzlich zur Abschlussarbeit ein Kurzfilm erstellt wurde, in dem alle Funktionalitäten dargestellt werden. Sofern die bisherigen Kapitel aufmerksam gelesen wurden, bekommt der Leser mit Hilfe der zusätzlichen Erläuterungen im Kurzfilm, auch ohne detailliertes Wissen über die technische Umsetzung einen guten Überblick über die Funktionsweise der im Projekt implementierten Komponenten.

6 Realisierung

In diesem Kapitel wird detailliert darauf eingegangen, wie das in Abschnitt 5 beschriebene Konzept umgesetzt wurde. Zur besseren Übersicht wurde der Weg der Gesundheitsdaten von der Messung bis zum FHIR-Testserver in vier Teile gegliedert: Die Durchführung der Messung, das Kopieren der Messdaten auf den Raspberry Pi, das Konvertieren der Messdaten aus der proprietären Darstellung in ein passendes Format und den Einsatz des Kommunikationsservers Mirth Connect zum Upload der Daten auf den FHIR-Testserver.

6.1 Durchführung der Messung

Zu Beginn wird die Health Mate App, über den App Store auf dem Smartphone Samsung Galaxy J5 installiert (Abb.13). Um die Gesundheitsgeräte mit der App zu synchronisieren, müssen diese lediglich gestartet werden. Die Health Mate App erkennt die Geräte automatisch und fügt sie der Geräteliste hinzu. Anschließend kann die erste Messung durchgeführt werden. Wird ein registriertes Gerät gestartet, erhält der Benutzer detaillierte Anweisung von der Health Mate App, wodurch - sofern die Anweisungen befolgt werden - eine fehlerfreie Messung garantiert wird.



Abbildung 13: Übersichtsseite der App „Health Mate“

Quelle: <https://www.withings.com/de/de/press#products-resources>

Nachdem die Messung abgeschlossen ist, wird das Ergebnis auf dem Display des Smartphones angezeigt und die Messdaten werden in eine SQL-Datenbank mit dem Namen „WithingsWiScale“ abgelegt. Diese besteht aus 27 Tabellen, die Informationen über den Nutzer, Aktivitäten und Messungen enthalten. Die eigentlichen Messergebnisse sind in der Tabelle „*measure*“ enthalten, von der ein Ausschnitt in Abbildung 14 dargestellt ist. Jeder Messung wird eine eindeutige ID (*id*) und Messgruppe (*measuregroup*) zugewiesen. Außerdem wird der User (*user*), der die Messung durchgeführt hat, die Unixzeit (*x*), zu der die Messung durchgeführt wurde und das Ergebnis der Messung (*y*) angegeben. Um zu erkennen, um welche Art der Messung es sich handelt, wird der Messtyp unter dem Bezeichner „*type*“ geführt. Folgende Messtypen sind für dieses Projekt relevant:

Tabelle 8: Für dieses Projekt relevante Messtypen der Withings-Datenbank

Type	Bedeutung
-12	Body-Mass-Index (BMI)
1	Körpergewicht
4	Körpergröße
8	Körperfett
9	Diastolischer Blutdruck
10	Systolischer Blutdruck

Type	Bedeutung
11	Herzfrequenz
71	Körpertemperatur
76	Muskelmasse des Körpers
77	Wasseranteil im Körper
88	Knochenmasse

Ein verdeutlichendes Beispiel ist der in Abb. 14 markierte rote Bereich. Hierbei handelt es sich um eine Messung mit dem Blutdruckmessgerät BPM, bei der der diastolische und systolische Blutdruck und die Herzfrequenz vom User 24702178 aufgenommen wurden. Jedem dieser Parameter wurde eine eindeutige ID zugewiesen. Da alle Ergebnisse während der gleichen Messung aufgenommen wurden, sind alle Messergebnisse unter der Messgruppe (*measuregroup*) 3 aufgeführt und die Unixzeit (*x*) ist bei allen drei Ergebnissen gleich. Das Messergebnis mit der *id*=16 hat den *type*=9 und einen *y*-Wert von 71.0 – d.h. es wurde ein diastolischer Blutdruck von 71.0 mmHg gemessen. Äquivalent dazu wurde ein systolischer Blutdruck von 137.0 mmHg und eine Herzfrequenz von 59 /min bestimmt.

	id	measuregroup	x	unit	value	type	y	algoVersion	appVersion	platform	user
	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern	Filtern
1	7	1	1612704355000.0	-3	80000.0	1	80.0	0	NULL	NULL	24702178
2	8	2	1612704355000.0	-2	185.0	4	1.85	0	NULL	NULL	24702178
3	12	1	1612704355000.0	0	23.3747253417969	-12	23.3747253417969	NULL	NULL	NULL	24702178
4	16	3	1612704462000.0	0	71.0	9	71.0	0	NULL	NULL	24702178
5	17	3	1612704462000.0	0	137.0	10	137.0	0	NULL	NULL	24702178
6	18	3	1612704462000.0	0	59.0	11	59.0	0	NULL	NULL	24702178
7	25	4	161287507000.0	0	83.0	9	83.0	0	NULL	NULL	24702178
8	26	4	161287507000.0	0	138.0	10	138.0	0	NULL	NULL	24702178
9	27	4	161287507000.0	0	62.0	11	62.0	0	NULL	NULL	24702178
10	31	5	1612876545000.0	0	80.0	9	80.0	0	NULL	NULL	24702178
11	32	5	1612876545000.0	0	135.0	10	135.0	0	NULL	NULL	24702178
12	33	5	1612876545000.0	0	61.0	11	61.0	0	NULL	NULL	24702178
13	34	6	1612877878000.0	-3	81938.0	1	81.938	0	NULL	NULL	24702178
14	46	6	1612877878000.0	0	23.9409790039062	-12	23.9409790039062	NULL	NULL	NULL	24702178
15	48	8	1599171774000.0	-3	22852.0	12	22.852	0	NULL	NULL	24702161
16	49	8	1599171774000.0	-3	36919.0	71	36.919	0	NULL	NULL	24702161
17	50	8	1599171774000.0	-3	35155.0	73	35.155	0	NULL	NULL	24702161
18	51	9	1599171781000.0	-3	22931.0	12	22.931	0	NULL	NULL	24702161
19	52	9	1599171781000.0	-3	36931.0	71	36.931	0	NULL	NULL	24702161
20	53	9	1599171781000.0	-3	35176.0	73	35.176	0	NULL	NULL	24702161

Abbildung 14: Ausschnitt der Tabelle „measure“ aus der Datenbank „WithingsWiScale“
Quelle: Eigene Aufnahme

6.2 Kopieren der Messdaten auf den Raspberry Pi

Um die Messwerte zu verarbeiten, soll die WithingsWiScale-Datenbank, die alle Messergebnisse enthält, auf den Raspberry Pi kopiert werden. Dazu muss das Smartphone gerootet sein, da der Benutzer sonst keinen Zugriff auf die App-internen Daten hat. Unter „rooten“ versteht man den Prozess, bei dem der Benutzer zusätzliche Zugriffsrechte auf Systemdaten seines Smartphones erhält und somit als sogenannter *Superuser* (oder *root*) agieren kann [53]. Für Android-Geräte existieren etliche Rooting-Tools wie *Magisk* oder *KingoRoot*, mit denen der Root-Prozess einfach und unkompliziert durchgeführt werden kann. Zu beachten ist hierbei, dass durch den Root-Vorgang die Herstellergarantie erlischt und das Smartphone nach dem Rooten anfälliger für Malware ist. [62]

Um die Withings-Datenbank auf den Raspberry Pi zu kopieren, wird das Smartphone über USB mit dem Raspberry Pi verbunden und es wird eine Android Debug Bridge (adb) eingerichtet (Abb. 15). *adb* ist ein vielseitiges Kommandozeilen-Tool, das zur Kommunikation mit einem Android-Gerät genutzt werden kann. Mit Hilfe der *adb*-Befehle lassen sich beispielsweise Apps installieren und debuggen und man erhält Zugriff auf eine Unix-Shell, mit der sich eine Vielzahl von Befehlen auf dem Android-Gerät ausführen lassen. Um die Android Debug Bridge auf einem über USB angeschlossenen Tool nutzen zu können, muss das USB-Debugging in den Entwickleroptionen des Smartphones aktiviert sein. [54]

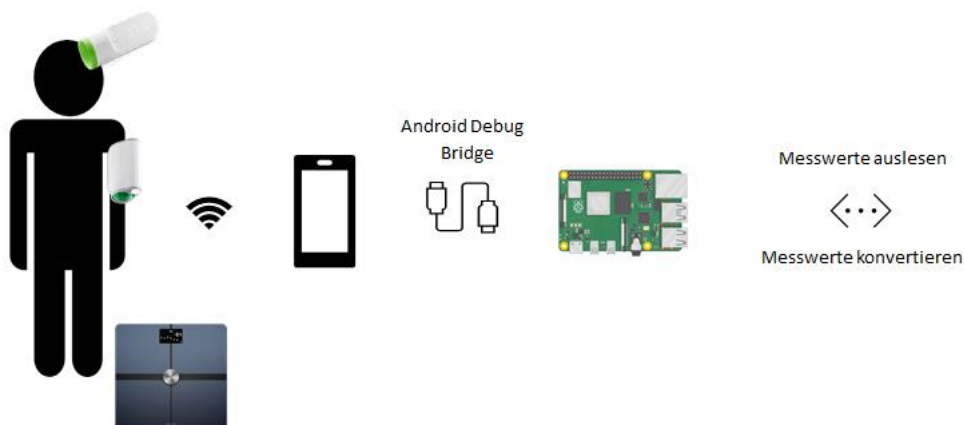


Abbildung 15: Visualisierung des Einsatzes der verwendeten Hardware

Quelle: Eigene Darstellung

Die Android Debug Bridge besteht aus drei Komponenten [54]:

1. Ein **Client**, der Befehle sendet und auf dem Raspberry Pi läuft. Der Client kann über die Befehlszeile aufgerufen werden, indem ein *adb*-Befehl eingegeben wird.
2. Ein **Daemon**, der als Hintergrundprozess Befehle auf einem Gerät ausführt und als *adbd* bezeichnet wird.
3. Ein **Server**, der die Kommunikation zwischen dem Client und dem Daemon verwaltet und als Hintergrundprozess auf dem Raspberry Pi läuft.

Mit dem Befehl „adb devices“ wird der Daemon gestartet und eine Liste der verbundenen Geräte wird angezeigt (Abb. 16). „29b6a955“ ist der Hostname des verwendeten Samsung Smartphones. Um Zugriff auf die Withings-Datenbank zu erhalten, muss der Daemon abdb über den Befehl „adb root“ mit Root-Rechten neugestartet werden, was jedoch vom Hersteller verweigert wird (Abb. 16).

```
pi@raspberrypi:~ $ adb devices
List of devices attached
* daemon not running; starting now at tcp:5037
* daemon started successfully
29b6a955    device

pi@raspberrypi:~ $ adb root
abdb cannot run as root in production builds
```

Abbildung 16: Terminal-Ausgabe des Raspberry Pis (adb devices/root)
Quelle: Eigene Aufnahme

In einer 2019 verfassten Abschlussarbeit der Hochschule Trier mit dem Titel „Assistenzsystem zur Vitalparametererfassung“, wurde sich bereits mit diesem Problem befasst. In der zugrundeliegenden Arbeit wurde das Problem über die App „abdb insecure“ gelöst, die dem Daemon Root-Rechte zuweist, sofern das Smartphone gerootet ist. Da die App vom Samsung J5 jedoch nicht unterstützt wird, wurden mit Hilfe des „chmod“-Befehls die Zugriffsrechte der WithingsWiScale-Datei so geändert, dass alle Benutzertypen lesenden und schreibenden Zugriff auf die Datenbank haben (Abb. 17). Anschließend kann die gesuchte Datei über den Befehl „adb pull“ in das entsprechende Verzeichnis des Raspberry Pis gezogen werden. Damit diese Abfrage automatisch abläuft, wurde ein Cronjob implementiert, der die in Abbildung 17 dargestellten Befehle, alle 30 Sekunden ausführt. Die vom Cronjob ausgeführten Skripte sind im Abgabeordner zu finden und werden in der zum Ordner gehörenden Informationsdatei „README.txt“ detailliert beschrieben.

```
pi@raspberrypi:~ $ adb shell su -c "chmod 666 /data/data/com.withings.wiscale2/databases/Withings-WiScale"
pi@raspberrypi:~ $ adb pull /data/data/com.withings.wiscale2/databases/Withings-WiScale /home/pi/Bachelorarbeit
/data/data/com.withings.wiscale2/databases/Withings-WiScale: 1 file pulled. 2.6 MB/s (294912 bytes in 0.108s)
```

Abbildung 17: Terminal-Ausgabe des Raspberry Pis (adb chmod/pull)
Quelle: Eigene Aufnahme

6.3 Konvertieren der Messdaten

Um die Messdaten weiter verarbeiten zu können, muss die Withings-Datenbank ausgelesen werden. Zu diesem Zweck wurde die Java-Applikation *GetWithingsData* erstellt, die die in der Datenbank „Withings-WiScale“ enthaltenen Tabellen ausliest und die Daten in passendem Format speichert, sodass diese vom Mirth-Connect verarbeitet werden können. Die Funktionalitäten, die hierbei verwendet wurden, basieren teilweise auf der in Kapitel 6.2 erwähnten Abschlussarbeit „Assistenzsystem zur Vitalparametererfassung“.

In der *main()*-Methode der Java-Applikation wird die Methode *readAllData()* aufgerufen, die zu Beginn eine Verbindung zur „Withings-SQL-Datenbank“ aufbaut. Anschließend werden mit dem SQL-Statement

```
SELECT * FROM measure JOIN users on measure.user=users.id ORDER BY measure.measuregroup
```

alle Inhalte der Tabelle „*measure*“ ausgewählt und nach der Messgruppe sortiert. Außerdem werden die passenden Benutzer-Informationen aus der Tabelle „*users*“ ausgewählt. Diese Inhalte werden zeilenweise ausgelesen und die Daten Mess-ID, Messgruppe, Messtyp, Messzeitpunkt, Messergebnis, User-ID, Vorname und Nachname werden temporär gespeichert.

Da die Withings-Datenbank mit jeder Messung erweitert wird, muss verhindert werden, dass die gleichen Messergebnisse mehrmals aufgenommen werden. Daher wird nachdem ein Messergebnis aufgenommen wurde, die Mess-ID in ein Check-File mit dem Namen „*AlreadyRecorded*“ geschrieben. Beim nächsten Aufruf der Methode *readAllData()* werden so alle bereits aufgenommen Messungen erkannt und nur die neuen Messungen werden weiterverarbeitet.

Wird eine Messung als „neu“ erkannt, wird die Methode *writeXML()* aufgerufen, die das Messergebnis und die dazugehörigen Informationen als XML-Dokument im Verzeichnis „*XML_Output*“ des Raspberry Pis speichert (Abb. 18).

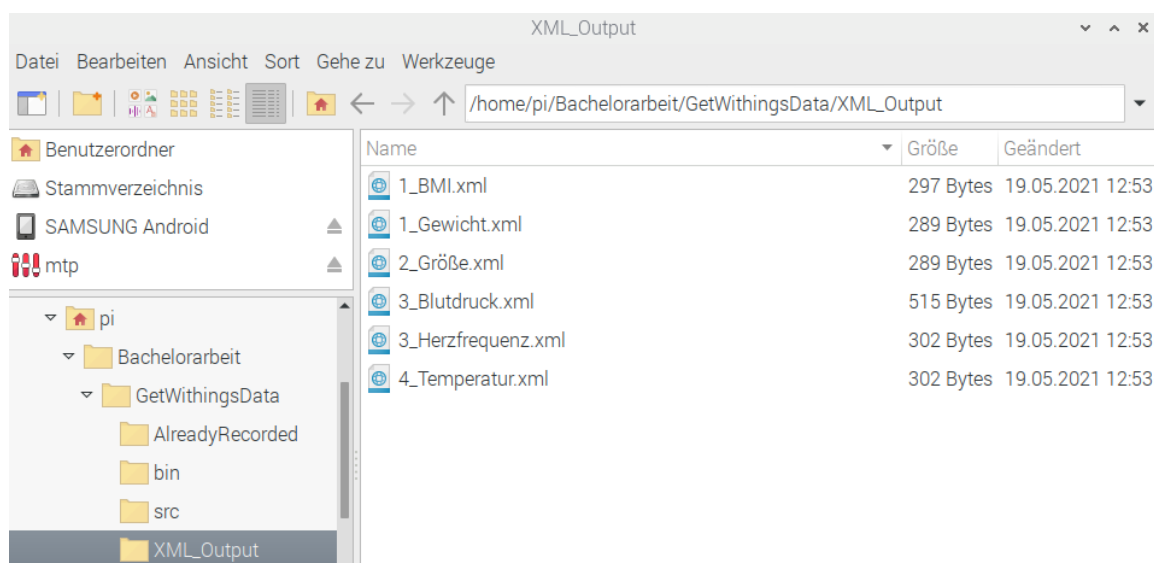


Abbildung 18: Beispielhafter Ausschnitt des Ordners „XML_Output“
Quelle: Eigene Aufnahme

Zudem wird die Unixzeit in menschenlesbare Form umgerechnet und der Messtyp wird nach den Zuordnungsregeln aus Tabelle 8 dargestellt. Das Dateiformat XML bietet den Vorteil, dass die Daten strukturiert dargestellt werden und somit eine einfache Weiterverarbeitung mit dem Kommunikationsserver Mirth-Connect möglich ist. Der Aufbau der XML-Datei ist in Abbildung 19 beispielhaft für eine Temperaturmessung dargestellt.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Temperaturmessung>
  - <Temperatur>
    <patient_id>1981552</patient_id>
    <measure_id>71</measure_id>
    <user>24702178</user>
    <firstname>Marius </firstname>
    <lastname>Müller </lastname>
    <time>2021-02-16 11:06:19</time>
    <y>37.19</y>
  </Temperatur>
</Temperaturmessung>
```

Abbildung 19: Beispielhafte XML-Datei mit den Messergebnissen einer Temperaturmessung
Quelle: Eigene Aufnahme

Einen Sonderfall bei diesem Prozess stellt die Blutdruckmessung dar. Bei der Messung mit dem BPM Core wird der systolische und diastolische Blutdruck ermittelt, jedoch werden beide Messergebnisse separat in der Withings-Datenbank ablegt. Da im FHIR-Standard der systolische und diastolische Blutdruck jedoch in einer Ressource angegeben werden, ist es sinnvoll beide Messergebnisse bereits in einer XML-Datei zu speichern. Aus diesem Grund wurde die Klasse *DiastolicBPBuffer* implementiert, in der der diastolische Blutdruck einer Messgruppe zwischengespeichert wird, bis der entsprechende systolische Blutdruck aufgenommen wurde, um dann beide Messergebnisse in einer XML-Datei zu speichern. Der Aufbau einer XML-Datei für die Blutdruckmessung ist in Abbildung 20 beispielhaft dargestellt.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Blutdruckmessung>
  - <Blutdruck_systolisch>
    <patient_id>1981552</patient_id>
    <measure_id>10</measure_id>
    <user>24702178</user>
    <firstname>Marius </firstname>
    <lastname>Müller </lastname>
    <time>2021-02-07 14:27:42</time>
    <y>137.0</y>
  </Blutdruck_systolisch>
  - <Blutdruck_diastolisch>
    <measure_id>9</measure_id>
    <user>24702178</user>
    <firstname>Marius </firstname>
    <lastname>Müller </lastname>
    <time>2021-02-07 14:27:42</time>
    <y>71.0</y>
  </Blutdruck_diastolisch>
</Blutdruckmessung>
```

Abbildung 20: Beispielhafte XML-Datei mit den Ergebnissen einer Blutdruckmessung
Quelle: Eigene Aufnahme

Der Quellcode befindet sich im Abgabeordner im Verzeichnis „*GetWithingsData*“. Zusätzlich wurde eine Informationsdatei „*README.txt*“ im Verzeichnis erstellt, die alle implementierten Klassen der Java-Applikation detaillierter beschreibt.

6.4 Upload der Daten auf FHIR-Testserver mit Mirth Connect

Nachdem eine neue Messung mit einem der Withings-Geräte durchgeführt und die in Kapitel 6.3 beschriebene Java-Applikation *GetWithingsData* ausgeführt wurde, befinden sich eine oder mehrere XML-Dateien im Verzeichnis *XML_Output*. Um diese XML-Dateien nun in Form einer FHIR-Ressource auf dem FHIR-Testserver abzulegen, wird der Kommunikationsserver Mirth-Connect auf dem Raspberry Pi installiert. Mit den folgenden Befehlen kann der Server gestartet werden:

```
./mcservice start
```

```
./mcserver
```

Da der Admin-Launcher nicht auf dem Raspberry Pi installierbar ist, wird dieser von einem Laptop aus gestartet. Da sich der Laptop und der Raspberry Pi im gleichen WLAN befinden, kann der Kommunikationsserver über die IP-Adresse des Pis und dem Port 8080 adressiert und konfiguriert werden (Abb. 21).

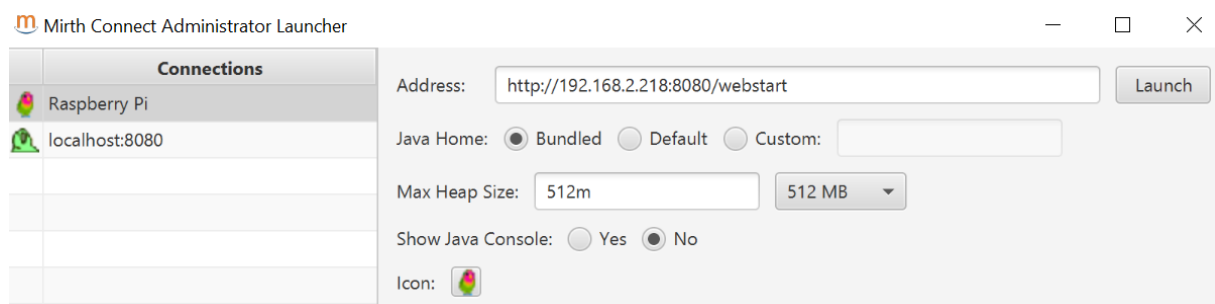


Abbildung 21: Admin Launcher des Kommunikationsservers Mirth Connect
Quelle: Eigene Aufnahme

6.4.1 Einrichten des Channels

Zur Verarbeitung der XML-Dateien wurde der Channel „XML_to_FHIR“ erstellt. Der Channel wurde als XML-Datei exportiert und befindet sich zusammen mit den Skripten zum Starten des Servers im Abgabeordner. „XML_to_FHIR“ besitzt einen Source-Connector des Typs *File-Reader* und überwacht das Verzeichnis *XML_Output* des Raspberry Pis (Abb. 18) in einem Intervall von 5 Sekunden. Wird eine neue XML-Datei im Verzeichnis abgelegt, wird diese vom Source Connector aufgenommen und anschließend gelöscht (Abb. 22).

Abbildung 22: Source Connector des Mirth Connect Channels „XML_to_FHIR“

Als Destination Connector wird ein FHIR-Sender eingesetzt, der mit Hilfe der in Kapitel 5.2 angesprochenen FHIR-Extension, installiert werden kann. Ein FHIR-Sender verhält sich ähnlich wie ein HTTP-Sender – er sendet also HTTP-Anfragen an einen externen FHIR-Server. Das Ziel ist es, eine FHIR Ressource des Typs *Observation* auf dem HAPI FHIR-Testserver zu erstellen. Somit soll nach Tabelle 3 ein POST-Request der Form

POST <http://hapi.fhir.org/baseR4/Observation>

an den FHIR-Server gesendet werden. Dies kann mit folgenden Settings des FHIR-Senders realisiert werden (Abb. 23):

Da in dieser Arbeit mit FHIR Version 4 gearbeitet wird, wird als Version *R4* und als URL <http://hapi.fhir.org/baseR4> angegeben. Als Interaction wird *create* auf eine Ressource des Typs *Observation* und als Path der oben genannte POST-Request gewählt.

Ein großer Vorteil des Kommunikationsservers Mirth Connect ist es, dass den Destination Connectoren Inbound- und Outbound-Templates zugefügt werden können (Abb. 24). Die Inbound-Templates werden dabei über die Variable *msg* angesprochen – die Outbound-Templates über die Variable *tmp*. Zur einfacheren Handhabung bietet Mirth Connect das Feature an, die Elemente der Templates per Drag-and-Drop auszuwählen.

Wird in den Settings des FHIR-Senders als Format „*Use Templates*“ ausgewählt, wird als Body des POST-Requests der Inhalt des Outbound-Templates, ergänzt mit ausgewählten Elementen des Inbound-Templates, ausgegeben. Da in dieser Arbeit eine FHIR-Ressource auf einem FHIR-Testserver erstellt werden soll, wurden als Outbound-Templates, die in Kapitel 5.4 vorgestellten Templates der FHIR-Ressourcen des Typs *Observation* für Vitalparameter verwendet. Um die mit den Withings-Geräten aufgenommenen Vitalparameter an die entsprechende Stelle des Outbound-Templates zu mappen, wurden die XML-Dateien als Inbound-Template eingesetzt.

Abbildung 23: Destination Connector des Mirth Connect Channels „XML_to_FHIR“
Quelle: Eigene Aufnahme

Abbildung 24: Message Templates für den Mirth-Connect Channel „XML_to_FHIR“
Quelle: Eigene Aufnahme

Damit die richtigen Informationen der XML-Dateien an die richtige Stelle im FHIR-Template gemappt werden, wurden mehrere Filter und Transformer erstellt, die im folgenden Kapitel vorgestellt werden.

6.4.2 Einrichten der Filter und Transformer

Zur Konvertierung der in der XML-Datei angegebenen Vitalparameter in den FHIR-Standard, wurden fünf Destination-Connectoren implementiert – einer für jeden Messtypen. Die Connectoren für die Messtypen Gewicht, BMI, Herzfrequenz und Temperatur besitzen jeweils einen Filter und drei Transformer. Die Filter überprüfen die Mess-IDs der XML-Dateien nach den in Tabelle 8 dargestellten Zuordnungsregeln. Destination 5 beispielsweise verarbeitet Temperaturmessungen und akzeptiert daher lediglich Nachrichten mit der Mess-ID '71' (Abb. 25). Wird also vom Source-Connector eine Temperaturmessung mit der ID '71' aufgenommen, wird diese nacheinander von den Filtern aller Destination Connectoren abgelehnt, bis sie schließlich vom Filter des Destination-Connectors 5 akzeptiert wird. Auf diese Weise ist garantiert, dass jede Messung nur einmal und an der richtigen Stelle auf dem FHIR-Testserver abgelegt wird.

Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0	Accept message if *msg["Temperatur"]([measure_id]).toString() equals '71'	Rule Builder

Rule \ Generated Script \

Behavior: Accept

Field: msg["Temperatur"]([measure_id]).toString()

Condition: ☐ Exists ☐ Not Exist ☒ Equals ☐ Not Equal ☐ Contains ☐ Not Contain

Values:

New Delete

Abbildung 25: Destination-Filter des Mirth-Connect Channels „XML_to_FHIR“
Quelle: Eigene Aufnahme

Wird eine Nachricht akzeptiert, wird diese von den drei Destination-Transformern verarbeitet. Der erste Transformer ist vom Typ *Message Builder* und mappt das Messergebnis, das in der XML-Datei enthalten ist, an die entsprechende Stelle im FHIR-Template (also an das Attribut *value* des Elements *valueQuantity*)(Abb. 26).

Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0		Message Builder
<input checked="" type="checkbox"/>	1		JavaScript
<input checked="" type="checkbox"/>	2		JavaScript

Step \ Generated Script \

Message Segment: tmp[valueQuantity][value]

Mapping: msg["Temperatur"]([y]).toString()

Default Value:

String Replacement:

New Delete

Abbildung 26: 1. Destination-Transformer (Message Builder) des Mirth Connect Channels „XML_to_FHIR“
Quelle: Eigene Aufnahme

Transformer 2 und 3 führen JavaScript-Code aus und mappen den in der XML-Datei angegebenen Messzeitpunkt (Abb. 27) und die Patienten-ID (Abb. 28) in das FHIR-Template. Dabei werden beide Elemente so verarbeitet, dass sie den syntaktischen Vorgaben des FHIR-Standards genügen.

Edit Channel - XML_to_FHIR - Destination 5 Transformer			
Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0		Message Builder
<input checked="" type="checkbox"/>	1		JavaScript
<input checked="" type="checkbox"/>	2		JavaScript

```
Step \ Generated Script \
1 var time = msg['Temperatur']['time'].toString();
2 var timeSplit = time.split(' ');
3
4
5 tmp['effectiveDateTime'] = validate(timeSplit[0]+'T'+timeSplit[1]+'Z', '', new Array());
```

Abbildung 27: 2. Destination-Transformer (JavaScript) des Mirth Connect Channels „XML_to_FHIR“
Quelle: Eigene Aufnahme

Edit Channel - XML_to_FHIR - Destination 5 Transformer			
Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0		Message Builder
<input checked="" type="checkbox"/>	1		JavaScript
<input checked="" type="checkbox"/>	2		JavaScript

```
Step \ Generated Script \
1 var pid = msg['Temperatur']['patient_id'].toString();
2
3 tmp['subject']['reference'] = validate('Patient/' + pid, '', new Array());
```

Abbildung 28: 3. Destination-Transformer (JavaScript) des Mirth Connect Channels „XML_to_FHIR“
Quelle: Eigene Aufnahme

7 Visualisierung

Nachdem die Messergebnisse auf dem FHIR-Server abgelegt wurden, sollen diese visualisiert werden. Dazu wurde die Web-Applikation „GetFHIRData“ entwickelt, die mit der REST-API des FHIR-Servers interagiert, die Vitalparameter abfragt und sie dann mit Hilfe von Tabellen und Diagrammen grafisch darstellt. Außerdem wurde ein Login-System implementiert, mit dem der Benutzer identifiziert werden kann. Auf diese Weise können die erfassten Messungen dynamisch dem gerade eingeloggten Benutzer zugeordnet werden. Die so aufgenommenen Daten werden in einer MySQL-Datenbank abgelegt, die drei Tabellen enthält: Die Tabelle *accounts*, in der alle Benutzer-Informationen der bisher angelegten Accounts gespeichert sind, die Tabelle *aktueller_user*, die die Informationen zum aktuell eingeloggten Benutzer bereithält und die Tabelle *observations*, die alle bisher aufgenommenen Messungen enthält. Im Folgenden werden alle Funktionalitäten der Web-Applikation. Screenshots aller Funktionalitäten der Web-Applikation sind in *Anhang I* zu finden. Der Quellcode befindet sich im Abgabeordner im Verzeichnis „GetFHIRData“ zusammen mit einer Informationsdatei „*README.txt*“, die alle implementierten Quellcode-Dateien detailliert beschreibt.

7.1 Das Login-System

Beim Start der Web-Applikation wird der Benutzer aufgefordert, sich mit seiner E-Mail-Adresse und dem von ihm gewählten Passwort anzumelden (Anhang I Abb. I). Hat der Benutzer noch keinen Account, kann dieser mit einem Klick auf den Button „Neues Konto erstellen“ angelegt werden. Hier wird der Benutzer aufgefordert Informationen zum Namen, Adresse, Geschlecht, Geburtsdatum und Telefonnummer anzugeben (Anhang I Abb. II).

Mit dem Erstellen des Accounts werden diese Informationen in der MySQL-Datenbank in der Tabelle *accounts* gespeichert. Anschließend wird ein Patient auf dem FHIR-Testserver angelegt. Dazu wird gemäß *Tabelle 4* ein POST-Request an die RESTful API des Servers gesendet.

```
POST http://hapi.fhir.org/baseR4/Patient
```

Als Body wird eine reduzierte Version des in Kapitel 2.2 erstellten Patienten-Templates verwendet, dass mit den angegebenen Benutzer-Informationen ergänzt wird. Im Header wird als Content-Type „*application/json*“ angegeben. Der Response-Body enthält die vom Server vergebene Ressourcen-ID, die extrahiert wird und ebenfalls in der Tabelle *accounts* gespeichert wird. Auf diese Weise kann jeder Account eindeutig einer Ressource des Typs *Patient* auf dem FHIR-Testserver zugeordnet werden.

Loggt sich ein Benutzer ein, werden seine Benutzer-Daten in die Tabelle *aktueller_user* geladen – beim Ausloggen werden diese aus der Tabelle entfernt. Damit alle Komponenten des im Rahmen dieser Arbeit erstellten Gesamtsystems die Informationen des momentan eingeloggten Users einsehen können, wurde eine API implementiert, die eben diese Informationen als Response-Body zurückgibt.

Ist aktuell ein Benutzer eingeloggt, antwortet die API auf

```
GET http://localhost/GetFHIRData/myapi/api.php
```

mit dem Response-Code 200 – anderenfalls mit Response-Code 204. Wird die Interaktion mit der API der Web-Applikation in das Java-Programm *GetWithingsData* integriert, ergibt sich ein lauffähiges Gesamtsystem:

Will ein Benutzer eine neue Messung durchführen, loggt er sich zuerst in die Web-Applikation ein und startet anschließend das Withings-Messgerät. Nachdem die Messung durchgeführt wurde, wird das Messergebnis von der *Health Mate App* in der Datenbank *Withings-WiScale* gespeichert, die vom Raspberry Pi per Cronjob alle 60 Sekunden über die Android Debug Bridge in den lokalen Speicher des Pis geladen wird. Anschließend wird die Java-Applikation *GetWithingsData* gestartet, die erst einmal überprüft, ob aktuell ein Benutzer eingeloggt ist. Sofern dies der Fall ist, werden alle neuen Messergebnisse und die von der API erhaltenen FHIR-Patienten-ID in Form einer XML-Datei im Verzeichnis *XML_Output* gespeichert. Der Channel *XML_to_FHIR* des Kommunikationsservers Mirth-Connect überprüft das Verzeichnis alle 5 Sekunden auf neue XML-Dateien, konvertiert diese in eine FHIR-Ressource des Typs Observation und lädt sie auf den FHIR-Testserver. Dadurch, dass in der XML-Datei die Patienten-ID des momentan angemeldeten Benutzers dokumentiert ist, wird jede Messung vom Mirth-Connect dem richtigen Patienten auf dem FHIR-Server zugeordnet.

7.2 Darstellung der Messergebnisse

Loggt sich ein Benutzer mit seiner E-Mail-Adresse und seinem Passwort ein, wird er auf die Startseite der Web-Applikation weitergeleitet. Hier erscheint ihm ein Begrüßungstext, in dem er über die Inhalte der Abschlussarbeit und die Funktionalitäten der Web-Applikation aufgeklärt wird (Anhang I Abb. III). Im Header der Website stehen dem Benutzer folgende Navigationsmöglichkeiten zur Verfügung:

Mein Konto

Hier werden die vom Benutzer angegebenen Informationen angezeigt (Anhang I Abb. IV).

Database

Beim Öffnen dieser Seite wird zuerst ein GET-Request an den FHIR-Server gesendet, mit dem alle bisher aufgenommenen Observations des aktuell eingeloggten Benutzers abgefragt werden. Da auf dem FHIR-Testserver Paging implementiert wurde und somit immer nur 20 Observations pro Seite angezeigt werden, muss der Parameter *count* in die URL integriert werden. *Count* gibt die Menge der vom Server abgerufenen Ressourcen an und wird gleich der Anzahl aller Observations des aktuell eingeloggten Benutzers gesetzt, damit alle Ergebnisse zurückgegeben werden.

```
GET http://hapi.fhir.org/baseR4/Observation/?patient=[pid]&_count=[number_of_observations]
```

Anschließend wird eine Verbindung zur MySQL-Datenbank aufgebaut und die Ergebnisse aller Observations werden in der Tabelle *observations* abgelegt. Anschließend werden mit dem SQL-Statement

```
SELECT * FROM observations WHERE patient=[pid] ORDER BY effective_date ASC
```

alle Observations des aktuell eingeloggten Patienten abgefragt und nach dem Messzeitpunkt sortiert. Diese Informationen werden für den Benutzer in Form einer Tabelle angezeigt. Ein beispielhafter Ausschnitt dieser Tabelle ist im Anhang I Abb. V zu finden.

An dieser Stelle sei noch einmal darauf hingewiesen, dass die Abschlussarbeit mit dem Ziel erstellt wurde, einen möglichst hohen Lerneffekt zu erzielen und einen möglichst tiefen Einblick in viele verschiedene Teilgebiete im Bereich der Medizintechnik und Medizininformatik zu erlangen. Der Response-Body des GET-Requests hätte ebenso beispielsweise in einer Session-Variable gespeichert und dann in Form einer Tabelle dargestellt werden können. Trotzdem wurde sich dazu entschieden eine Datenbank zu implementieren, da so erste Erfahrungen mit dem Umgang mit Patientendaten im Zusammenhang mit SQL gemacht werden konnten.

Ergebnisse

Zur Visualisierung der Messergebnisse wurde *Highcharts*, eine Softwarebibliothek für Diagramme, verwendet [55]. Die Daten wurden mit Hilfe des im vorherigen Abschnitt „Database“ genannten SQL-Statements abgefragt. Dabei wurden auf der x-Achse der Messzeitpunkt und auf der y-Achse das entsprechende Messergebnis aufgetragen. Im Anhang befinden sich beispielhafte Ausschnitte aus der Darstellung von Temperatur- Herzfrequenz-, Körpergewicht, BMI und Blutdruckmessungen. (Anhang I Abb. VI-X).

Filter Optionen

Unter dem Reiter *Filter Optionen* hat der Benutzer die Möglichkeit die Messergebnisse zu filtern (Abb. 29). Mithilfe der Radio-Buttons kann zwischen der Anzeige aller Werte, des Mittelwerts, des höchsten und des niedrigsten Werts gewechselt werden. Im Dropdown-Menü *Messtyp* wird die Art der Messung ausgewählt – also Blutdruck-, Herzfrequenz-, Gewicht-, BMI-, oder Temperaturmessung. Durch die im Dropdown-Menü *Operation* ausgewählte Operation, wird festgelegt, auf welche Art und Weise die Messwerte gefiltert werden sollen. Hier hat der Benutzer die Wahl, ob die angezeigten Ergebnisse größer, größer/gleich, kleiner oder kleiner/gleich des im Eingabefeld *Wert* angegebenen Werts sein sollen. Mit einem Klick auf den Button *search* wird die Suche gestartet.

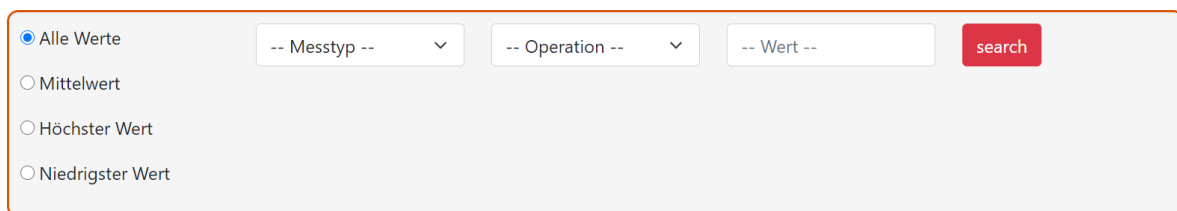


Abbildung 29: Abbildung der Filter Optionen, die dem Benutzer der Web-Applikation „GetFHIRData“ zur Verfügung stehen
Quelle: Eigene Aufnahme

Sollen beispielsweise alle Fieberwerte angezeigt werden, könnte man dies über die Filteroptionen *Messtyp: Temperatur; Operation: größer als; Wert: 38* realisieren (Anhang I Abb. XI). Darstellungen der Anzeige von Minimal-, Maximal- und Mittelwert sind im Anhang in Abb. XII-XIV zu finden.

Betätigt der Benutzer den Search-Button, wird eine Anfrage des Typs *search* gemäß *Tabelle 4* an die RESTful API des FHIR-Servers gesendet. Als Parameter werden die ausgewählten Filterkriterien übergeben. Bei der oben genannten Abfrage der Fieberwerte, würde folgender GET-Request gesendet werden:

```
GET http://hapi.fhir.org/baseR4/Observation/?patient=[pid]&code=8310-5&value-quantity=gt38
```

Der Response-Body wird mit Hilfe der Softwarebibliothek *Highcharts* als Diagramm dargestellt.

8 Fazit

8.1 Zusammenfassung der Ergebnisse

Das Ziel der Arbeit war es, die Grundlagen des Datenaustauschs im Gesundheitswesen mit besonderem Fokus auf dem FHIR-Standard kennenzulernen und dabei einen möglichst großen Lerneffekt zu erzielen. Dazu wurde die Entwicklung von Standards und Terminologien ausführlich erarbeitet, wodurch die Notwendigkeit der Standardisierung von Schnittstellen zwischen verschiedenen Systemen im Gesundheitswesen verdeutlicht wurde. Es wurde ersichtlich von welcher großer Bedeutung eine semantische und syntaktische Interoperabilität ist und dass ein Kommunikationsstandard unabdingbar ist, um eine digitale Infrastruktur im Gesundheitswesen aufzubauen.

Als Lösung des Interoperabilitäts-Problems wurde der von HL7 ins Leben gerufene Standard FHIR („Fast Healthcare Interoperability Resources“) vorgestellt. Dieser wurde ausführlich und von Grund auf beschrieben, sodass auch Leser mit wenigen Vorkenntnissen die Chance hatten, die Grundkonzepte von FHIR zu verstehen. Es wurde herausgearbeitet, dass die Nachfrage danach, Patientendaten einrichtungs- und fachrichtungsübergreifend zu kommunizieren immer weiter steigt und dass der Trend dabei von Offline zu Online, von Desktop zu Tablet und von Software zu App geht. Es wurde aufgezeigt, dass mit FHIR versucht wurde einen agilen und einfach zu implementierenden Datenstandard ins Leben zu rufen, der Interoperabilität „out-of-the-box“ bereitstellt. Welches hohe Ansehen FHIR schon heute genießt, wurde durch die Anwendungsbeispiele „Medizinische Informationsobjekte“ und „Digitale Gesundheits-anwendungen“ verdeutlicht, die bei Implementierung beide auf FHIR setzen.

Um die Anwendung von FHIR zu demonstrieren, wurde im praktischen Teil der Arbeit eine prototypische Digitale Gesundheitsanwendung entwickelt, mit der die Erfassung der Vitalparameter Blutdruck, Herzfrequenz, Körpergewicht, BMI und Körpertemperatur mit Consumer-Geräten umgesetzt wurde. Die so erfassten Daten wurden jedoch nicht in der proprietären Cloud des Herstellers der Consumer-Geräte, sondern auf einem frei gewählten FHIR-Server in Form einer FHIR-Ressource gespeichert. Auf diese Weise konnte erste Erfahrung mit dem Umgang und der Kommunikation von Gesundheitsdaten mit FHIR gesammelt werden.

Die erfassten Vitalparameter wurden von einem Smartphone, das über Bluetooth mit den Consumer-Geräten verbunden ist, auf einen Raspberry Pi geladen, wo sie mit Hilfe eines Kommunikationsservers in den FHIR-Standard konvertiert und anschließend auf einem FHIR-Testserver gespeichert wurden. Durch die Wahl des Kommunikationsservers Mirth Connect konnte ein erster Kontakt mit der Arbeit mit Kommunikationsservern und einem Tool gemacht werden, das auch im professionellen Umfeld eingesetzt wird.

Nachdem die Messergebnisse auf dem FHIR-Server abgelegt wurden, sollten diese visualisiert werden. Dazu wurde die Web-Applikation „GetFHIRData“ entwickelt, die mit der REST-API des FHIR-Servers interagiert, die Vitalparameter abfragt und sie dann mit Hilfe von Tabellen und Diagrammen grafisch aufbereitet. Die Web-Applikation wurde mit einem Login-System ausgestattet, sodass die erfassten Vitalparameter dynamisch dem gerade eingeloggten Benutzer zugeordnet werden konnten.

Außerdem wurde zusätzlich zur Dokumentation der praktischen Arbeit ein Kurzfilm angefertigt, der den prototypischen Ablauf einer Messung inklusive der technischen Aspekte des Abgreifens der Daten, deren Konvertierung und Ablage auf dem FHIR-Testserver, sowie das Retrieval der Daten darstellt.

8.2 Ausblick

Abschließend kann die Abschlussarbeit mit dem Titel „FHIR in Digitalen Gesundheitsanwendungen – Automatisierte Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Testserver“ als voller Erfolg gewertet werden. Der Lerneffekt, der beim Verfassen der Arbeit erzielt wurde, war enorm. Dadurch dass so viele Facetten des Datenaustauschs im Gesundheitswesen beleuchtet wurden, konnte sich ein erster Überblick über ein sehr komplexes Themengebiet verschafft werden. Außerdem konnten erste Erfahrungen in vielen Gebieten der Medizintechnik und Medizininformatik gesammelt werden - sei es im Einsatz von Consumer-Geräten zur Erfassung von Vitalparametern, im Umgang mit Kommunikationsservern oder im Erstellen von Web-Applikation, um Ergebnisse medizinischer Untersuchungen darzustellen. Durch den Schriftverkehr mit dem Technischen Komitee FHIR HL7 Deutschland (Anhang II, Schriftverkehr mit HL7 Deutschland) konnten zudem erste Kontakte in der FHIR-Szene geknüpft werden.

Auch der im Rahmen dieser Arbeit erstellte Prototyp zur automatischen Erfassung, Ablage und Retrieval von Vitaldaten kann als Erfolg gewertet werden. Das System konnte mit allen in der Einleitung spezifizierten Erwartungen umgesetzt werden - darüber hinaus konnte sogar eine Web-Applikation zur Visualisierung implementiert werden. Auch wenn am Ende kein verkaufsfertiges Produkt entstanden ist (was auch nicht das Ziel der Arbeit war) können die hier beschriebenen Features dennoch als Grundlage für weiterführende Arbeiten im Fachbereich Informatik der Hochschule Trier dienen. Sofern ein hochschuleigener FHIR-Server eingerichtet wird, können die vorgestellten Funktionalitäten genutzt werden, um eigenständige Projekte mit Einbindung des FHIR-Servers der Hochschule zu entwickeln, die unabhängig von externen Anbietern sind.

Da die Kommunikation der Gesundheitsdaten mit Hilfe des Open-Source Kommunikationsservers Mirth Connect umgesetzt wurde und auch die FHIR-Spezifikation frei zugänglich ist, kann diese Arbeit auch von Hochschul-fremden Personen verwendet werden, um eigene FHIR-Schnittstellen zu erzeugen.

„Per App ins digitale Krankenhaus [1]“

Trotz der positiven Entwicklung im Bereich der digitalen Gesundheitsversorgung, scheint das *digitale Krankenhaus*, vor allem in Deutschland, noch immer Zukunftsmusik zu sein. Doch auch wenn die digitale Infrastruktur im Gesundheitswesen noch in den Kinderschuhen steckt, macht der Fortschritt in diesem Bereich Mut für die Zukunft. Wir leben in einer aufregenden Zeit, in der die Digitalisierung immer mehr Einzug in die verschiedensten Bereiche des Lebens erhält. Die Möglichkeiten, die Datenstandards wie FHIR für die Digitalisierung des Gesundheitswesens bieten, sind noch lange nicht ausgeschöpft und ich hoffe, dass diese Arbeit ihren Teil zum Fortschritt in diesem Gebiet beitragen kann.

Quellen

- [1] Halpick, Silke (2019): *Per App ins Krankenhaus*. Abgerufen am 03.03.2021, von <https://www.lr-online.de/lausitz/cottbus/digitalisierung-per-app-ins-krankenhaus-38330052.html>
- [2] HL 7 International: *About HL7*. Abgerufen am 25.01.2021, von <https://www.hl7.org/about/index.cfm?ref=common>
- [3] Dipl. – Ing. Luber, Stefan (2018): *Was ist das OSI-Modell?* Online im Internet bei: IP Insider – mehr. Abgerufen am 25.01.2021, von <https://www.ip-insider.de/was-ist-das-osi-modell-a-605831/>
- [4] HL7 Deutschland: *Warum FHIR?* Abgerufen am 25.01.2021, von <https://hl7.de/themen/hl7-fhir-mobile-kommunikation-und-mehr/warum-fhir/>
- [5] Kassenärztliche Bundesvereinigung – MIO (o. D.): *Was ist FHIR?* Abfrage am 25.01.2021, von <https://mio.kbv.de/pages/viewpage.action?pageId=33357957>
- [6] HL 7 Deutschland: *HL7 – Die Familie von Kommunikationsstandards für das Gesundheitswesen*. Abgerufen am 29.01.2021, von <https://hl7.de/themen/hl7-die-familie-von-kommunikationsstandards/>
- [7] HL7 Deutschland: *HL7 v2.x Nachrichten – IT-Kommunikation im Krankenhaus*. Abgerufen am 29.01.2021, von <https://hl7.de/themen/hl7-v2-x-nachrichten/>
- [8] HL7 Deutschland (o. D.): *HL7 RIM – Ein umfassendes Referenzinformationsmodell für alle Anwendungen*. Abgerufen am 30.01.2021, von <https://hl7.de/themen/hl7-v2-x-nachrichten/>
- [9] HL7 Deutschland: *HL7 CDA – Clinical Document Architecture für standardisierte medizinische Dokumente*. Abgerufen am 30.01.2021, von <https://hl7.de/themen/hl7-cda-clinical-document-architecture/>
- [10] HL7 Deutschland (o. D.): *HL7 Deutschland*. Abgerufen am 25.01.2021, von <https://hl7.de/ueber-hl7/hl7-deutschland/>
- [11] Palaria, Amrit (2020): *Introduction to HL7 FHIR data model*. Digitale Aufzeichnung des HL7 India FHIR Connectathon 2020. Abgerufen am 02.02.2021, von <https://www.youtube.com/watch?v=oSl6g3uy3mY>
- [12] Palaria, Amrit (2020): *Introduction to FHIR API*. Digitale Aufzeichnung des HL7 India FHIR Connectathon 2020. Abgerufen am 02.02.2021, von <https://www.youtube.com/watch?v=1UhtBV0DW5I>
- [13] Bundesinstitut für Arzneimittel und Medizinprodukte: *Von der ICD zur ICD-10*. Abgerufen am 05.02.2021, von <https://www.dimdi.de/dynamic/de/klassifikationen/icd/icd-10-who/historie/ilcd-bis-icd-10/>
- [14] Bundesministerium für Gesundheit: *Was bedeutet ICD?* Abgerufen am 05.02.2021, von <https://gesund.bund.de/service/was-ist-der-icd-code#icd-verschluesselung>
- [15] Laimingas/ Klein, Manfred (2020): *Was ist SNOMED CT?* Online im Internet bei: Healthcare computing – Glossar. Abgerufen am 05.02.2021, <https://www.healthcare-computing.de/was-ist-snomed-ct-a-912647/#:~:text=Die%20Abk%C3%BCrzung%20SNOMED%20CT%20steht,und%20wichtigsten%20medizinischen%20Terminologien%20weltweit.>
- [16] Kassenärztliche Bundesvereinigung – MIO: *Was ist SNOMED CT?* Abgerufen am 05.02.2021, von <https://mio.kbv.de/pages/viewpage.action?pageId=33358841>

-
- [17] Deutsches Ärzteblatt (2021): *BfArM stellt medizinische Terminologie SNOMED CT in Deutschland zur Verfügung*. Abgerufen am 06.02.2021, von <https://www.aerzteblatt.de/nachrichten/119856/BfArM-stellt-medizinische-Terminologie-SNOMED-CT-in-Deutschland-zur-Verfuegung>
- [18] DocCheck Flexikon: *Arden-Syntax*. Abgerufen am 10.02.2021, von <https://flexikon.doccheck.com/de/Arden-Syntax>
- [19] HL7 International: *Arden Syntax V2.7*. Abgerufen am 10.02.2021, von https://www.hl7.org/implement/standards/product_brief.cfm?product_id=2
- [20] Deutsches Institut für Medizinische Dokumentation und Information (2009): *Logical Observation Identifiers Names and Codes – Benutzerhandbuch*. Abgerufen am 13.02.2021, von <https://www.dimdi.de/static/.downloads/deutsch/loinc-german-user-guide.pdf>
- [21] Deutsches Ärzteblatt (2003): *Labor- und Vitaldatenkommunikation: Standardisierung durch LOINC*. Abgerufen am 13.02.2021, von <https://www.aerzteblatt.de/archiv/36045/Labor-und-Vitaldatenkommunikation-Standardisierung-durch-LOINC>
- [22] Bundesinstitut für Arzneimittel und Medizinprodukte: *LOINC und RELMA*. Abgerufen am 13.02.2021, von <https://www.dimdi.de/dynamic/de/klassifikationen/weitere-klassifikationen-und-standards/loinc-relma/>
- [23] Integrating the Healthcare Enterprise: *About IHE*. Abgerufen am 14.02.2021, von https://www.ihe.net/about_ihe/
- [24] Ihls, Alexander (2017): *Patientenakten auf Basis von IHE-Profilen*. In: EHEALTHCOM, Ausgabe 6/17, S. 2-4. Abgerufen am 14.02.2021, von https://www.intersystems.com/de/wp-content/uploads/sites/4/2018/10/EHC_6_2017_Standards_IHE-Akten_S47.pdf
- [25] HL7 Deutschland: *HL7 CDA – Clinical Document Architecture für standardisierte medizinische Dokumente*. Abgerufen am 16.02.2021, von <https://hl7.de/themen/hl7-cda-clinical-document-architecture/>
- [26] Interoperability of electronic health records: *What is HL7 CDA?* Abgerufen am 16.02.2021, von <http://iehr.eu/knowledge/what-is-hl7-cda/>
- [27] HL7 Deutschland: *HL7 RIM – Ein umfassendes Referenzinformationsmodell für alle Anwendungen*. Abgerufen am 19.02.2021, von <https://hl7.de/themen/hl7-v3-rim-das-referenzinformationsmodell/>
- [28] Brull, Rob (2011): *HL7 v3 RIM: Is It Really that Intimidating?* Online im Internet bei: Health Standards – Blog. Abgerufen am 19.02.2021, von <https://healthstandards.com/blog/2011/05/31/hl7-v3-rim-is-it-really-that-intimidating/>
- [29] Corepoint Health (2010): *The HL7 Evolution – Comparing HL7 Version 2 to Version 3, Including a History of Version 2*. Abgerufen am 21.02.2021, von <https://3ivzq3425pi8198k6t3v761a-wpengine.netdna-ssl.com/wp-content/uploads/hl7-v2-v3-evolution.pdf>
- [30] Mederith, John (2017): *Why did HL7 Version 3 Fail?* Abgerufen am 21.02.2021, von <https://www.archetextur.es/why-did-hl7-version-3-fail/>
- [31] Inteliware software development: *The HL7 Games Catching FHIR*. Abgerufen am 22.02.2021, von <https://www.inteliware.com/hl7-games-catching-fhir/>
- [32] HL7 FHIR Release 4: *Introducing HL7 FHIR*. Abgerufen am 22.02.2021, von <https://hl7.org/fhir/summary.html>

-
- [33] HL7 FHIR FHIR Specification: *Publication (Version) History*. Abgerufen am 22.02.2021, von <http://hl7.org/fhir/directory.html>
- [34] HL7 FHIR Release 4 – *Defining Extensions*. Abgerufen am 27.02.2021, von <https://www.hl7.org/fhir/defining-extensions.html>
- [35] HL7 FHIR Release 4 – *FHIR Exchange Module*. Abgerufen am 27.02.2021, von <https://www.hl7.org/fhir/exchange-module.html>
- [36] Dipl. – Ing. Drilling, Thomas (2017): *Konzept, Aufbau und Funktionsweise von REST*. Online im Internet bei: DEV Insider – Solution Stack. Abgerufen am 28.02.2021, von <https://www.dev-insider.de/konzept-aufbau-und-funktionsweise-von-rest-a-603152/>
- [37] Dipl. – Ing. Luber, Stefan (2017): *Was ist eine API?* Online im Internet bei: DEV Insider – Specials. Abgerufen am 02.02.021, von <https://www.dev-insider.de/was-ist-eine-api-a-583923/>
- [38] HL7 FHIR Release 4: *Profiling FHIR*. Abgerufen am 03.03.2021, von <https://www.hl7.org/fhir/profiling.html>
- [39] Kassenärztliche Bundesvereinigung: *Erklärung FHIR*. Abgerufen am 29.01.2021, von <https://mio.kbv.de/pages/viewpage.action?pagelId=30146985>
- [40] Kassenärztliche Bundesvereinigung: *Erklärung EPA*. Abgerufen am 29.01.2021, von <https://mio.kbv.de/pages/viewpage.action?pagelId=53248149>
- [41] Kassenärztliche Bundesvereinigung: *MIO – Medizinische Informationsobjekte*. Abgerufen am 29.01.2021, von <https://mio.kbv.de/site/mio#expand-WassindmedizinischeInformationsobjekteMIOS>
- [42] Kassenärztliche Bundesvereinigung: *KBV-Basis-Profile*. Abgerufen am 29.01.2021, von <https://mio.kbv.de/display/BASE1X0>
- [43] Kassenärztliche Bundesvereinigung: *Impfpass Übersicht*. Abgerufen am 03.03.2021, von <https://mio.kbv.de/display/IM>
- [44] Kassenärztliche Bundesvereinigung: *Zahnärztliches Bonusheft 1.0.0*. Abgerufen am 03.03.2021, von <https://mio.kbv.de/display/ZB1x0>
- [45] Kassenärztliche Bundesvereinigung: *Mutterpass 1.0.0*. Abgerufen am 03.03.2021, von <https://mio.kbv.de/display/MP1X0>
- [46] Kassenärztliche Bundesvereinigung: *U-Heft 1.0.0*. Abgerufen am 03.03.2021, von <https://mio.kbv.de/display/MP1X0>
- [47] Bundesinstitut für Arzneimittel und Medizinprodukte: *Digitale Gesundheitsanwendungen (DiGa)*. Abgerufen am 03.03.2021, von https://www.bfarm.de/DE/Medizinprodukte/DVG/_node.html
- [48] Bundesinstitut für Arzneimittel und Medizinprodukte: *Das Fast Track Verfahren für digitale Gesundheitsanwendungen (DiGa) nach §139e SGB V – Ein Leitfaden für Hersteller, Leistungserbringer und Anwender*. Abgerufen am 06.03.2021, von https://www.bfarm.de/SharedDocs/Downloads/DE/Service/Beratungsverfahren/DiGA-Leitfaden.pdf?__blob=publicationFile&v=1
- [49] Withings (2021): *Startseite*. Abgerufen am 12.03.2021, von <https://www.withings.com/de/de/>
- [50] Raspberry Pi Foundation: *About Us*. Abgerufen am 12.03.2021 von, <https://www.raspberrypi.org/about/>
- [51] NextGen Connect (2020): *User Guide – for version 3.9*. Abgerufen am 12.03.2021, von <https://www.nextgen.com/-/media/files/nextgen-connect/nextgen-connect-39-user-guide.pdf>

-
- [52] Mirth: *FHIR Connect Extension (3.9)*. Abgerufen am 12.03.2021, von <http://www.mirthcorp.com/community/wiki/pages/viewpage.action?pageId=38798954>
- [53] Verclas, Stephan; Linnhoff-Popien, Claudia (Hrsg.) (2012): *Smart Mobile Apps: Mit Business-Apps ins Zeitalter mobile Geschäftsprozesse*. Berlin Heidelberg, Springer. S. 231 ff.
- [54] Google Developers: *Android Debug Bridge (adb)*. Abgerufen am 13.03.2021, von <https://developer.android.com/studio/command-line/adb>
- [55] Online im Internet: *Highcharts*. Abgerufen am 13.03.2021, von <https://www.highcharts.com/>
- [56] HL FHIR Release 4: *Observation-vitalsigns*. Abgerufen am 27.02.2021, von <https://www.hl7.org/fhir/observation-vitalsigns.html>
- [57] Carl-Thiem-Klinikum Cottbus (2020): *Per Klick ins Krankenhaus -CTK eröffnet Express Check-In*. Abgerufen vom 03.03.2021, von <https://www.ctlk.de/klinikum/aktuelles-veranstaltungen/news-presse-meldungen/details/per-klick-ins-krankenhaus-ctlk-eroeffnet-express-check-in.html>
- [58] McKinsey & Company (2020): *eHealth Monitor 2020 – Deutschlands Weg in die digitale Gesundheitsversorgung – Status quo und Perspektiven*. Abgerufen am 03.03.2021, von <https://www.mckinsey.de/~media/mckinsey/locations/europe%20and%20middle%20east/deutschland/news/presse/2020/2020-11-12%20ehealth%20monitor/ehealth%20monitor%202020.pdf>
- [59] Bundesministerium für Gesundheit (2020): *E-Health – Digitalisierung im Gesundheitswesen*. Abgerufen a 03.03.2021, von <https://www.bundesgesundheitsministerium.de/e-health-initiative.html>
- [60] Lorenz, Wolf-Dietrich (2020): *HL7 FHIR – mobile Kommunikation und mehr*. Online im Internet bei: Krankenhaus-IT Online Journal – News/Newsarchiv. Abgerufen am 04.03.2021, von <http://www.medizin-edv.de/modules/AMS/article.php?storyid=4946>
- [61] *HAPI FHIR*. Abgerufen am 13.03.2021, von <http://hapi.fhir.org/>
- [62] *Magisk Manager*. Abgerufen am 13.03.2021, von <https://magiskmanager.com/>
- [63] Thieme Pflegepädagogik: *Vitalparameter und Körpertemperatur*. Abgerufen am 03.03.2021, von <https://www.thieme.de/de/pflegepaedagogik/16-vitalparameter-und-koerpertemperatur-68656.htm>
- [64] Mobyen Uddin Ahmed (2017), *An Intelligent Healthcare Serviceto Monitor Vital Signs in Daily Life – A Case Study on Health-IoT*, *International Journal of Engineering Research and Applications* 7
- [65] K. Swaroop et al. (2019), *A health monitoring system for vital signs using IoT*, *Internet of Things* Volume 5
- [66] M. Al-Shaher, N. Al-Khafaji (2017), *E-healthcare system to monitor vital signs* International Conference on Electronics, Computers and Artificial Intelligence (ECAI)
- [67] B. Franz, A. Schuler, O.Krauss (2015), *Applying FHIR in an Integrated Health Monitoring System*
- [68] S. El-Shappagh et al. (2019), *A mobile health monitoring-and-treatment system based on integration of the SSN sensor ontology and the HL7 FHIR standard*, *BMC Medical Informatics and Decision Making*
- [69] A, El Murabet et al. (2018), *Ambient Assisted living system’s models and architectures: A survey of the state of the art*, *Journal of King Saud University – Computer and Information Science*

Index

HL7	Health Level 7
FHIR	Fast Healthcare Interoperability Resources
CDA	Clinical Document Architecture
RIM	Reference Information Model
IHE	Integrating the Healthcare Enterprise
ICD	International Statistical Classification of Diseases and Related Health Problems
SNOMED	Systematized Nomenclature of Medicine
LOINC	Logical Observation Identifiers Names and Codes
BfArM	Bundesinstitut für Arzneimittel und Medizinprodukte
DiGA	Digitale Gesundheitsanwendungen
MIO	Medizinische Informationsobjekte
ePA	Elektronische Patientenakte
KBV	Kassenärztliche Bundesvereinigung
XML	Extensible Markup Language
JSON	JavaScript Object Notation
HTML	Hyper Text Markup Language
XHTML	Extensible Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
REST	Representational State Transfer
API	Application Programming Interface

Anhang

Anhang I: Abbildungen	IX
Anhang II: Organisatorisches.....	XXIII
Pflichtenheft und Projektplan	XXIII
Arbeitstitel	XXIII
Pflichtenheft.....	XXIII
Meilensteine & Projektplan	XXV
Fortschrittsberichte	XXVIII
Fortschrittsbericht vom 12.02.2021	XXVIII
Fortschrittsbericht vom 16.03.2021	XXIX
Fortschrittsbericht vom 20.04.2021	XXX
Besprechungsprotokolle:.....	XXXI
Besprechung vom 21.01.2021	XXXI
Besprechung vom 02.03.2021	XXXI
Besprechung vom 28.03.2021	XXXII
Schriftverkehr mit HL7-Deutschland	XXXIII
Erklärung des Kandidaten	XXXIV

Anhang I: Abbildungen

Abb. I: Login-Seite der Web-Applikation „GetFHIRData“	X
Abb. II: Register-Seite der Web-Applikation „GetFHIRData“	X
Abb. III: Startseite der Webapplikation „GetFHIRData“	XI
Abb. IV: Seite „Mein Konto“ der Webapplikation „GetFHIRData“	XII
Abb. V: Datenbank der Webapplikation „GetFHIRData“	XIII
Abb. VI: Darstellung der Temperatur-Messwerte in der Web-Applikation „GetFHIRData“ ...	XIV
Abb. VII: Darstellung der Herzfrequenz-Messwerte in der Web-Applikation „GetFHIRData“	XV
Abb. VIII: Darstellung der Körpergewicht-Messwerte in der Web-Applikation „GetFHIRData“	XVI
Abb. IX: Darstellung der BMI-esswerte in der Web-Applikation „GetFHIRData“	XVII
Abb. X: Darstellung der Blutdruck-Messwerte in der Web-Applikation „GetFHIRData“	XVIII
Abb. XI: Filter Optionen für FHIR-Ressourcen am Beispiel der Temperatur-Messwerte (größer als 38)	XIX
Abb. XII: Beispielhafte Mittelwertberechnung der Herzfrequenz-Messwerte mit Hilfe der Filter Optionen.....	XX
Abb. XIII: Beispielhafte Berechnung des Maximalwertes der Herzfrequenz-Messwerte mit Hilfe der Filter-Optionen.....	XXI
Abb. XIV: Beispielhafte Berechnung des Minimalwertes der Herzfrequenz-Messwerte mit Hilfe der Filter-Optionen.....	XXII

The screenshot shows a login interface for 'FHIR-DATABASE'. It features a white form box on a light gray background. Inside the box, there are two input fields: 'Email' and 'Password'. Below the 'Password' field is a blue button labeled 'Anmelden'. At the bottom of the form box is a green button labeled 'Neues Konto erstellen'.

Abb. I: Login-Seite der Web-Applikation „GetFHIRData“
Quelle: Eigene Aufnahme

The screenshot shows a registration interface titled 'Registrieren'. It features a white form box on a light gray background. The form contains several input fields: 'E-Mail', 'Password', 'Vorname', 'Nachname', 'Geschlecht' (a dropdown menu), 'Geburtsdatum' (a date picker), 'Telefon Nummer' (with a country code dropdown), 'Straße', 'Hausnummer', 'Stadt', 'Postleitzahl', and 'Land' (a dropdown menu). At the bottom of the form box is a blue button labeled 'Account erstellen'. Below the form box is a green button labeled 'Ich habe bereits einen Account'.

Abb. II: Register-Seite der Web-Applikation „GetFHIRData“
Quelle: Eigene Aufnahme



Home

Mein Konto

Database

Ergebnisse

Filter Optionen

Abmelden

Herzlich Willkommen Marius!

Dies ist die Startseite der Web-Applikation *GetFHIRData*, die zur Visualisierung meiner Bachelorarbeit dient.

Mein Name ist Marius Müller und ich studiere momentan Medizintechnik an der Hochschule Trier. Anfang des Jahres 2021 habe ich meine Bachelorarbeit zum Thema "*FHIR in Digitalen Gesundheitsanwendungen - Automatisierte Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Testserver*" verfasst. Ziel der Arbeit war es, Vitaldaten mit Consumer-Geräten der Firma Withings zu erfassen und diese automatisiert auf einem FHIR-Testserver abzulegen. Dazu mussten die Messergebnisse aus der proprietären Darstellung der Withings Android-App abgegriffen und mit Hilfe des Kommunikationsservers Mirth-Connect ins FHIR/JSON-Format konvertiert werden. Die so erstellten FHIR-Ressourcen wurden auf einem [FHIR-Testserver](#) abgelegt.

Zur Visualisierung der Messergebnisse wurde diese Web-Applikation erstellt. Mit dem Registrierungsvorgang wurde automatisch ein Patient mit den von Ihnen angegebenen Patienten-Daten erstellt, den Sie [hier](#) einsehen können. Solange Sie in dieser Web-Applikation eingeloggt sind, werden alle neu aufgenommen Messungen mit den Withings Consumer-Geräten, dem von Ihnen erstellten Patienten zugeordnet. Wenn Sie die Ihnen zugeordneten Messergebnisse einsehen wollen, stehen Ihnen in der Hauptnavigation folgende Möglichkeiten zur Verfügung:

Mein Konto: Hier können Sie die von Ihnen angegebenen Benutzer-Informationen einsehen.

Database: Hierbei handelt es sich um eine Datenbank mit allen Messungen, die Ihnen zugeordnet werden.

Ergebnisse: Hier werden alle Messergebnisse, sortiert nach Messtyp grafisch dargestellt.

Filter Optionen: Hier können Sie die Suche nach den Messwerten eingrenzen und so beispielsweise den Mittelwert alle Temperaturmessungen anzeigen lassen.

Abb. III: Startseite der Webapplikation „GetFHIRData“

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Benutzer Informationen:

Vorname: Marius

Nachname: Müller

Geschlecht: männlich

Geburtsdatum: 1998-09-30

Adresse: Hohenzollernstraße 14, 54290 Trier, Germany

Telefon Nr.: +49651509876

Abb. IV: Seite „Mein Konto“ der Webapplikation „GetFHIRData“

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Observations:

ID	Observation ID	Patient	Effective Date	Category	Category-Code	Observation Type	Type-Code	Result	Unit
1447	2001455	Patient/1981552	2020-09-04 00:22:54	Vital Signs	vital-signs	Body temperature	8310-5	36.9189987	°C
1448	2001454	Patient/1981552	2020-09-04 00:23:01	Vital Signs	vital-signs	Body temperature	8310-5	36.9309997	°C
1449	2001453	Patient/1981552	2021-01-23 13:26:46	Vital Signs	vital-signs	Body temperature	8310-5	35.8219985	°C
1450	2001452	Patient/1981552	2021-01-23 13:26:53	Vital Signs	vital-signs	Body temperature	8310-5	35.6780014	°C
1442	2001460	Patient/1981552	2021-01-23 13:27:38	Vital Signs	vital-signs	Body temperature	8310-5	35.7579994	°C
1443	2001459	Patient/1981552	2021-01-23 13:27:47	Vital Signs	vital-signs	Body temperature	8310-5	36.1329994	°C
1444	2001458	Patient/1981552	2021-01-23 13:28:03	Vital Signs	vital-signs	Body temperature	8310-5	35.9449996	°C
1445	2001457	Patient/1981552	2021-01-23 13:28:08	Vital Signs	vital-signs	Body temperature	8310-5	36.0540008	°C
1464	2001438	Patient/1981552	2021-01-23 13:28:08	Vital Signs	vital-signs	Body temperature	8310-5	36.054	°C
1446	2001456	Patient/1981552	2021-01-25 16:04:35	Vital Signs	vital-signs	Body temperature	8310-5	36.6240005	°C
1463	2001439	Patient/1981552	2021-01-25 16:04:35	Vital Signs	vital-signs	Body temperature	8310-5	36.624	°C
1435	2001467	Patient/1981552	2021-01-25 18:13:40	Vital Signs	vital-signs	Body temperature	8310-5	36.4090003	°C
1462	2001440	Patient/1981552	2021-01-25 18:13:40	Vital Signs	vital-signs	Body temperature	8310-5	36.409	°C
1477	2001425	Patient/1981552	2021-02-07 14:25:55	Vital Signs	vital-signs	Body mass index	39156-5	23.374725341797	kg/m2
1478	2001424	Patient/1981552	2021-02-07 14:25:55	Vital Signs	vital-signs	Body weight	29463-7	80	kilogram
1475	2001427	Patient/1981552	2021-02-07 14:27:42	Vital Signs	vital-signs	Heart rate	8867-4	59	beats/minute
1476	2001426	Patient/1981552	2021-02-07 14:27:42	Vital Signs	vital-signs	Blood pressure systolic & diastolic	85354-9	137/71	mmHg
1473	2001429	Patient/1981552	2021-02-09 14:01:47	Vital Signs	vital-signs	Heart rate	8867-4	62	beats/minute
1474	2001428	Patient/1981552	2021-02-09 14:01:47	Vital Signs	vital-signs	Blood pressure systolic & diastolic	85354-9	138/83	mmHg
1471	2001431	Patient/1981552	2021-02-09 14:15:45	Vital Signs	vital-signs	Heart rate	8867-4	61	beats/minute

Abb. V: Datenbank der Webapplikation „GetFHIRData“

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Temperatur Messwerte

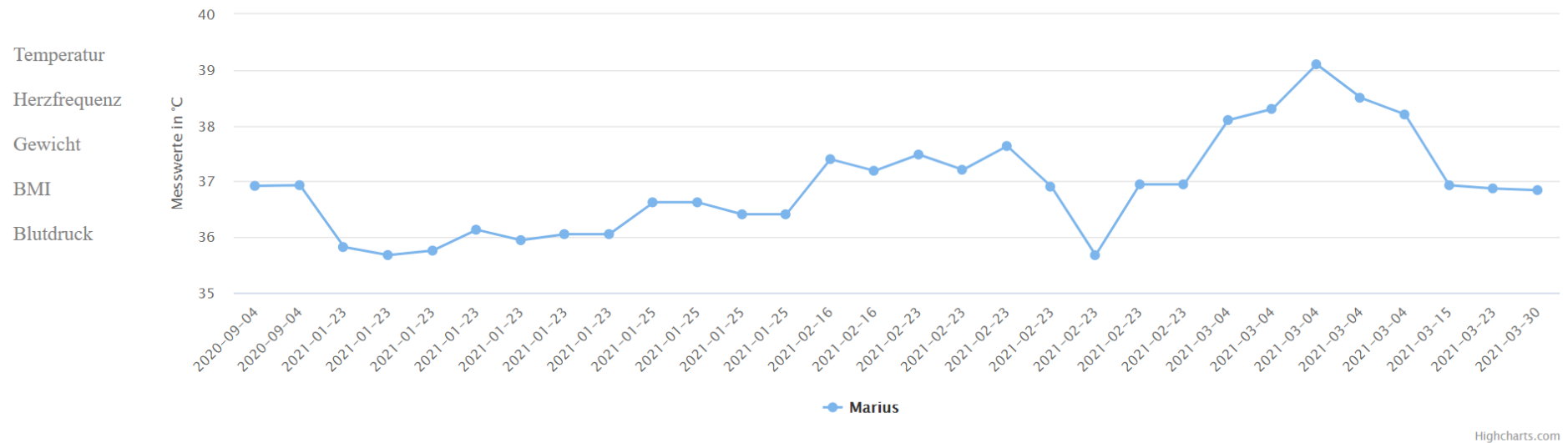


Abb. VI: Darstellung der Temperatur-Messwerte in der Web-Applikation „GetFHIRData“
Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Herzfrequenz Messwerte

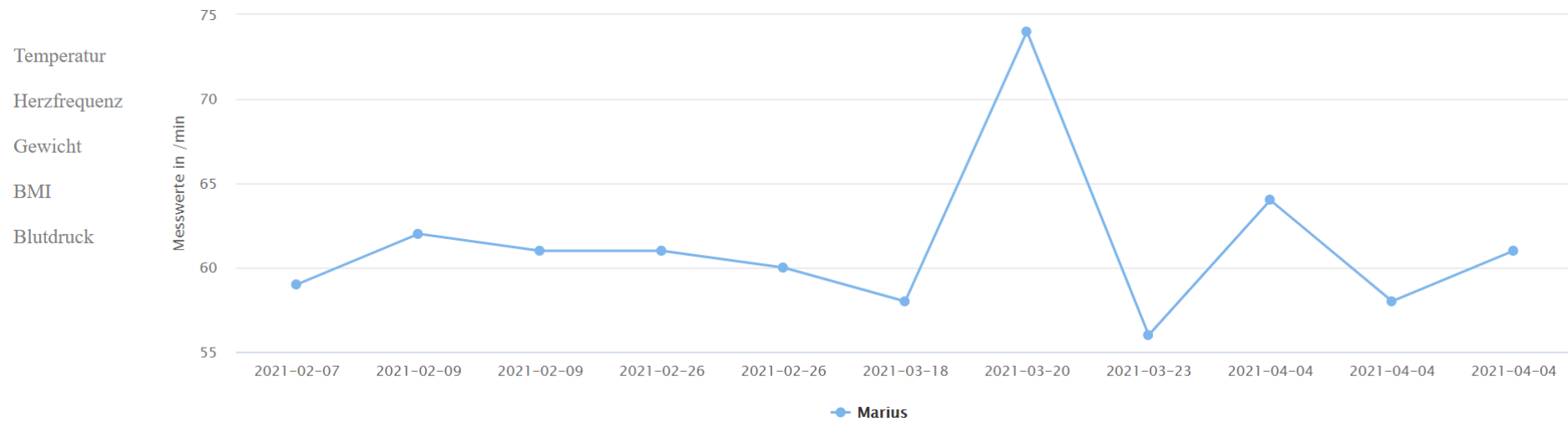
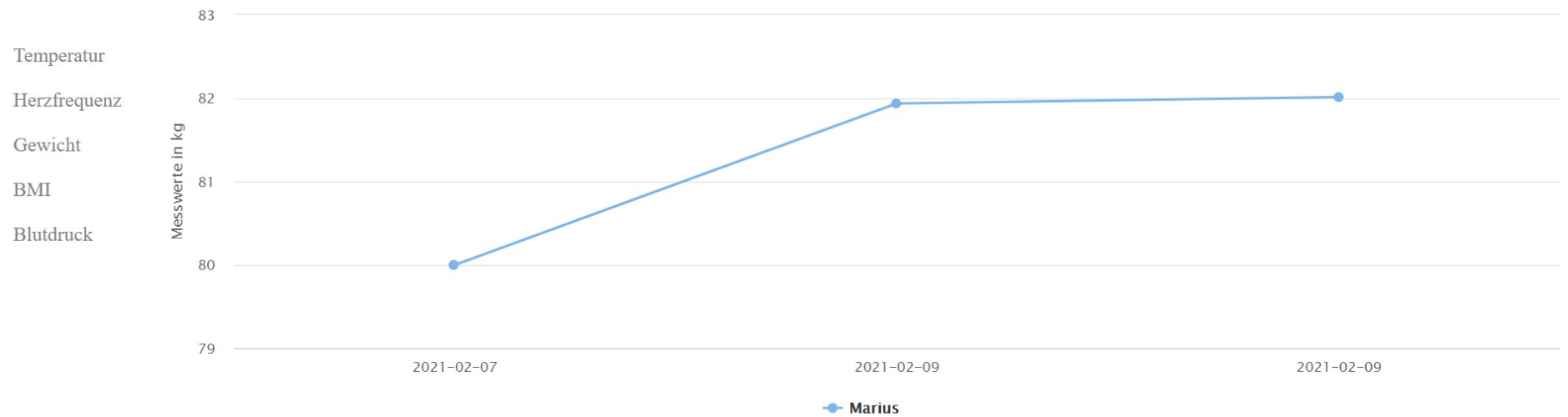


Abb. VII: Darstellung der Herzfrequenz-Messwerte in der Web-Applikation „GetFHIRData“

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Körpergewicht Messwerte



Highcharts.com

Abb. VIII: Darstellung der Körpergewicht-Messwerte in der Web-Applikation „GetFHIRData“

Quelle: Eigene Aufnahme

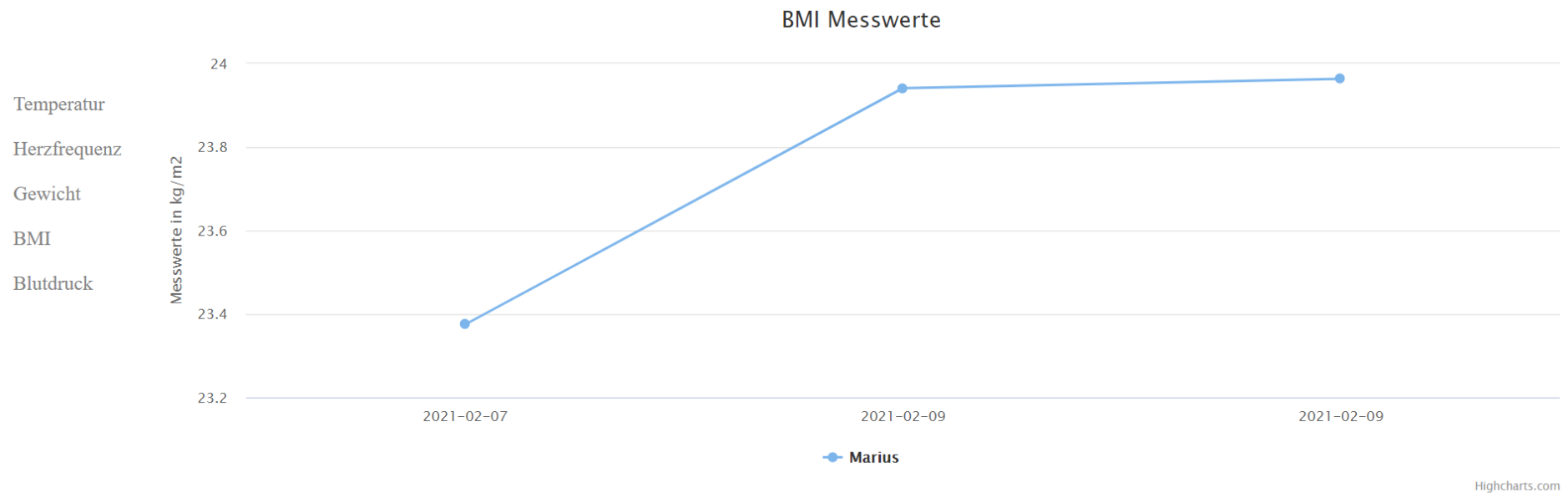
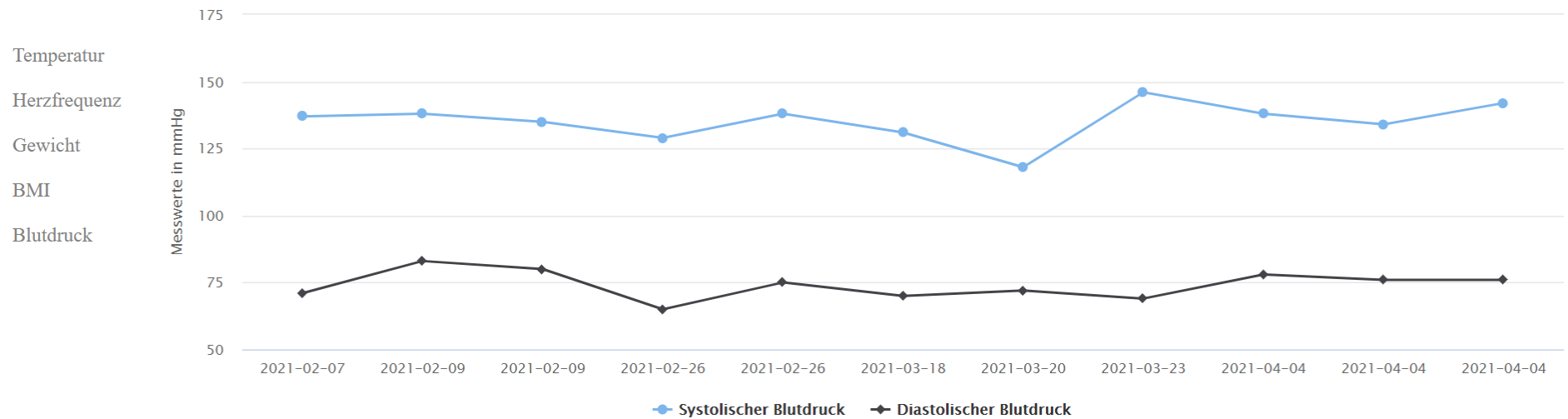
[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Abb. IX: Darstellung der BMI-esswerte in der Web-Applikation „GetFHIRData“
Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Blutdruck Messwerte



Highcharts.com

Abb. X: Darstellung der Blutdruck-Messwerte in der Web-Applikation „GetFHIRData“

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Filter Optionen für FHIR-Ressourcen

☒ Alle Werte

☐ Mittelwert

☐ Höchster Wert

☐ Niedrigster Wert

Temperatur

größer als

38

search

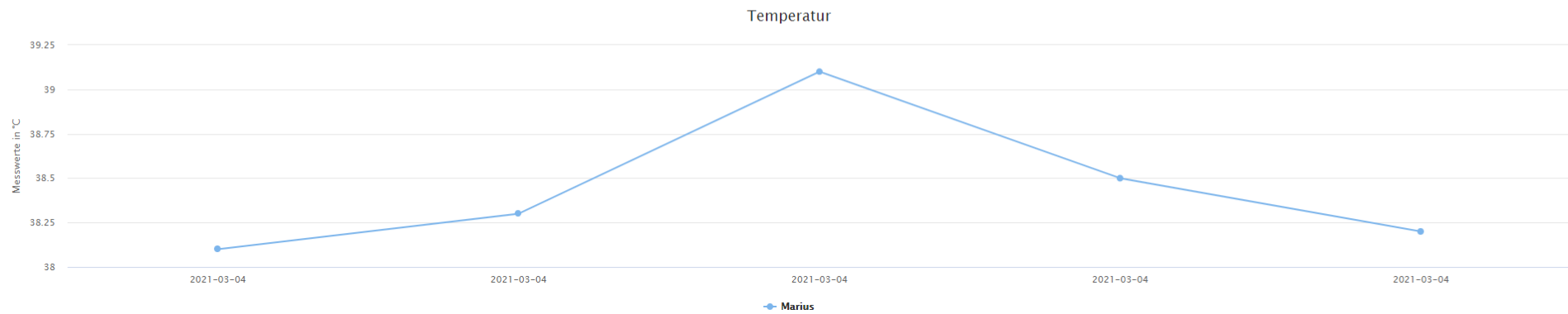


Abb. XI: Filter Optionen für FHIR-Ressourcen am Beispiel der Temperatur-Messwerte (größer als 38)

Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Filter Optionen für FHIR-Ressourcen

☐ Alle Werte

☒ Mittelwert

☐ Höchster Wert

☐ Niedrigster Wert

Herzfrequenz

-- Operation --

-- Wert --

search

Herzfrequenz: Der Mittelwert beträgt 61.27/min

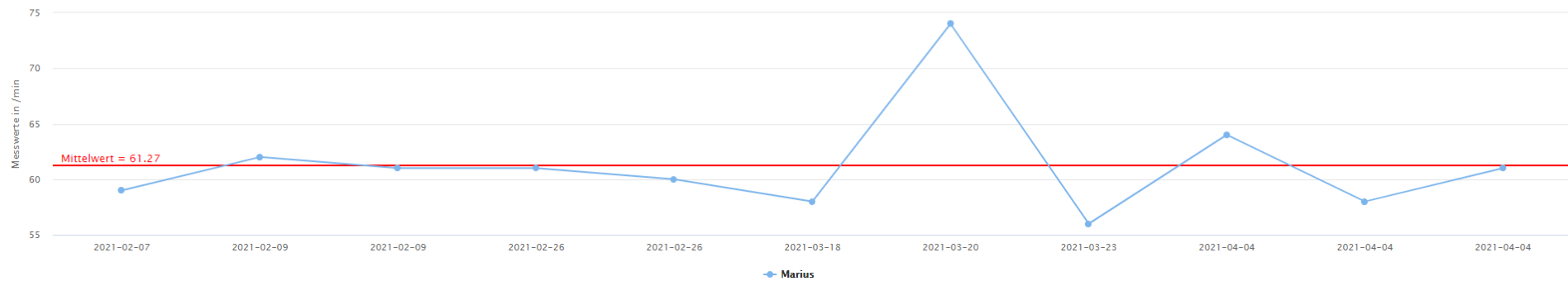


Abb. XII: Beispielhafte Mittelwertberechnung der Herzfrequenz-Messwerte mit Hilfe der Filter Optionen
Quelle: Eigene Aufnahme

[Home](#)[Mein Konto](#)[Database](#)[Ergebnisse](#)[Filter Optionen](#)[Abmelden](#)

Filter Optionen für FHIR-Ressourcen

- ☐ Alle Werte
- ☐ Mittelwert
- ☒ Höchster Wert
- ☐ Niedrigster Wert

Herzfrequenz



-- Operation --



-- Wert --

search

Herzfrequenz: Der maximale Wert beträgt 74/min

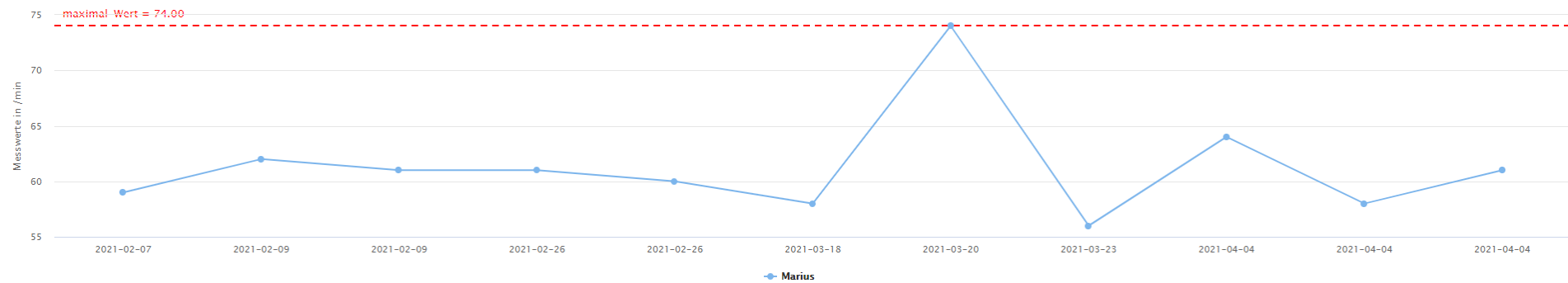


Abb. XIII: Beispielhafte Berechnung des Maximalwertes der Herzfrequenz-Messwerte mit Hilfe der Filter-Optionen
Quelle: Eigene Aufnahme

Filter Optionen für FHIR-Ressourcen

☒ Alle Werte

-- Messtyp --

-- Operation --

-- Wert --

search

☐ Mittelwert

☐ Höchster Wert

☐ Niedrigster Wert

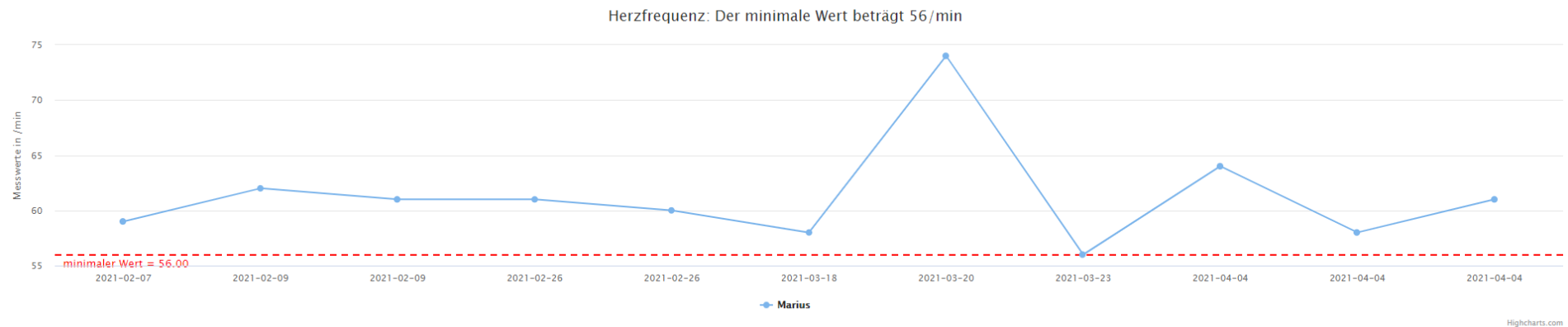


Abb. XIV: Beispielhafte Berechnung des Minimalwertes der Herzfrequenz-Messwerte mit Hilfe der Filter-Optionen
Quelle: Eigene Aufnahme

Anhang II: Organisatorisches

Pflichtenheft und Projektplan

Arbeitstitel

Folgender Arbeitstitel wird vorgeschlagen:

FHIR in Digitalen Gesundheitsanwendungen

Automatisierte Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Testserver

Pflichtenheft

Durch den Fortschritt der Digitalisierung im Gesundheitswesen, geht der Trend von offline zu online, von Desktop zu Tablet und von Server zu Cloud. Somit steigt auch die Nachfrage danach, medizinische Informationen einrichtungs- und sektorübergreifend kommunizieren zu können. *FHIR* (Fast Healthcare Interoperability Resources) ist ein von HL7 erarbeiteter Standard, der den Datenaustausch zwischen Softwaresystemen im Gesundheitssektor, sowie mobile und cloud-basierte Anwendungen unterstützt. Im Rahmen meiner Bachelorarbeit werde ich aufzeigen, wie mit *FHIR* versucht wird, die Schnittstelle zwischen verschiedenen Einrichtungen auszufüllen und Patienten ortsunabhängig mit ihren Daten zu verbinden. Dabei wird dargestellt, welche Vorteile *FHIR* gegenüber bereits etablierten Standards (wie HL7 V2, CDA, etc.) bietet, und wie *FHIR* durch die Verwendung von Technologien, die aus Applikationen wie Google, Twitter und Facebook bekannt sind, eine einfache Handhabung für Entwickler garantiert.

Der Einsatz von *FHIR* wird anhand von *Medizinischen Informationsobjekten (MIOs)* der Kassenärztlichen Bundesvereinigung (KBV) erläutert. Dabei wird besonders auf die Frage eingegangen, warum *FHIR* für die Umsetzung der *MIOs* genutzt wird: Die Kommunikation mittels *FHIR* setzt sich immer wieder aus denselben Bausteinen zusammen, unabhängig davon, ob diese als einzelne Ressource, als komplexes Dokument oder Nachricht übermittelt werden. Dieser Vorteil der Wiederverwendbarkeit (und auch der Anpassungsfähigkeit durch Profilierung einzelner Ressourcen) wurde von der KBV genutzt, um Basis-Profile zu erstellen, die in sehr vielen *MIOs* wiederverwendet werden und auf die Bedürfnisse der KBV angepasst wurden - gleiche Information soll auch an gleicher Stelle wiederzufinden sein. Dieser Komplex wird genauer beschreiben, um im Anschluss einen Ausblick auf die Zukunft zu geben und in den praktischen Teil der Arbeit überzuleiten.

Ziel der Arbeit ist es, die prototypische Erfassung von Vitaldaten (wie Blutdruck, Körpergewicht, Körpertemperatur) mit Consumer-Geräten umzusetzen. Diese Daten sollen auf einem *FHIR*-Server abgelegt werden. Somit wird ein Bezug zu einer fiktiven Digitalen Gesundheitsanwendung (DiGA) hergestellt, die eine ähnliche Funktionalität wie die zu den verwendeten Geräten gehörenden Apps anbietet, die die erfassten Daten jedoch nicht in einer proprietären Cloud, sondern auf einem frei gewählten *FHIR*-Server ablegt.

Die Grundlage des praktischen Teils ist eine Abschlussarbeit einer Kommilitonin, in der mit drei medizinischen Messgeräten (Blutdruckmessgerät, Schläfenthermometer, Waage) des Herstellers Withings und dessen Android-App, Vitaldaten und Körpergewicht erfasst wurden. Diese Daten wurden aus der App abgegriffen, in die Hausautomation OpenHab2 übertragen und dort mit Grafana individuell visualisiert. Die Hausautomation lief dabei auf einem Raspberry-Pi, sodass eine Extraktion der Daten erarbeitet werden musste. Diese Arbeit soll als Vorlage genutzt werden, um den Einstieg zu erleichtern.

Im Rahmen meiner Arbeit wird der vorgegebene Aufbau nachgestellt, jedoch wird anstelle der Hausautomation, der Kommunikationsserver MIRTH auf dem Raspberry Pi installiert. Die Daten der Withings-App werden vom Android-Betriebssystem des verwendeten Smartphones per Chronjob auf den Raspberry Pi gezogen und dort mit Hilfe des MIRTH in das FHIR-JSON Format (HL7 ORU) gebracht. Anschließend werden die Daten vom MIRTH auf einem FHIR-Testserver abgelegt. Dort soll ein fiktiver Patient angelegt werden, dem die Vitaldaten im passenden Format übergeben werden. Zusätzlich soll eine Funktionalität eingebaut sein, die die abgelegten Daten abgreift, aufbereitet und grafisch darstellt.

Meilensteine & Projektplan

Meilensteine		
Nr.	Bezeichnung	Datum
1	Anforderungssammlung und Problembeschreibung	24.01.2021
2	Anmelden der Abschlussarbeit	25.01.2021
3	Recherche beendet	15.02.2021
4	Abschluss der Vorbereitung auf praktischen Teil	08.03.2021
5	Fertiger Systementwurf	29.03.2021
6	Fertiger Entwurf für Visualisierung der Daten	05.04.2021
7	Funktionierendes Gesamtsystem	19.04.2021
8	Erfassung der ersten Patientendaten	26.04.2021
9	Abgabe der Abschlussarbeit	25.05.2021
10	Präsentation der Methoden und Ergebnisse	N.N.

Arbeitstitel:	
Start des Projekts:	13.01.2021
Aktueller Planungshorizont bis zum:	31.05.2021

#	Kategorie	Aktivität	MS
I.	Projektvorbereitung	Anforderungssammlung und Problembeschreibung - Erste Recherche - Festlegen des allgemeinen Rahmens der Arbeit - Erstellen der Pflichtenhefts - Erstellen des Projektplans Absegnen aller Anforderungen durch Betreuer Anmelden der Abschlussarbeit	 1 1 1 1 2 2
II.	Recherche	Lesen & verstehen der Abschlussarbeit die als Grundlage dient Recherchearbeit über: - HL7 FHIR - MIOs der KBV - Kommunikationsserver MIRTH - Umgang mit Consumer-Geräten - Umgang mit Raspberry-Pi Stichwortartiger Entwurf des theoretischen Teils der Arbeit	 3 3 3 3 3 3 3
III.	Vorbereitung auf praktischen Teil	Einrichten der Komponenten - Installation des MIRTH auf Raspberry Pi - Einrichten des Admin-Launchers - Rooten des Smartphones Erlernen des Umgangs mit verwendeten Komponenten	 4 4 4 4 4

		Erlernen des Umgangs mit Consumer-Geräten	4
		Erster Systementwurf	5
IV.	Entwicklung	Erstellen eines Patienten auf FHIR-Testserver	5
		Übermittlung der erfassten Daten von Smartphone auf Raspberry-Pi	5
		Einrichten des Kommunikationsservers MIRTH	5
		Ablage der Daten auf FHIR-Testserver	5
		Abgreifen der abgelegten Daten	6
		Visualisierung der abgegriffenen Daten	6
V.	Testphase	Test der Funktionalitäten	7
		- Fehlerbehebung	7
		- Überarbeitung	7
		- Optimierung	7
		Erstellen eines Patienten auf FHIR-Testserver	8
		Erfassen von Vitaldaten mit Consumer-Geräten	8
		Dokumentation der Ergebnisse	9
VI.	Dokumentation	Stichwortartiger Entwurf des praktischen Teils der Abschlussarbeit	9
		Fertigstellung des theoretischen Teils	9
		Fertigstellung des praktischen Teils	9
		Erstellen der abgabefertigen Abschlussarbeit	9
VII.	Präsentation	Zusammenfassen aller Methoden & Ergebnisse für Präsentation	10
		Erstellen der finalen Präsentation	10

Fortschrittsberichte

Fortschrittsbericht vom 12.02.2021

Nachdem die ersten Recherchearbeiten zu FHIR und dem Einsatz von FHIR im Zusammenhang mit Medizinischen Informationsobjekten abgeschlossen waren, wurde die ausgeliehene Hardware ausgiebig getestet (Nokia Körperwaage, Blutdruckmessgerät, Schläfenthermometer, Raspberry PI 3B). Zuerst wurde Raspbian und anschließend der MIRTH Connect Server auf dem Raspberry PI installiert. Über die IP-Adresse des Raspberry PIs und dem entsprechenden Port, konnte der Server über den Admin-Launcher auf dem PC gestartet werden. Außerdem wurde die Withings Health Mate App auf dem hier verwendeten Smartphone installiert und die Geräte mit dieser konfiguriert.

Anschließend wurde die Bachelorarbeit von Frau Feldmann (*“Assistenzsystem zur Vitalparametererfassung - Anleitung für Zuhause und Konzept für die Kinder-Uni“*) ,die als Grundlage für die Arbeit gilt, durchgearbeitet und versucht diese grundlegend nachzustellen. Dazu sollte im ersten Schritt ein Smartphone, bzw. Tablett gerootet werden. Da es sich bei dem gewählten Smartphone jedoch um ein Gerät der Marke Huawei handelte und Huawei seit 2018 keine Unlock-Codes mehr herausgibt, stellte sich das Rooten als schwierig heraus. Daher wurde ein alternatives Gerät von Samsung verwendet, was sich ohne Probleme rooten ließ. Um einen ersten Überblick zu erhalten, wurde sich dazu entschieden, den Raspberry Pi erst einmal beiseitezulegen und die Funktionalitäten auf dem PC zu testen, da hier eine einfachere Handhabung möglich ist. Im Laufe der nächsten Wochen wird sich intensiver mit dem Raspberry PI auseinandergesetzt.

Nachdem das Smartphone gerootet war, wurde eine Android Debug Bridge (ADB) installiert, jedoch konnte die App „adb Insecure“ nicht fehlerfrei verwendet werden, da ein Starten der App das USB-Debugging unterbrach. Somit wurde der Zugriff auf die gesuchte „Withings-WiScale“ Datei weiterhin vom Smartphone untersagt. Daher wurde ein Root Explorer installiert, mit dem die Datei in ein nicht-geschütztes Verzeichnis kopiert wurde und so von der ADB abgefragt werden konnte. Dies stellt jedoch keine endgültige Lösung dar, weshalb sich im Laufe der Arbeit mit diesem Problem auseinandergesetzt werden muss.

Nachdem die gesuchte “Withings-WiScale“-Datei zur Verfügung stand, wurde ein Java Programm geschrieben, dass die Daten verarbeitet (da mir Java als Programmiersprache noch nicht bekannt war, musste ich diese erst einmal erlernen; komme mittlerweile jedoch gut zurecht). Auf Grundlage des Programms von Frau Feldmann werden die Daten eingelesen und in Form einer XML-Datei ausgegeben. Diese XML-Datei wird vom MIRTH Connect Server abgefragt und ins Format HL7 ORU gebracht. Dazu wurde ein Template einer HL7 ORU Message erstellt und in MIRTH eingefügt.

Im nächsten Schritt wird versucht die Struktur der HL7 ORU Message ins FHIR Format zu übertragen. Ist dies gelungen, wird ein Patient auf dem FHIR Testserver <https://hapi.fhir.org/> angelegt, dem die Messergebnisse im FHIR Format übergeben werden.

Fortschrittsbericht vom 16.03.2021

Aus den im ersten Fortschrittsbericht beschriebenen grundlegenden Überlegungen, wurde in den letzten Wochen ein erster Prototyp. Die XML-Dateien, die die von den Withings-Geräten aufgenommenen Gesundheitsdaten enthalten, werden nun vom MIRTH Connect ins FHIRFormat gebracht und auf dem FHIR-Testserver <https://hapi.fhir.org> abgelegt. Dazu wurden passende Templates der FHIR-Ressourcen im JSON-Format erstellt und in den MIRTH eingefügt. Außerdem wurde der Raspberry Pi in das Projekt integriert, sodass die Software, die die Daten der Withings-Geräte ausliest und der MIRTH Connect Server nun auf dem Raspberry Pi laufen.

Ziel der Arbeit ist es, die Vitaldaten automatisiert zu erfassen und abzulegen. Daher wurde ein Cronjob auf dem Raspberry Pi implementiert, der die gesuchte „Withings-WiScale“ Datei auf dem Smartphone alle 30 Sekunden über die Android Debug Bridge (ADB) abfragt und anschließend die Software ausführt, die die Daten ins XML-Format bringt und im passenden Verzeichnis ablegt. Dort werden die Daten automatisiert vom MIRTH Connect erfasst, erkannt, mit Hilfe der Templates in FHIR-JSON-Format gebracht und anschließend auf dem FHIRTestserver abgelegt. Im Cronjob werden vor der eigentlichen Abfrage die Rechte der WithingsDatei geändert, sodass sie auch ohne Root-Rechte problemlos von der ADB abgefragt werden kann. Auf diese Weise wurde das Problem mit der App „abdb Insecure“ umgangen, da sie nun nicht mehr von Nöten ist.

Nachdem der erste Prototyp stand, wurde mit der Erstellung der eigentlichen Abschlussarbeit begonnen. Der theoretische Grundlagenteil der Arbeit ist nun fertiggestellt. Im nächsten Schritt wird das Konzept des Projekts beschrieben und die verwendete Software und Geräte werden vorgestellt. Danach soll das Projekt komplett fertiggestellt und unterdessen dokumentiert werden. Dazu werden die bisherigen Arbeitsabläufe überarbeitet und eine Möglichkeit zum Retrieval und zur Visualisierung der abgelegten FHIR-Daten implementiert.

Fortschrittsbericht vom 20.04.2021

Seit dem letzten Fortschrittsbericht wurden alle Teile des Projekts zu einem lauffähigen System zusammengefügt. Zur Visualisierung der Daten wurde eine Web-Applikation entwickelt. In der folgende Funktionalitäten implementiert sind:

Beim Start der Web-Applikation wird man zu der Login-Seite weitergeleitet. Hier hat man die Möglichkeit einen Account zu erstellen und sich dann mit der angegebenen E-Mail + Passwort anzumelden. Mit der Erstellung des Accounts wird automatisch eine FHIR-Ressource des Typs „Patient“ auf dem FHIR-Testserver erzeugt und die Patienten-ID wird mit den Account-Informationen zusammen in einer MySQL-Datenbank abgelegt. Loggt sich ein Benutzer ein, wird eine Session gestartet und alle Informationen des Benutzers werden in einer Session-Variable gespeichert.

Wird nun eine Messung mit den Withings-Geräten durchgeführt, wird das Messergebnis (wie in den vorherigen Fortschrittsberichten beschrieben) von der proprietären Withings-Darstellung ins XML-Format gebracht. Bevor die Daten nun auf den FHIR-Server geladen werden, wird überprüft, welcher Benutzer gerade in der Web-Applikation eingeloggt ist. Dazu habe ich ein kleine „API“ implementiert, die die Benutzer-Informationen und die Patienten-ID des momentan eingeloggten Benutzers zurückgibt. Diese Patienten-ID von vom Mirth verwendet, sodass die Messergebnisse, dem richtigen Patienten (gerade eingeloggter User) zugeordnet werden.

In der Web-Applikation sind einige Funktionalitäten zur Visualisierung der Messergebnisse enthalten: Die Patienten-Daten können angezeigt werden, die Messergebnisse werden als Graphen dargestellt und es gibt die Möglichkeit nach speziellen Messergebnissen zu suchen. Beispielsweise könnte nach allen Temperatur-Messwerten über 36°C gesucht werden.

Außerdem wurde die Dokumentation der Abschlussarbeit fertiggestellt. Im nächsten Schritt wird ein Fazit verfasst und anschließend Formatierungsarbeiten vorgenommen. Zusätzlich zur Dokumentation der praktischen Arbeit wird ein Kurzfilm erstellt, in dem der prototypische Ablauf einer Messung inklusive der technischen Aspekte des Abgreifens der Daten, deren Konvertierung und Ablage auf dem FHIR-Testserver, sowie das Retrieval der Daten dargestellt wird.

Besprechungsprotokolle:

Besprechung vom 21.01.2021

Die erste Besprechung zur Abschlussarbeit fand am 21.01.2021 online über BigBlueButton statt. In der Besprechung wurde ein grober Rahmen für die Abschlussarbeit geschaffen und es wurde geklärt, welches Konzept umgesetzt werden soll.

Dabei wurde insbesondere auf die 2019 von Frau Feldmann verfasste Abschlussarbeit der Hochschule Trier mit dem Titel „Assistenzsystem zur Vitalparametererfassung“ eingegangen, die als Grundlage für diese Arbeit dienen soll, um so einen einfacheren Einstieg in die Thematik zu ermöglichen.

Es wurde festgelegt, dass Vitalparameter mit Consumer-Geräten der Firma Withings aufgenommen und auf einen Raspberry Pi geladen werden sollen. Dort sollen die Messergebnisse (ähnlich wie in der Arbeit von Frau Feldmann) verarbeitet werden. Danach sollen die Daten in FHIR-Ressourcen konvertiert und auf einen FHIR-Testserver geladen werden. Wie die Daten vom Server abgegriffen werden sollen, sei freigestellt. Der Umfang der Arbeit soll 60-100 Seiten betragen.

Die benötigte Hardware kann bei Herrn Prof. Dr. Benzschawel ausgeliehen werden.

Besprechung vom 02.03.2021

Die zweite Besprechung zur Abschlussarbeit fand am 21.01.2021 online über BigBlueButton statt. Zu Beginn wurde der bisher erreichte Fortschritt beschrieben und inhaltliche Fragen wurden gestellt.

Anschließend wurde über die Gliederung der Arbeit gesprochen. Von Herrn Prof. Benzschawel wurde angemerkt, dass ein Kapitel „Related Work“ eingefügt werden soll, in dem der Zweck der Arbeit beschrieben werden soll. Außerdem soll der internationale Standard „LOINC“, der für die Kodierung der Messwerte verwendet wird, beschrieben werden.

Zudem wurde geklärt, wie das „Endprodukt“ aussehen soll. Es wurde sich darauf geeinigt, eine Web-Applikation zu erstellen, da hierfür eine simple Entwicklungsumgebung verwendet werden kann. Herr Prof. Benzschawel wies darauf hin, dass viele Screenshots der Web-Applikation und der anderen Komponenten in die Arbeit integriert werden sollen, da die einzelnen Teile so in einen Kontext gebracht werden können.

Zum Schluss wurde geklärt, ob und in welcher Form Quellcode in die Arbeit integriert werden kann.

Besprechung vom 28.03.2021

Die zweite Besprechung zur Abschlussarbeit fand am 21.01.2021 online über BigBlueButton statt. In dieser Besprechung wurden die letzten bestehenden Fragen zum Aufbau und zur Abgabe der Abschlussarbeit geklärt.

Zu Beginn wurde der Aufbau der Arbeit mit allen Unterkapiteln besprochen und Anpassungen vorgenommen. Es wurde geklärt in welcher Form die Arbeit und der Quellcode abgegeben wurden. Es wurde sich darauf geeinigt, die Arbeit im PDF-Format und einen Abgabeordner mit allen Quellcode-Dateien in Seafile hochzuladen. Es ist darauf zu achten, dass der Code ausreichend dokumentiert ist und dass die Ordnerstruktur nachvollziehbar ist.

Außerdem wurde geklärt, welche Formatvorlage für die Formatierung der Arbeit verwendet werden soll und wie die Quellen angegeben werden sollen. Dazu stellte Herr Prof. Benzschawel einige nützliche Links zur Verfügung.

Zum Schluss wurde geklärt welche Elemente im Abschluss-Video enthalten sein sollen und wie das abschließende Kolloquium ablaufen soll.

Schriftverkehr mit HL7-Deutschland

Re: HL7 Kontaktformular – Neue Nachricht von Marius Müller



Von Alexander Zautke <alexander@fire.ly>
An muelmari@hochschule-trier.de <muelmari@hochschule-trier.de>
Kopie Heckmann, Simone <sh@gefyr.de>
Datum 2021-04-08 12:36

Hallo Marius,

die Verwendung des HL7 FHIR Logos ist vollkommen unproblematisch. Die Vorgaben die eingehalten werden müssen, sind hier aufgeführt: <https://www.hl7.org/fhir/license.html#2.20.2.1>. Die dort verlinkten Logos können verwendet werden, solange die entsprechenden Trademark-Hinweise aufgeführt werden.

Viel Erfolg bei deiner Arbeit!

Viele Grüße
Alexander Zautke
(Stellvertretender Leiter Technisches Komitee FHIR HL7 Deutschland)

Von: HL7 Deutschland <muelmari@hochschule-trier.de>
Antworten an: <muelmari@hochschule-trier.de>
Datum: Mittwoch, 7. April 2021 um 12:46
An: <info@hl7.de>
Betreff: HL7 Kontaktformular – Neue Nachricht von Marius Müller

Nachricht über das HL7-Kontaktformular:

Firma:

Vorname:

Marius

Nachname:

Müller

Betreff:

Allgemein

Nachricht:

Guten Tag, mein Name ist Marius Müller, ich studiere Medizintechnik an der Hochschule Trier und verfasse momentan meine Bachelorarbeit zu dem Thema: "FHIR in Digitalen Gesundheitsanwendungen - Automatisierte Erfassung, Ablage und Retrieval von Vitaldaten auf einem FHIR-Testserver". Wie der Name verrät geht es darum, Vitalparameter mit Consumer-Geräten aufzunehmen und diese mit Hilfe eines Kommunikationsservers (Mirth-Connect) automatisiert ins FHIR/JSON-Format zu konvertieren und auf dem FHIR-Testserver (<http://hapi.fhir.org>) abzulegen. Um die Messergebnisse zu visualisieren, habe ich eine Web-Applikation (die lokal gehostet wird und somit nicht über eine URL erreichbar ist) entwickelt, die die Vitalparameter aufbereitet und für den Benutzer grafisch darstellt. Da sich in der Bachelorarbeit alles um das Thema FHIR dreht, würde ich gerne im Header der Web-Applikation das FHIR-Logo platzieren. Auch wenn die Web-Applikation nicht online erreichbar sein wird und diese lediglich von den bewertenden Professoren eingesehen werden kann, wollte ich Sie trotzdem um Erlaubnis bitten. Mit freundlichen Grüßen Marius Müller

Datenschutzerklärung:

Ja, akzeptiert

Erklärung des Kandidaten

- ☒ Die Arbeit habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen- und Hilfsmittel verwendet.

Datum

Unterschrift der Kandidatin / des Kandidaten