

**Fachhochschule Aachen
Campus Jülich**

Fachbereich 9
Medizintechnik und Technomathematik

Analyse und Konzeption einer mobilen Applikation für die graphische Darstellung von Informationen eines medizinischen Falles

Seminararbeit
im Studiengang Angewandte Mathematik und Informatik

von

Abdullah Enes Kilic
Matr.-Nr.: 3276242

15. Dezember 2022

1. Prüfer: Prof. Dr. Philipp Rohde
2. Prüfer: Gökhan Karakas



Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

Analyse und Konzeption einer mobilen Applikation für die graphische Darstellung von Informationen

eines medizinischen Falles

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Name: Abdullah Enes Kilic

Aachen, den 15.12.2022

The image shows a handwritten signature in black ink. The signature consists of a large, stylized capital letter 'E' followed by a period and the name 'Kilic' in a cursive script.

Unterschrift der Studentin / des Studenten

Kurzfassung

Krankenhäuser führen für Patienten eine Patienten- und eine Fallakte. Diese beinhalten Informationen über den jeweiligen Patienten und medizinischen Fall u.a. Diagnosen, Erkrankungen oder die Aufenthalts- bzw. Behandlungsdauer. In diesen werden der Allgemeinzustand eines Patienten und weitere Informationen erfasst. Der Allgemeinzustand wird in fünf Grade von null bis fünf eingeordnet. Darunter werden die körperlichen, seelischen und geistigen Verfassungen eines Patienten gemessen und bewertet. Es ist für Ärzte anspruchsvoll, diese Informationen aus dem Gedächtnis oder den Akten abrufen zu können. Dies gilt insbesondere, wenn viele und wechselnde Patienten zu versorgen sind. Deshalb bietet sich eine mobile EDV-Lösung zur Unterstützung der Ärzte an.

In dieser Arbeit wird dargelegt, wie man die Patienten- und Falldaten erfassen kann und diese für den behandelnden Arzt in einer App graphisch visualisiert darstellen kann. Hierzu wird die Art der Visualisierung diskutiert. Es werden Programmiersprachen vorgestellt und analysiert, da die App auf verschiedenen Plattformen laufen muss. Hier werden die Vorteile sowie Nachteile einzelner Programmiersprachen zum Entwickeln der mobilen Applikation im Hinblick auf die Funktionen, als auch der Bereitstellung im App Store, Play Store oder beiden, bewertet.

Diese Arbeit legt die Grundlagen für die Entwicklung einer mobilen Applikation und der graphischen Visualisierung der Werte von Patienten für die behandelnden Ärzte.

Inhalt

1	Einleitung	1
1.1	Einleitung	1
1.2	Anforderung und Ziel der Arbeit.....	1
2	Grundlagen	2
2.1	Organisatorische Grundlagen	2
2.1.1	Patientenakte	2
2.1.2	Fall.....	2
2.1.3	Fallakte	3
2.1.4	Stationäre, Ganztägig Ambulante und Ambulante Behandlung.....	3
2.2	Medizinische Grundlagen.....	4
2.2.1	Allgemeinzustand	4
2.2.2	Pflegeanamnese	4
2.2.3	Ernährungszustand	5
2.2.4	Entwicklungszustand.....	5
2.3	EDV-Grundlagen.....	6
2.3.1	Java Virtual Machine (JVM)	6
2.3.2	Objektorientierte und Plattformübergreifende Programmiersprache	6
2.3.3	Funktionale Programmierung	6
2.3.4	Widget	6
2.3.5	SchemaId	6
2.4	Visualisierung der Informationen der Daten	6
3	Anforderungen an die App	10
3.1	Anforderung	10
3.2	Benötigte Werte und deren Gruppierung.....	10
3.2.1	Werte	10
3.2.2	Gruppierung.....	11
3.3	Aufbau der App.....	13
3.4	User Interface	13
4	Auswahl der Programmiersprache	16
4.1	Programmiersprachen unter Android	16
4.1.1	Java.....	16
4.1.2	Kotlin.....	17
4.2	Programmiersprachen unter IOS: Swift	17
4.3	Crossplattform: Flutter	18

4.4	Fazit.....	18
5	Konzeption der App mit Flutter	20
5.1	Austausch von Daten.....	20
5.2	Verbindung zwischen der App und dem Server.....	20
5.3	Serverseitig benötigte Funktionen.....	21
5.4	Anmelden	21
5.5	Ausgabe der Patientenliste	22
5.6	Visualisierung der Daten.....	23
6	Zusammenfassung und Ausblick.....	25
7	Literaturverzeichnis.....	26

Abbildungsverzeichnis

Abbildung 1	Beispiel für die Darstellung der Allgemeinen Informationen eines Patienten	7
Abbildung 2	Beispiel Liniendiagramm für den Allgemeinzustand.....	7
Abbildung 3	Beispiel Liniendiagramm des Ernährungszustands.....	8
Abbildung 4	Beispiel für den Entwicklungszustand eines Kindes.....	9
Abbildung 5	Dokument für den Allgemeinzustand.....	11
Abbildung 6	Dokument für den Ernährungszustand.....	11
Abbildung 7	Dokument für den Entwicklungszustand.....	12
Abbildung 8	Use Case Diagramm.....	13
Abbildung 9	Anmeldeseite.....	14
Abbildung 10	Startseite.....	14
Abbildung 11	Ausgabeseite.....	15
Abbildung 12	Java.....	16
Abbildung 13	Kotlin.....	17
Abbildung 14	Swift	17
Abbildung 15	Flutter	18
Abbildung 16	JSON: Anfrage an Server.....	20
Abbildung 17	JSON: Antwort von Server.....	20
Abbildung 18	Verbindung zwischen der mobilen Applikation und dem Server.....	21
Abbildung 19	Beispiel http-Request in Flutter.....	21
Abbildung 20	Aktivitätsdiagramm für die Anmeldung auf Seiten des Servers	22
Abbildung 21	Aktivitätsdiagramm für das Lesen der Patientenliste auf Seiten des Servers.....	22
Abbildung 22	Aktivitätsdiagramm für das Lesen der Patientendaten auf Seiten des Servers.....	23
Abbildung 23	min. Klassendiagramm.....	24

1 Einleitung

1.1 Einleitung

Bei einem Krankenhausbesuch, sei es wegen einer Erkältung, aufgerissenen Wunden oder Knochenbrüchen etc. werden für Patienten sowohl eine Patienten- als auch eine Fallakte, wo Informationen abgelegt und gespeichert werden, erstellt. Diese Informationen werden jahrelang gespeichert, sodass Ärzte in den Krankenhäusern die Informationen im Bedarfsfall abrufen können. Die gespeicherten Daten können hilfreich und sogar ausschlaggebend bei der Bewertung aktueller Krankheitssymptome sein.

Das Studium der Akten kann ermüden. Die Anzahl der Patienten mit unterschiedlichen Symptomen erschwert die Übersicht und führt zu einer Gefahr die Daten oder Informationen der Patienten zu verwechseln. Mit einer graphischen Darstellung dieser Informationen sind die Informationen schnell zu erfassen und einfach zu verarbeiten.

Auch können ursächliche Schemata schnell erkannt werden und Behandlungen früher und effizienter eingeleitet werden. Dies erhöht die Effizienz der Behandlung.

Wenn sich die Krankheit schematisch wiederholt, so kann das Schema als Ursache festgestellt werden. Wenn eine Krankheit saisonal im Winter auftaucht, so liegt die Jahreszeit als ursächlich nahe.

1.2 Anforderung und Ziel der Arbeit

Diese Arbeit legt die Grundlagen, die Informationen und den Zustand des Patienten graphisch dar. Die mobile Applikation wird analysiert und konzipiert, um mittels guter theoretischer Grundlagen deren Entwicklung zu vereinfachen und diese möglichst effizient programmieren zu können.

Eine mobile Applikation mit einer visuellen Darstellung bietet sich hierfür an. Sie wertet anhand der Informationen der Patienten- und Fallakten, die über den Server von den Krankenhausdatenbanken gezogen werden, den Allgemeinzustand und weitere Informationen aus und stellt diese graphisch dar. Die Graphik soll eine einfache und logische Übersicht der Daten gewährleisten.

2 Grundlagen

Vor der Analyse und die Konzeption der mobilen Applikation sind grundlegende Begriffe der Medizin und der Informatik zu klären.

2.1 Organisatorische Grundlagen

2.1.1 Patientenakte

Eine Patientenakte ist vom Arzt für jeden Patienten zu führen. Diese hat alle wesentlichen medizinischen Daten aus Untersuchungen, Diagnosen, Therapien und deren Ergebnisse zu dokumentieren.

Enthaltene Informationen:

- Anamnese: Die Anamnese enthält bekannte Beschwerden, psychische Verfassungen, soziale Belastungen und Krankheitsfälle in der Familie des Patienten.
- Diagnosen: Unter Diagnosen werden festgehaltene Krankheiten oder Allergien dokumentiert.
- Hier werden sowohl Aufzeichnungen über Umstände, als auch über den Verlauf von Behandlungen dokumentiert.
- Es werden auch Untersuchung und deren Befunde (= Ergebnisse) festgehalten. Beispiele hierfür sind Bluttest oder EKGs (= Herzuntersuchungen)
- Therapien und ihre Wirkungen: Darunter werden sowohl verordnete Behandlungen und Arzneimittel, als auch ihre Wirkung, Folgen und mögliche Nebenwirkungen aufgelistet.
- Eingriffe und ihre Auswirkungen: Ein Beispiel sind Operationsberichte.
- Alle vom Patienten erfolgten Einwilligungen und Aufklärungen, wie zum Beispiel das Einsehen der Akten durch weitere Ärzte.
- Arztbriefe: Diese Briefe sind Mitteilungen von oder an andere Mediziner¹.

Die Patientenakte ist entsprechend § 630f BGB nach Abschluss der Behandlung für mindestens zehn Jahre aufzubewahren. Der Patient hat ein Einsichtsrecht.²

2.1.2 Fall

Ein medizinischer Fall hat immer eine eindeutige Fallnummer und enthält die Fallakte. Des Weiteren werden Patientendaten, Aufnahme- und Entlassungsdaten, Ärzte etc. dokumentiert. Je nach Behandlung, sei es eine stationäre, ganztägig ambulante oder ambulante Behandlung, wird ein Fall angelegt.

¹ [Ges20]

² [Bür22]

2.1.3 Fallakte

Um den Informationsfluss zwischen Ärzten zu gewährleisten, gibt es Fallakten, die die Informationen im Kontext einer Behandlung sammeln.

Da mehrere Personen auf die Fallakte zugreifen und dort Daten einstellen oder entnehmen können, gibt es klare Bedingungen für die Daten und die Einwilligung des Patienten ist erforderlich.³

Bevor der Patient im Krankenhaus eingewiesen wird, wird in der Regel eine Voruntersuchung des Patienten vom niedergelassenen Arzt durchgeführt. Sobald der Patient im Krankenhaus eintrifft, wird eine Fallakte erstellt, in die die Dokumente des einweisenden Arztes übernommen werden. So kann der aufzunehmende Arzt anschließend im Krankenhausinformationssystem, kurz KIS, sehen, dass bereits Informationen in der Fallakte vorliegen und kann mit diesen weiterarbeiten. Bei stationären Behandlungen werden kontinuierlich Dokumente, wie OP-Berichte, Röntgenbilder, Befunde, Entlassbriefe⁴ oder das Ergebnis einer Tumorkonferenz⁵ etc., in die Fallakte eingestellt, wenn diese für die Weiterbehandlung von Bedeutung sind. Wenn die Erkrankung zur Folge hat, dass ein häufiger Wechsel zwischen ambulanten und stationären Behandlungen stattfindet, wird über die Fallakte eine aktuelle Datengrundlage sichergestellt.⁶

2.1.4 Stationäre, Ganztägig Ambulante und Ambulante Behandlung

Stationäre Behandlung: Wenn der Patient zeitlich ununterbrochen (mindestens einen Tag und eine Nacht) im Krankenhaus verbringt, dann ist dies eine stationäre Behandlung. Hier findet eine physische und organisatorische Eingliederung des Patienten in das spezifische Versorgungssystem des Krankenhauses statt.

Ein Patient ist dann noch stationär eingestuft, wenn dieser eigenmächtig das Krankenhaus, während er als stationär eingestuft ist, verlässt.

Ganztägig Ambulante (= Teilstationäre) Behandlung: Hier findet wie in der stationären Behandlung eine physische und organisatorische Eingliederung des Patienten in das spezifische Versorgungssystem des Krankenhauses statt, jedoch beschränkt sich die Unterbringung des Patienten in Tages- und Nachtkliniken. Somit ist die teilstationäre Behandlung keine ununterbrochene Anwesenheit.

Ambulante Behandlung: Eine ambulante Behandlung kommt der teilstationären Behandlung sehr nahe. Hier liegt eine ambulante Behandlung vor, wenn der Patient weder die Nacht vor, noch die Nacht nach dem Eingriff im Krankenhaus verbringt.⁷

³ [Neu10]

⁴ [Dmrz]

⁵ [Neu10]

⁶ [Neu10]

⁷ [Rei18]

2.2 Medizinische Grundlagen

2.2.1 Allgemeinzustand

Der Allgemeinzustand, kurz AZ, beschreibt die Allgemeine körperliche (physische), geistige (psychische) und seelische (emotionale) Verfassung eines Patienten und die Auswirkung auf seinen Alltag.⁸ Dazu gehört die Konstitution, Körperhygiene, Mobilität, Aktivität, Fähigkeit zur Selbstversorgung, Kommunikation und kognitive Fähigkeiten.⁹

Hier findet meistens schon eine Erfassung des Allgemeinzustandes im Rahmen des Aufnahmegesprächs statt und gilt als ärztliche Untersuchung als auch Pflegeanamnese. Dabei ist das Ergebnis im klinischen Alltag nicht nach objektiv messbarem Parameter gebunden, sondern es ist hier eine mehr oder weniger subjektive Einschätzung des Untersuchers, die auf Erfahrungswerten beruht. Man unterscheidet zwischen gutem Allgemeinzustand, leicht reduziertem Allgemeinzustand, reduziertem Allgemeinzustand und stark reduziertem Allgemeinzustand.¹⁰

Jedoch gibt die WHO (Weltgesundheitsorganisation) hier auch einen standardisierten Index zur Beurteilung des Allgemeinzustandes eines Patienten an. Diese werden in fünf verschiedenen Stadien beginnend mit der Null unterschieden.

- Grad 0: Der Patient zeigt volle Aktivität und ist zu einem normalen Leben und Arbeiten befähigt
- Grad 1: Der Patient zeigt eine eingeschränkte Aktivität, leichte Arbeit ist allerdings noch möglich
- Grad 2: Der Patient kann sich noch selbst versorgen, aber ist nicht arbeitsfähig. Allerdings ist er nicht bettlägerig und weniger als 50% der Tageszeit werden als Ruhezeit benötigt.
- Grad 3: Die Selbstversorgung des Patienten ist stark eingeschränkt und ist auf Pflege und Hilfe angewiesen. Allerdings benötigt der Patient mehr als 50% der Tageszeit als Ruhezeit.
- Grad 4: Der Patient ist ständig bettlägerig und pflegebedürftig.¹¹

In der mobilen Applikation wird zum Messen des Grades und der graphischen Darstellung der Informationen zu einem Fall eines Patienten der standardisierte Index der WHO verwendet.

2.2.2 Pflegeanamnese

Die Pflegeanamnese ist eine Sammlung an Daten über den Patienten und seinen Hintergrund, die bei der Analyse seines Gesundheitszustandes verwendet werden kann. Sie bildet somit die

⁸ [Wik21]

⁹ [MNSF]

¹⁰ [Wik21]

¹¹ [MNSF]

Grundlage des Pflegeprozesses und der Pflegeplanung, um eine optimale Pflegepraxis zu ermöglichen.¹²

2.2.3 Ernährungszustand

Der Ernährungszustand (kurz EZ genannt) ist ein durch Beobachtung und Messung erhobener Teilaspekt des klinischen Gesamtzustandes eines Patienten, der wichtige Rückschlüsse auf die Stoffwechselsituation hergibt.¹³

Zu der Bewertung werden folgende Parameter betrachtet:

Körperlicher Aspekt (Konstitution), Körpergewicht, Körpergröße, Body-Mass-Index (BMI), Hautfaltendicke, Kreatinin-Größen-Index (KGI) und Bioelektrische Impedanz

Hier gibt es fünf verschiedene Formen des EZ.

- Kachexie (pathologischer Gewichtsverlust)
- Reduzierter Ernährungszustand
- Guter Ernährungszustand
- Übergewicht
- Adipositas (übermäßige Vermehrung des Fettgewebes)¹⁴

„Als Faustregel gilt: Verliert der Patient ungewollt über zehn Prozent seines Gewichts in den letzten sechs Monaten oder über fünf Prozent im letzten Monat, deutet dies auf eine Mangelernährung hin. Geachtet werden sollte auch auf Alarmsignale eines beginnenden Nährstoffmangels wie Appetitlosigkeit, einseitige Ernährung, das Auslassen und Ablehnen von Mahlzeiten, Schluckstörungen, körperliche Schwäche, Hautveränderungen, Teilnahmslosigkeit und Depressionen.“¹⁵

2.2.4 Entwicklungszustand

Dieser Zustand wird nur bei Kindern gemessen. Hier wird die Kognition (das Hören, Sehen, Spüren und Denken), die Sprachentwicklung, die Feinmotorik (z. B.: das Umgehen mit Stiften und Scheren oder das Greifen und Setzen bestimmter kleiner Gegenstände), die Grobmotorik (z. B.: das Laufen, Klettern, Fahrradfahren, Weit- und Hochsprung), die Soziale Entwicklung und Selbständigkeit, die Emotionale- und Ich-Entwicklung und das Lern- und Spielverhalten beurteilt.¹⁶

¹² [Wik22]

¹³ [Ant14]

¹⁴ [Ant14]

¹⁵ [Gra08]

¹⁶ [StiNe]

2.3 EDV-Grundlagen

2.3.1 Java Virtual Machine (JVM)

Die JVM ist Teil der Laufzeitumgebung von Java und dient zur Ausführung des Java-Bytecodes. Jedes gestartete Java-Programm wird in seiner eigenen virtuellen Maschine ausgeführt. Dadurch, dass die JVM nicht zum Betriebssystem gehört, fällt beim Absturz des Programms das Betriebssystem nicht aus.¹⁷

2.3.2 Objektorientierte und Plattformübergreifende Programmiersprache

Objektorientiert: Dieses Programmierparadigma konzentriert sich auf das Darstellen von Problemen unter Verwendung „realer“ Objekte, denen ein bestimmtes Verhalten zu eigen ist.¹⁸

Plattformübergreifend: Dies bedeutet, dass das Programm auf verschiedenen Plattformen (Webseiten, mobile Applikationen etc.) ausführbar und einsetzbar ist.

2.3.3 Funktionale Programmierung

Eine funktionale Programmierung ist ein Programmierparadigma, welches sich auf mathematische Grundlagen stützt.¹⁹

2.3.4 Widget

Widgets sind Elemente auf dem Bildschirm eines Smartphones, Tablets, etc., welche interaktiv genutzt werden können. Ein einfaches Beispiel für ein Widget ist eine Wetteranzeige.²⁰

2.3.5 SchemaId

In SAP ist eine SchemaId eine ID zu einem Dokument. Sie ist dazu da um Dokumente eindeutig zu referenzieren und im System zu suchen.

2.4 Visualisierung der Informationen der Daten

Im folgenden Abschnitt wird die Visualisierung der Informationen der Daten erläutert. Dabei wird darauf geachtet, welches Diagramm sich am besten für die darzustellende Informationen eignet. Des Weiteren wird geklärt, bei welchen Informationen sich eine graphische Darstellung anbietet.

Allgemeine Informationen: Die Allgemeinen Informationen können einfach aufgelistet werden, da sich hier keine anschauliche Diagrammdarstellung anbietet. Grund dafür ist, dass

¹⁷ [Wik+22]

¹⁸ [Kra18]

¹⁹ [Kra+18]

²⁰ [Ach22]

Namen, Alter, Adressen etc. von Patienten für eine graphische Darstellung nicht geeignet sind. Einfacher ist es, diese in einer Tabelle oder in einer einfachen Liste auszugeben.

Bsp.:

Name:	<input type="text" value="Sayajin"/>		
Vorname:	<input type="text" value="Son Goku"/>		
Zimmer-Nr.:	<input type="text" value="10Z101"/>		
Geb. Datum:	<input type="text" value="26.02.1986"/>	RR-Nr.:	<input type="text" value="101"/>
Beh. Arzt:	<input type="text" value="Meyer, Karolin"/>	Datum:	<input type="text" value="03.12.2022"/>
Therapeut:	<input type="text" value="Dr. Linsen"/>		

Abbildung 1 Beispiel für die Darstellung der Allgemeinen Informationen eines Patienten

Allgemeinzustand: Hier lohnt sich ein Liniendiagramm, wobei der Grad auf der Y-Achse und die Woche auf der X-Achse abgebildet werden.

Bsp.:

Ein Patient wird für 14 Wochen eingewiesen. Während dieser Zeit verschlechtert und verbessert sich sein Allgemeinzustand.

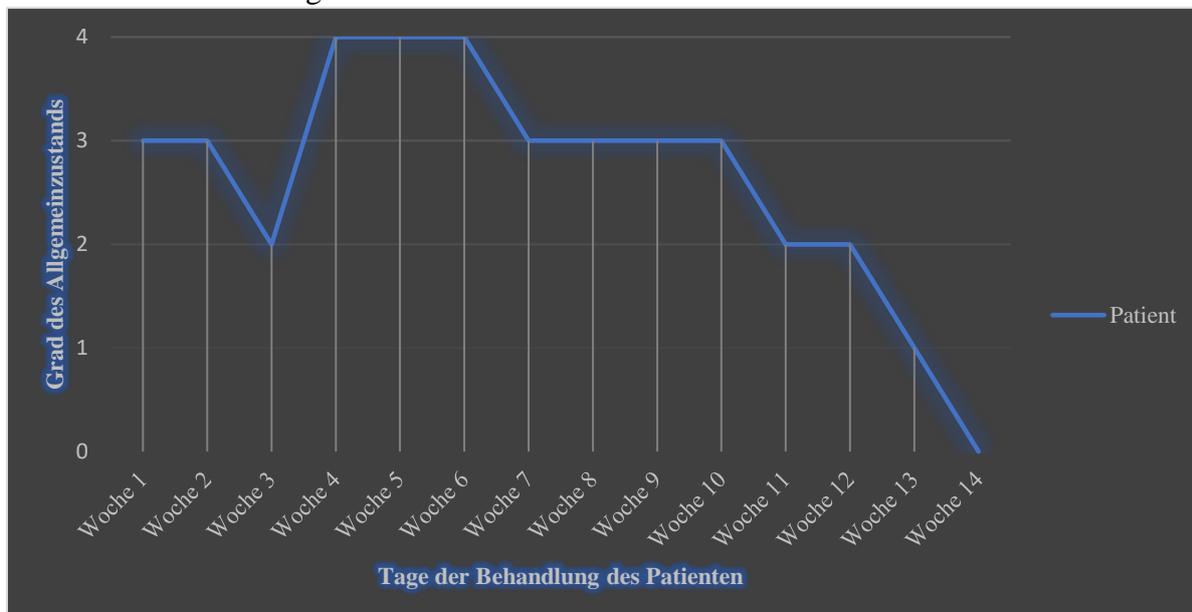


Abbildung 2 Beispiel Liniendiagramm für den Allgemeinzustand

Wie das Beispiel zeigt, macht ein Liniendiagramm deutlich, wie sich der Zustand des Patienten während seiner Behandlung verhält. Steigungen und Gefälle fallen sehr schnell ins Auge, wodurch die Behandlung dementsprechend angepasst werden kann. Zudem ist das Diagramm sehr trivial, wodurch das Verstehen des Diagramms nicht viel Zeit und Denken erfordert.

Ernährungszustand (bei Erwachsenen): Da hier die Werte, wie Körpergröße, Gewicht, BMI, Hautfaltendicke etc. sich nicht über wenige Tage ändern und die Werte sich sehr unterscheiden, ist ein einfaches Diagramm, welches diese Werte anzeigt sehr schwer darzustellen. Hier lohnt sich ein Liniendiagramm zu nutzen, welches nur das Ergebnis wöchentlich oder monatlich abbildet und die Bewertungsparameter unter dem Diagramm auflistet, sobald eine Woche oder ein Monat angeklickt wird. Das hat den Vorteil, dass der

Arzt beim Lesen des Diagramms nicht viele verschiedene Werte nachvollziehen muss, sondern sich nur die Daten ansieht, die er oder sie für Wichtig hält.

Dabei steht:

- -2 für „Kachexie“,
- -1 für „reduzierter Ernährungszustand“,
- 0 für „guter Ernährungszustand“,
- 1 für „Übergewicht“ und
- 2 für „Adipositas“.

Bsp.:

Ein Patient, welcher 12 Monate eingewiesen ist.

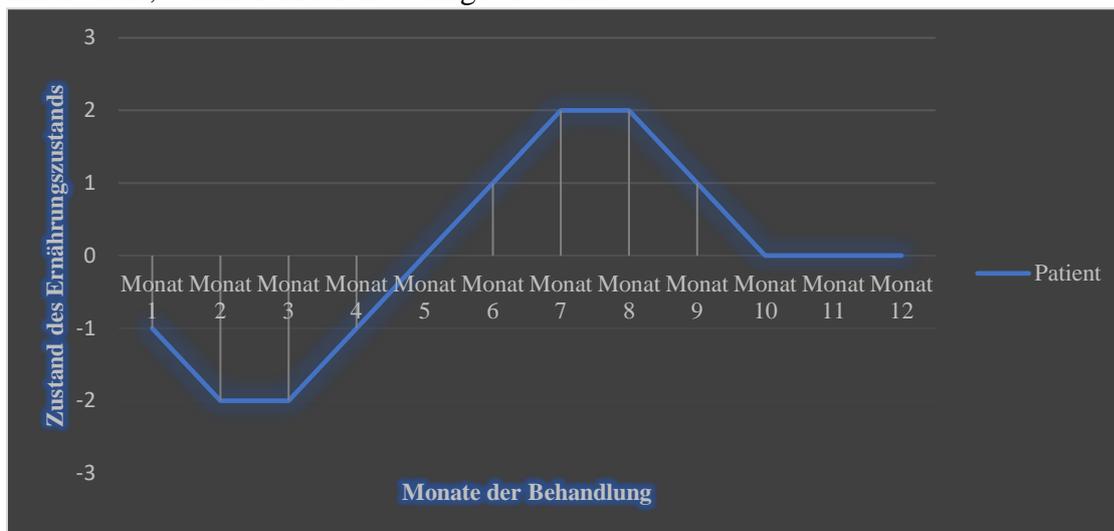


Abbildung 3 Beispiel Liniendiagramm des Ernährungszustands

Entwicklungszustand (bei Kindern): Für die Entwicklung eines Kindes gibt es vorgegebene Werte für das jeweilige Alter. Nachdem diese Werte ausgerechnet wurden, kann das Ergebnis in einem Säulendiagramm dargestellt werden, wobei die Y-Achse in Prozent den erreichten Wert und die X-Achse die Kategorie angibt. Diese Vorgehensweise hat gegenüber dem Liniendiagramms den Vorteil, dass der Fokus nur auf der aktuellen Entwicklung des Kindes liegt und nicht auf vergangene Entwicklungen. Hier ist es nämlich wichtig die aktuellen Schwächen auszubessern.

Bsp.:

Ein Kind hat bei Kognition 76%, Sprachentwicklung 92%, Feinmotorik 77%, Grobmotorik 63%, Soziale Entwicklung und Selbständigkeit 59%, Emotionale- und Ich-Entwicklung 33% und Lern- und Spielverhalten 55% erreicht.

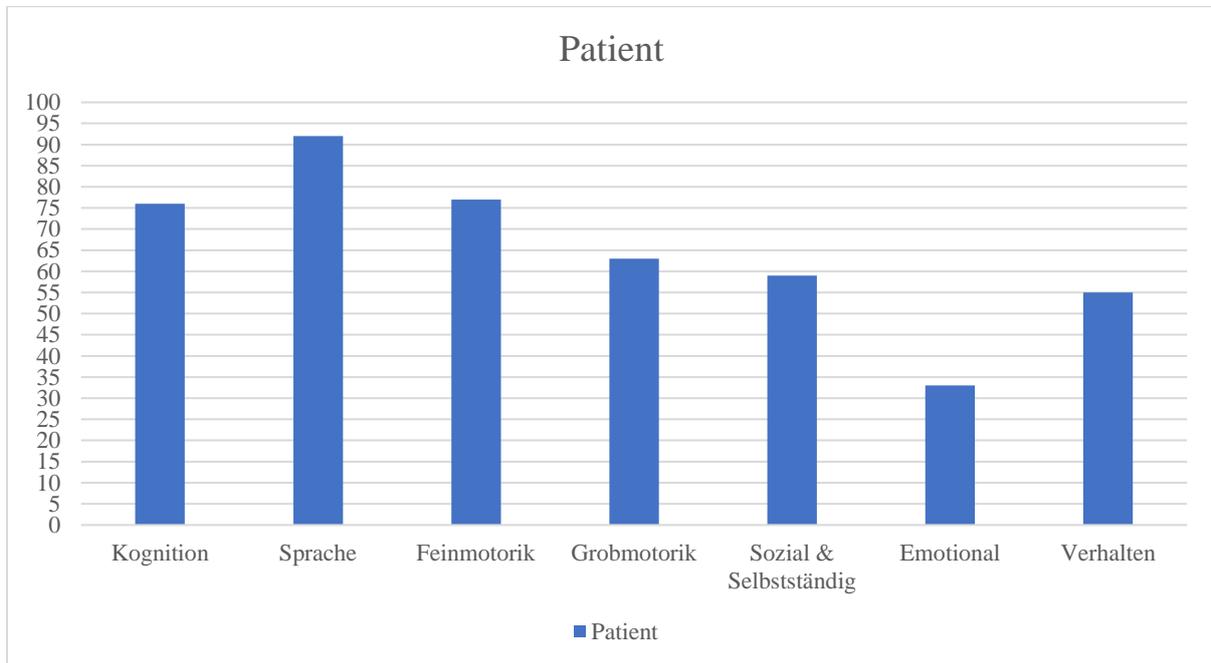


Abbildung 4 Beispiel für den Entwicklungszustand eines Kindes

3 Anforderungen an die App

3.1 Anforderung

In einem Gespräch mit meinem Ausbilder kam uns eine App in den Sinn, die das Leben der Ärzte einfacher machen könnte. Dabei entstand als erste Anforderung die graphische Darstellung. Da jeder Arzt am Tag mühevoll Texte mit vielen Fachbegriffen liest und diese später mit dem Patienten durchgehen muss, sollte diese mobile Applikation weniger Text mit gleichem Anteil an Informationen liefern. So fingen wir an die Anforderung weiter auszubauen.

Zunächst soll der Nutzer sich in der mobilen Applikation mit seinem Benutzernamen und Passwort anmelden können, da aus rechtlichen Gründen (Datenschutz) nur der Arzt die Daten seiner eigenen Patienten sehen darf. Sobald die Anmeldung erfolgreich war, sollen alle seine Patienten mit ihren Informationen aufgelistet werden.

Beim Anklicken eines Patienten sollen die Informationen des Patienten ausgegeben werden. Die auszugebenden Informationen sollen dabei der Allgemeinzustand und die Allgemeinen Informationen eines Patienten sein. Hinzu kommt, dass bei Erwachsenen der Ernährungszustand und bei Kindern der Entwicklungszustand beschrieben werden soll.

Um auch weiterhin mit möglichst wenig Text zu arbeiten, sollen Diagramme erstellt werden, die eine gute Übersicht über einen Zustand geben.

3.2 Benötigte Werte und deren Gruppierung

3.2.1 Werte

Um die benötigte visuelle Darstellung auszugeben, werden mehrere Werte benötigt. Diese werden im Folgenden kurz beschrieben.

Allgemeine Informationen: Hier sind einfache Daten, wie sie in jedem Krankenhaus zu finden sind, wichtig. Dazu gehören Patientenummer, Fallnummer, Name, Anschrift, Geburtsdatum, Alter, aktuelles Datum, belegtes Zimmer im Krankenhaus, behandelnder Arzt, Beruf/Tätigkeit und Telefonnummer.

Allgemeinzustand: Beim Allgemeinzustand werden die Werte vom Arzt abgeschätzt und in verschiedenen Dokumenten hinterlegt. Bei diesen Werten handelt es sich um die körperliche, geistige und seelische Verfassung eines Patienten und der Einschätzung des Grades, der vom Arzt dem Patienten zugeordnet wurde.

Ernährungszustand: Hier werden die Werte genau gemessen und errechnet. Die benötigten Werte sind die Konstitution, das Körpergewicht, die Körpergröße, der Body-Mass-Index (BMI), welches aus dem Körpergewicht und der Körpergröße berechnet wird, die Hautfaldendicke, der Kreatinin-Größen-Index (KGI) und die Bioelektrische Impedanz. Daraus wird dann der Zustand der Ernährung ermittelt.

Entwicklungszustand: Der Entwicklungszustand wird durch die Kognition, der Sprachentwicklung, der Feinmotorik, der Grobmotorik, der sozialen Entwicklung und

Selbständigkeit, der Emotionalen- und Ich-Entwicklung und das Lern- und Spielverhalten gemessen und in Prozent abgebildet, wie weit die Entwicklung vorangeschritten ist.

3.2.2 Gruppierung

Nun ist klar, welche Werte benötigt werden, jedoch fehlt die Information, wie diese gruppiert werden sollen.

Da zum einen diese Werte im SAP Standard nicht eingetragen werden und um nicht in vielen verschiedenen Dokumenten die benötigten Informationen suchen zu müssen, werden drei Dokumente erstellt, die zum entsprechenden Zeitpunkt aktualisiert werden müssen.

Das erste Dokument ist für die Auswertung des Allgemeinzustandes. Dieses hat den Namen „App für Seminararbeit – Allgemeinzustand“ und die SchemaId „APPALLG“.

App für Seminararbeit - Allgemeinzustand				
Verwaltungsdaten				
	Datum von	Datum bis	Grad	Grund
	05.12.2022	11.12.2022	1	test
	28.11.2022	04.12.2022	2	test

Eintrag hinzufügen Eintrag löschen

Abbildung 5 Dokument für den Allgemeinzustand

In dem Dokument ist eine Tabelle eingebaut, welche die Informationen enthält, die die App beim Ausführen anzeigen soll. Mit dieser kann später die Linie für das Liniendiagramm gezeichnet werden.

Im nächsten Dokument geht es um den Ernährungszustand des Patienten, welches den Namen „App für Seminararbeit – Ernährungszustand“ und die SchemaId „APPERNA“ besitzt.

App für Seminararbeit - Ernährungszustand									
Verwaltungsdaten									
Datum	Zustand	Konstitution	Körpergewicht	Körpergröße	BMI	Hautfalten dicke	KGI	Bioelektrische Impedanz	
01.01.2023	guter Ernährungszustand	Test	Test	Test	Test	Test	Test	Test	Test
01.12.2022	reduzierter Ernährungszustand	Test	Test	Test	Test	Test	Test	Test	Test
01.11.2022	Kachexie	Test	Test	Test	Test	Test	Test	Test	Test

Eintrag hinzufügen Eintrag löschen

Abbildung 6 Dokument für den Ernährungszustand

Um die graphische Darstellung leicht zu füllen hält dieses Dokument wie das erste Dokument die Daten ebenfalls in Tabellenform vor.

Das letzte Dokument, welches den Entwicklungszustand eines Kindes beinhaltet, wird mit Hilfe eines Säulendiagramms mit den aktuellsten Prozentzahlen der jeweiligen Kategorien dargestellt. Dieses hat den Namen „App für Seminararbeit – Entwicklungszustand“ und die SchemaId „APPENTW“.

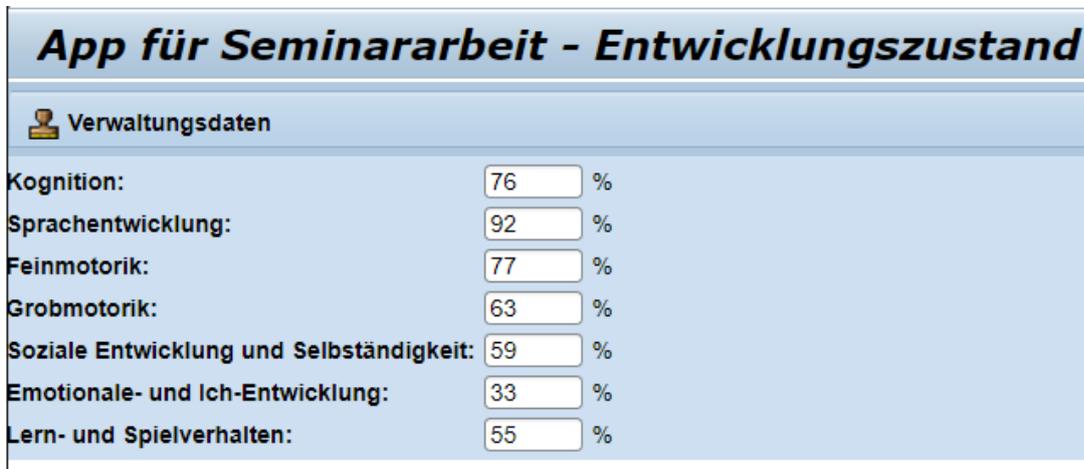


Abbildung 7 Dokument für den Entwicklungszustand

Falls diese drei Dokumente nicht angelegt sein sollten, werden entsprechende Meldungen in der App angezeigt. Des Weiteren darf nur ein Dokument des jeweiligen Dokuments existieren, da sonst Unstimmigkeiten zwischen den Dokumenten entstehen können. Dieses wird zuvor systemseitig sichergestellt.

Die benötigten Tabellen aus dem SAP-System sind USR02, NFPZ, NGPA, NFAL und NPAT. In diesen Tabellen werden die User-Informationen, die Informationen zum behandelnden Arzt und zum Patienten gespeichert.

Zu guter Letzt wird noch ein Web-Server benötigt, um der App nicht direkten Zugriff auf die Datenbank zu lassen. Dadurch wird die Kommunikation zwischen der App und dem KIS ermöglicht.

3.3 Aufbau der App

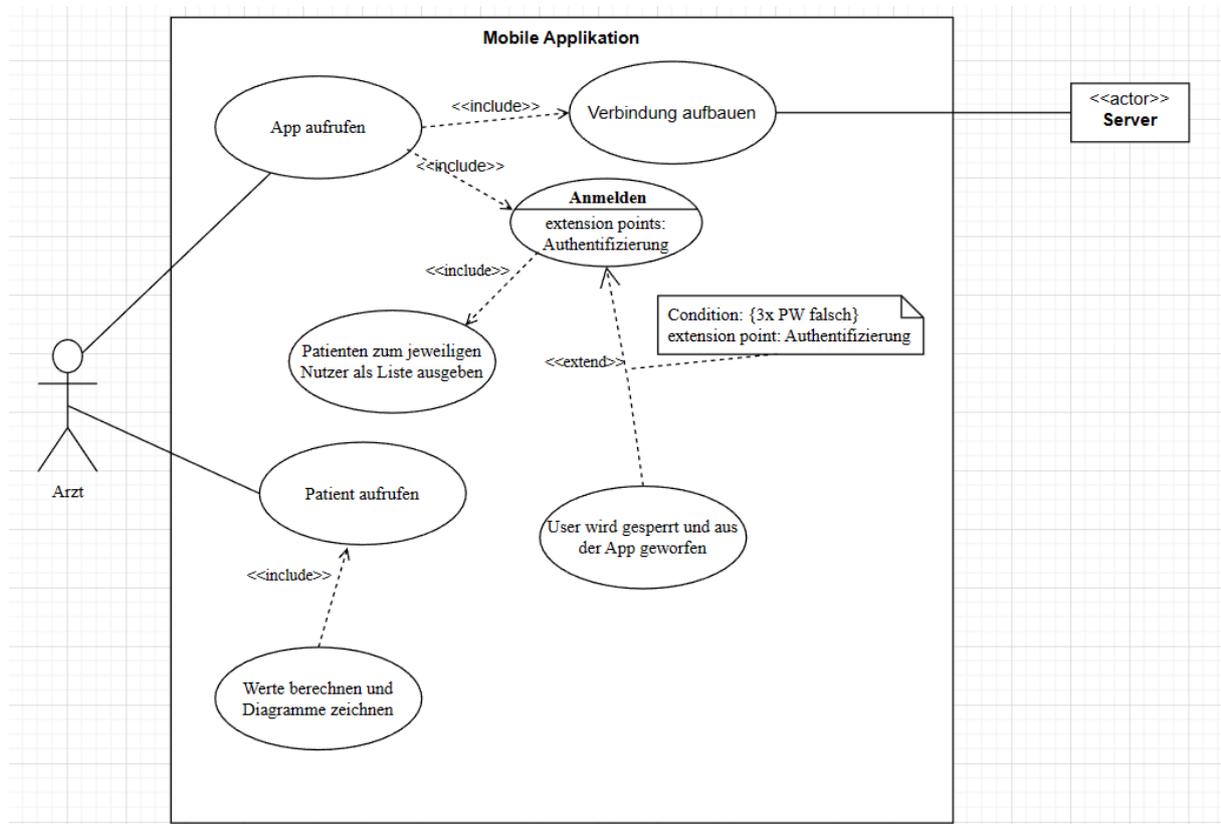


Abbildung 8 Use Case Diagramm

Sobald der Arzt die mobile Applikation startet, wird eine Verbindung zwischen der App und dem Server aufgebaut und der Arzt kommt auf die Anmeldeseite. Falls er sich richtig anmeldet, bekommt er eine Liste seiner Patienten zurück. Sollte er jedoch sein Passwort dreimal falsch eingeben, so wird er aus der App geworfen und sein User wird gesperrt. Mit der ausgegebenen Liste seiner Patienten, kann der Arzt einen Patienten auswählen und sich diesen anzeigen lassen. Sobald er einen Patienten auswählt, werden die Informationen geladen, die Werte berechnet, als ein Diagramm gezeichnet und für den Arzt ausgegeben.

3.4 User Interface

Um die App so gut wie möglich darzustellen wird im folgenden Text die einzelnen Seiten der mobilen Applikation dargestellt und beschrieben. Es folgen konzeptionelle Darstellungen des User Interface, weshalb die folgenden Seiten des User Interfaces noch angepasst werden können.

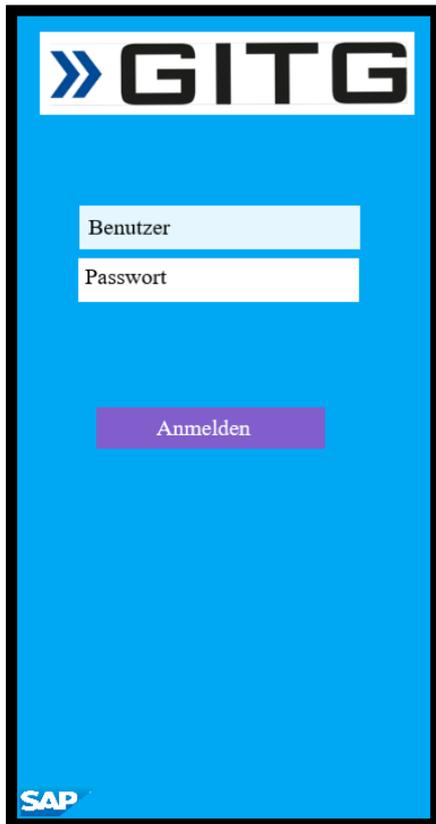


Abbildung 9 Anmeldeseite

Anmeldeseite:

Sobald die mobile Applikation gestartet wird, erscheint eine Anmeldeseite. Jeder Arzt muss sich hier anmelden, um diese mobile Applikation nutzen zu können.

Bei einer falschen Eingabe wird eine Error-Meldung angezeigt. Sollte das Anmelden jedoch beim dritten Versuch des Nutzers wieder scheitern, so wird der Nutzer aus der mobilen Applikation geworfen und sein User wird gesperrt.

Bei einer korrekten Eingabe kommt man über den Button „Anmelden“ auf die nächste Seite der mobilen Applikation.

Pat. Nr.	Name
2003001	Patient 1
2003002	Patient 2
2003003	Patient 3
2003004	Patient 4
2003005	Patient 5
2003006	Patient 6
2003007	Patient 7
2003008	Patient 8
2003009	Patient 9
2003010	Patient 10

Abbildung 10 Startseite

Startseite:

Sobald die Authentifizierung abgeschlossen ist, wird der Nutzer auf die Startseite weitergeleitet.

Hier werden alle Patienten, auf die der Nutzer der mobilen Applikation Zugriff hat, aufgelistet. Dabei wird zu jedem Patienten die Patientenummer, der Name des Patienten und weitere Werte, wie Fallnummer etc., Zeilenweise angegeben.

Des Weiteren kann jeder Nutzer sich hier aus der mobilen Applikation abmelden und diese verlassen.

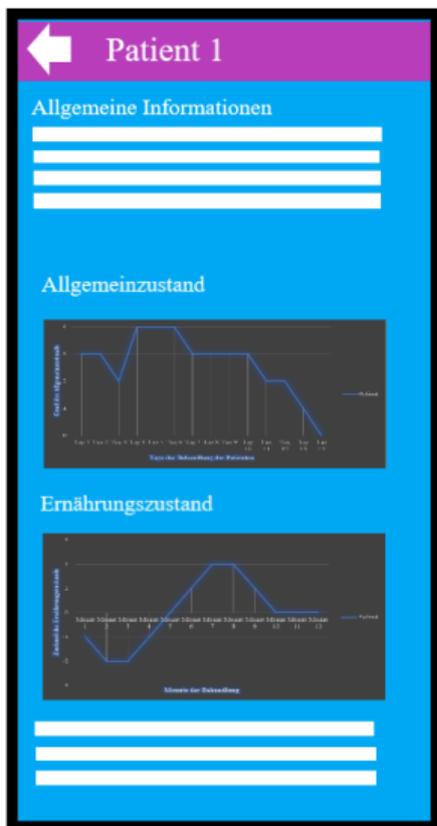


Abbildung 11 Ausgabeseite

Ausgabeseite:

Sobald ein Patient ausgewählt wurde, wird der Nutzer auf die Ausgabeseite der mobilen Applikation weitergeleitet.

Die ersten Informationen, die ausgegeben werden, sind die Allgemeinen Informationen mit den Werten aus dem Kapitel 3.2.

Im nächsten Abschnitt wird der Allgemeinzustand des ausgewählten Patienten anhand eines Liniendiagramms dargestellt.

Wenn es sich bei dem Patienten um einen Erwachsenen handelt, so wird hier der Ernährungszustand angezeigt. Dabei ist der Zustand für den Monat im Liniendiagramm erkennbar und weitere Informationen werden für den ausgewählten Monat unter dem Diagramm aufgelistet.

Sollte es sich hier doch um ein Kind handeln, so wird statt dem Ernährungszustand der Entwicklungszustand in der Form eines Säulendiagramms ausgegeben.

Sollten Werte auf dem Server fehlen oder andere Fehler ausgelöst werden, sodass keine Diagramme

gezeichnet oder Felder gefüllt werden können, so wird an der jeweiligen Stelle eine entsprechende Nachricht ausgegeben, die dem Nutzer das Problem erklären.

Über den Pfeil oben Links in der Ecke gelangt der Nutzer zurück zu der Startseite.

4 Auswahl der Programmiersprache

Im folgenden Kapitel geht es um die Auswahl der Programmiersprache, die für die App genutzt werden soll. Da die Auswahl an Programmiersprachen sehr groß ist, werden im Folgenden nur Java, Kotlin, Swift und Flutter betrachtet. Jede Programmiersprache hat ihre Stärken und Schwächen, weswegen zunächst kurz die Programmiersprachen selbst und ihre Vor- und Nachteile vorgestellt werden. Im Anschluss wird durch das Vergleichen der einzelnen Programmiersprachen die effektivste Sprache für die Entwicklung der mobilen Applikation ausgewählt.

4.1 Programmiersprachen unter Android

4.1.1 Java

Java ist eine reine objektorientierte und plattformunabhängige Programmiersprache, welches von Sun Microsystems im Jahre 1995 entwickelt wurde.²¹ Sie wird für allgemeine Zwecke verwendet. Darunter auch Spiele wie Minecraft. Des Weiteren bietet Java eine umfangreiche Klassenbibliothek an, welche viele Funktionen und Programmierungen leichter und effizienter macht.²²



Abbildung 12 Java

Früher wurde die Programmiersprache zur Erstellung für Applikationen auf dem Desktop oder direkt im Webbrowser genutzt. Diese werden jedoch immer weniger. Heutzutage wird Java eher zur Programmierung von Android Apps oder Serversystemen verwendet.

Wie zuvor erwähnt hat Java eine umfangreiche Klassenbibliothek, womit Apps, Web- und Desktop-Applikationen erstellt werden können. Des Weiteren gibt es zum einen viele Dokumentationen und zum anderen eine starke Community, da die Programmiersprache sehr verbreitet ist. Zudem kommt hinzu, dass Java eine sehr gute SDK (Software Development Kit) auch genannt JDK (Java Development Kit) besitzt und einfach zu lernen ist. Außerdem ist Java Open-Source, das heißt, die Sprache ist für jeden Nutzer kostenlos.²³

Zwar ist Java einfach zu lernen, aber dafür muss man bei dieser Sprache viel mehr Code als bei anderen Sprachen schreiben. Die Syntax ist also nicht sehr aussagekräftig. Ein weiterer Nachteil ist, dass Java Nullpointer-Exceptions hat. Zudem gibt es sowohl boilerplate code als auch keinen Read-eval-print-loop.²⁴ Dies bedeutet, dass es zum einen viele Code-Segmente gibt, die an mehreren Stellen mit kleinen bis gar keinen Änderungen wiederholt genutzt werden²⁵ und zum anderen, dass der Nutzer keine Zwischenergebnisse geliefert bekommt.²⁶

²¹ [Sbl17]

²² [Sbl17]

²³ [Jtp21]

²⁴ [Jtp21]

²⁵ [Wika22]

²⁶ [Wikb21]

Außerdem ist Java eine sehr strenge Programmiersprache, welche auch Floating-Point Errors hat.²⁷

4.1.2 Kotlin

„Kotlin ist eine statisch typisierte, höhere Programmiersprache, die Merkmale objektorientierter und funktionaler Programmierparadigmen kombiniert.“²⁸ Diese wurde von JetBrains im Jahre 2011 als Projekt eingeführt, wobei die erste stabile Version 2016 erschien. Ein Jahr später wurde Kotlin als offizielle Programmiersprache für Android anerkannt.



Abbildung 13 Kotlin

Kotlin wird hauptsächlich dafür verwendet, um Applikationen für Smartphone und weiteren Android basierten Geräten zu erstellen.

Ein Vorteil von Kotlin ist, dass die Syntax der Programmiersprache sehr kurz und aussagekräftig ist. Des Weiteren kombiniert der Code die Merkmale objektorientierter und funktionaler Programmierstile. Zudem läuft die Sprache wie Java durch die Hohe Kompatibilität der beiden Sprachen auch auf der JVM.²⁹ Hinzu kommt, dass Kotlin eine starke und aktive Community, sowie eine aktive Weiterentwicklung durch JetBrains und der Community hat.³⁰

Im Vergleich zu Java ist Kotlin schwerer zu lernen. Eine starke Community ist zwar vorhanden, jedoch ist es eine sehr kleine Community. Zudem gibt es wenige Tools um diese Programmiersprache zu lernen. Des Weiteren ist die Kompilation von Kotlin langsamer als andere Sprachen.³¹

4.2 Programmiersprachen unter IOS: Swift

Swift ist eine hauseigene Programmiersprache von Apple für die Apple Umgebung, welche 2014 veröffentlicht wurde. Die Sprache basiert auf Objective-C, ist jedoch vereinfacht worden und bestimmte Konzepte wurden verbessert.³²



Abbildung 14 Swift

Mit dieser Programmiersprache können Apps und Anwendungen für IOS- und macOS-Geräte entwickelt werden.³³

Wie Kotlin ist Swift eine Programmiersprache mit einer sehr kurzen und aussagekräftigen Syntax, wodurch der Code einfacher und sauberer für Programmieranfänger zu lesen ist. Darüber hinaus bietet Apple eine iPad-App „Swift Playgrounds“ an, die die Grundlagen spielerisch für Anfänger ohne Programmierkenntnis vermittelt und mit dem kleine

²⁷ [Jtp21]

²⁸ [Kra19]

²⁹ [Kra19]

³⁰ [Dier21]

³¹ [Kri]

³² [Sac18]

³³ [Sac18]

Anwendungen für Roboter und Drohnen geschrieben werden können.³⁴ Damit ergibt sich der Vorteil, dass der Einstieg in diese Programmiersprache einfacher ist als bei vielen anderen. Zudem ist der Support von Apple sehr groß.³⁵

Da Swift eine hausinterne Programmiersprache für die Apple-Hardware-Umgebung ist, ist es eine plattformspezifische Programmiersprache, was zur Folge hat, dass die gelernte Sprache beim Lernen einer neuen nicht viel hilft. Des Weiteren ist die Programmiersprache zwar am Anfang einfach zu lernen, jedoch ist das Meistern der Sprache viel schwieriger.³⁶

4.3 Crossplattform: Flutter

Flutter, welches von Google entwickelt und Ende 2018 als Open-Source veröffentlicht wurde, ist ein Framework zur Entwicklung von Apps und Anwendungen für verschiedene Plattformen. Die SDK basiert auf der von Google entwickelten Programmiersprache Dart. Dart ist eine Art Nachfolger von JavaScript. Insgesamt verfolgt die Strategie von Flutter „Alles ist ein Widget“ und setzt diese Strategie bei objektorientierter Programmierung konsequent bis in die Benutzeroberfläche um.³⁷



Abbildung 15 Flutter

Ein großer Vorteil von Flutter ist, dass Apps, die mit Flutter entwickelt worden sind, sich dem jeweiligen System (IOS, Android. etc.) anpassen und sich dementsprechend verhalten. Zudem ist das ganze Konzept der Widgets sehr flexibel. Des Weiteren ist die Programmiersprache für Umsteiger leicht zu erlernen. Um eine App zu bauen, werden umfangreiche Bibliotheken mit vorgefertigten UI-Elementen bereitgestellt. Außerdem hat Flutter eine „Hot Reload“-Funktion, welche das Testen während der Entwicklung beschleunigt.³⁸ Ein weiterer Vorteil ist, dass Flutter die UI mit einer eigenen Engine zeichnen kann, wodurch eine Multi-Plattform-Ausgabe ermöglicht wird und hohe Bildwiederholraten von 60 FPS am Endgerät erzielt werden können.³⁹

Zwar ist das „Alles ist ein Widget“-Konzept sehr flexibel, doch der Programmcode wird durch das Einbinden der Widgets schnell unübersichtlich. Ein weiterer Nachteil ist, dass die Sprache bis jetzt keine große Community hat, da die Verbreitung von Flutter sehr gering ist.⁴⁰

4.4 Fazit

Nun wurden mehrere Programmiersprachen vorgestellt. Dabei kam ich zu dem Schluss, dass Flutter die beste Option ist. Es kann zwar leicht unübersichtlich werden, aber dafür ist die Programmierung sehr flexibel. Zudem ist die Programmierung der App nicht so aufwändig wie mit Java oder ist nicht nur auf eine Marke von Herstellern (IOS, Android, etc.)

³⁴ [Sac18]

³⁵ [Kva21]

³⁶ [Kva21]

³⁷ [lon20]

³⁸ [lon20]

³⁹ [Sca22]

⁴⁰ [lon20]

beschränkt, wie bei Kotlin oder Swift. Außerdem passen sich Programme, Apps, Anwendungen, etc., die mit Flutter geschrieben worden sind, dem jeweiligen System an. Da die Community auch in Swift, Kotlin und Flutter sehr vergleichbar sind, ist dieser Punkt zwischen diesen Sprachen obsolet. Java hat zwar eine starke Community, aber dafür sind die Bibliotheken der UI-Elemente in Flutter um einiges umfangreicher und flexibler, während bei Java diese meist mehr programmiert werden müssen und größeren Aufwand zur Folge haben. Auch beim Testen und Kompilieren ist Flutter durch das „Hot Reload“ schneller und effizienter, während bei Kotlin beispielsweise die Kompilation sehr langsam ausfällt. Da Flutter mehr Vorteile als die restlichen besprochenen Programmiersprachen bietet, wird zu der Konzeption der App die Sprache Flutter verwendet.

5 Konzeption der App mit Flutter

Bis jetzt wurde die App analysiert und die Hintergrundinformationen erklärt. Im folgenden Kapitel wird die App mit Flutter konzipiert, sodass die App mit wenigen extra Recherchen und mithilfe dieser Konzeption umgesetzt oder implementiert werden kann.

5.1 Austausch von Daten

Die Daten werden im JSON-Format verschickt und empfangen, da JSON zum einen weit verbreitet und zum anderen einfach zu lesen ist. Auch komplexe Strukturen stellen hier keine großen Probleme dar. Des Weiteren werden vom Server nur XML- und JSON-Dateien akzeptiert, wobei der Aufwand mit JSON geringer ist.

Die Anfragen der Clients an den Server werden in JSON wie folgt definiert:

```
{
  "MessageType": "<Name_der_Funktion>",
  "MessageDaten": {} // (= Übergebene Parameter, wobei „{}“ leer bedeutet)
}
```

Abbildung 16 JSON: Anfrage an Server

Die Antworten des Servers an den Client werden in JSON wie folgt definiert:

```
{
  "Worst_Message_Type": "<Typ_der_Message>", // (= „S/I/E/W“)
  "Return": {}, // (= Übergebene Message, wobei „{}“ leer bedeutet)
  //...weitere Daten, die benötigt werden, in einer vom Nutzer vorgegebenen Form
}
```

Abbildung 17 JSON: Antwort von Server

Beim „Worst_Message_Type“ steht das „S“ für Success, „E“ für Error, „I“ für Information und „W“ für Warning.

Um diese JSON-Objekte nutzen zu können, werden Klassen und Funktionen benötigt, die die entsprechenden Objekte verarbeiten können. Dabei werden die JSON-Objekte über http-Requests vom Client zum Server, sowie vom Server zum Client, geschickt.

5.2 Verbindung zwischen der App und dem Server

Um eine sichere Verbindung zwischen dem Server und der App herzustellen muss ein VPN eingerichtet werden. Durch ein VPN wird eine sichere und stabile Verbindung zum Server aufgebaut. Es „agiert als sicherer Tunnel“⁴¹ zwischen der App auf dem Smartphone und dem Web-Server, denn die Verbindung wird durch den VPN verschlüsselt.

Um das VPN einzurichten, wird ein VPN-Client wie zum Beispiel FortiClient benötigt. Dort muss der PSK-Schlüssel, die Lokale ID, ein User und Passwort, der Verbindungsname, das Remote Gateway und der Schlüssel als IPsec VPN eingegeben werden. Aus

⁴¹ [Ble22]

unternehmensrechtlichen Gründen kann dieses Vorgehen anhand von Daten nicht genauer erläutert werden.



Abbildung 18 Verbindung zwischen der mobilen Applikation und dem Server

5.3 Serverseitig benötigte Funktionen

Auf der Seite des Servers muss eine Klasse erstellt werden, die die benötigten Daten liefert. Diese Klasse muss über ein http-Request angesprochen werden. Des Weiteren benötigt die Klasse die Funktionen „authorization“ für das Anmelden, „readPatientList“ für die Ausgabe der Patienten und „readPatient“ um die Daten des aktuellen Patienten zu lesen. Zudem muss die Klasse eine Funktion beinhalten, die die JSON-Objekte empfängt und an den Client zurücksendet.

5.4 Anmelden

Um in Flutter die Anmeldeseite zu bauen, werden Felder benötigt, wo zum einen ein Text, wie Benutzername und Passwort, geschrieben werden kann und zum anderen ein Login-Button, womit die Anmeldung erfolgt. Dazu werden die Widgets TextField und Textbutton benötigt.

Sobald der Login-Button betätigt wird, muss in Flutter ein http-Request aufgerufen werden, wie im folgenden Beispiel aufgezeigt:

```
Future<http.Response> authorization(String name, String pw) {  
  return http.post(  
    Uri.parse('<Link_zum_Server>'),  
    headers: <String, String>{  
      'Authorization': "Basic ${encodeBase64(Username:Passwort)}"  
      'Content-Type': 'application/json; charset=UTF-8',  
    },  
    body: jsonEncode(<String, String>{  
      'name': "Kilice",  
      'pw': "Passwort",  
    }  
  ),  
  );  
}
```

Abbildung 19 Beispiel http-Request in Flutter

Daraufhin wird auf der Seite des Servers die Methode „authorization“ in der erstellten Klasse aufgerufen. Dort wird mit einem Select-Befehl auf die Tabelle USR02 zugegriffen, um die Daten abzugleichen. Für jeden in der Tabelle registrierten Benutzernamen ist das Kennwort aus der Datenbanktabelle zu entschlüsseln.

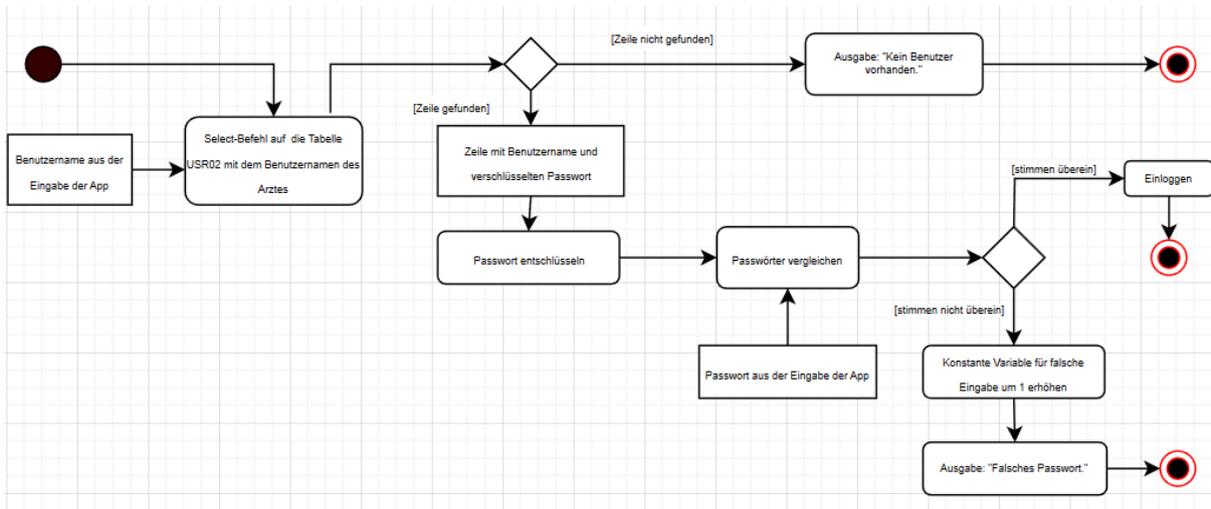


Abbildung 20 Aktivitätsdiagramm für die Anmeldung auf Seiten des Servers

5.5 Ausgabe der Patientenliste

Für die Ausgabe einer Liste auf einem Bildschirm in Flutter werden die Widgets „ListView“ und „ListTile“ benötigt. Die „ListView“ dient zum Anzeigen einer Liste und „ListTile“ arbeitet mit den Zeilen (Row-Funktion) und Spalten (Column-Funktion). Des Weiteren gibt es bei „ListTile“ „children“, die sich baumartig verzweigen.

Um die Liste der Patienten zu bekommen, muss zunächst die Einrichtung mit der ID des behandelnden Arztes mit einem http-Request an den Server an die Methode „readPatientList“ geschickt werden. Dort wird ein Select-Befehl auf die Datenbanktabelle NFPZ ausgeführt, sodass eine Liste an bearbeiteten Fällen in einer internen Tabelle gespeichert werden. Im nächsten Schritt wird durch die Fallliste und der Datenbanktabelle NFAL die Patientenliste ermittelt und in einer internen Tabelle gespeichert. Zudem wird aus der Tabelle NGPA der vollständige Name des behandelnden Arztes gezogen, um diesen anzuzeigen. Sobald alle Daten vollständig sind, werden diese als JSON zurück an die mobile Applikation geschickt, welches diese über die zuvor besprochenen Widgets anzeigen lässt.

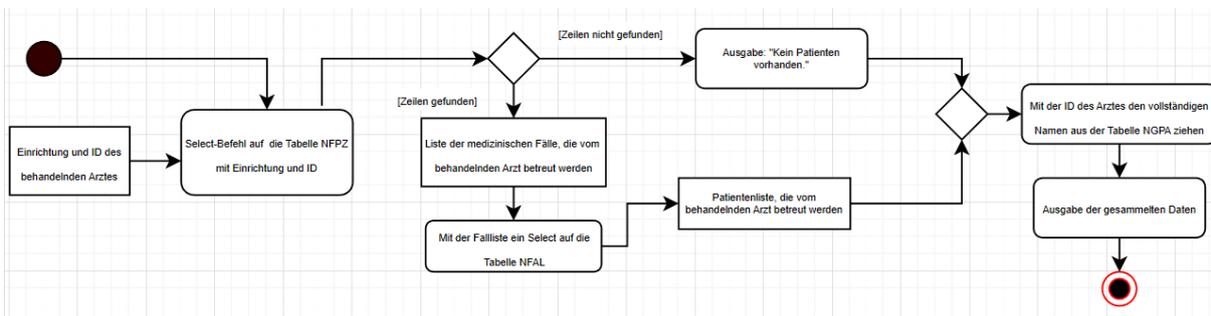


Abbildung 21 Aktivitätsdiagramm für das Lesen der Patientenliste auf Seiten des Servers

5.6 Visualisierung der Daten

Zur Visualisierung der Daten in Form von Diagrammen wird ein Package benötigt. Es werden mehrere angeboten, hier wird das Package „fi_chart“ 0.55.2⁴² benutzt. Dieses Package bietet verschiedene Diagramme und Funktionen an, zum Beispiel Linien- und Säulendiagramme. Linien- und Säulendiagramme werden in den Klassen „LineChart“ und „BarChart“ erstellt. Auf die Anpassung der X- und Y-Achsen ist zu achten, insbesondere wenn den Tabellen neue Werte hinzugefügt werden.

Für die Darstellung der Allgemeinen Informationen eines Patienten kann die Klasse „Text“ verwendet werden.

Zur Darstellung der Informationen wird mit einem http-Request die Patientenummer und die Fallnummer des ausgewählten Patienten und die Einrichtung, in der dieser sich befindet, an die Methode „readPatient“ der Klasse auf dem Server verschickt. Um die Allgemeinen Informationen darzustellen, werden die Informationen mit einem Select-Befehl aus der Datenbanktabelle NPAT gelesen und gespeichert. Des Weiteren werden die Informationen aus den drei erstellten Dokumenten mit der Schema-ID „APPALLG“, „APPERNA“ und „APPENTW“ benötigt. Diese werden gelesen und gespeichert. Sobald alle Daten vollständig sind, werden diese an die mobile Applikation weitergeleitet, wo die Informationen ausgewertet und in den Listen und Diagrammen dargestellt werden.

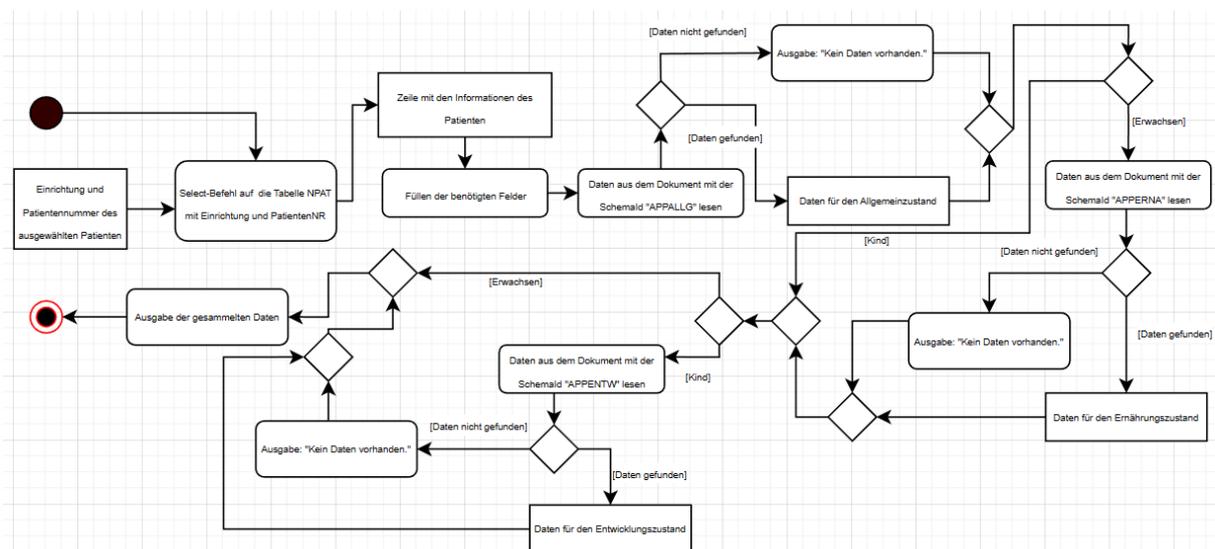


Abbildung 22 Aktivitätsdiagramm für das Lesen der Patientendaten auf Seiten des Servers

⁴² [Pub22]

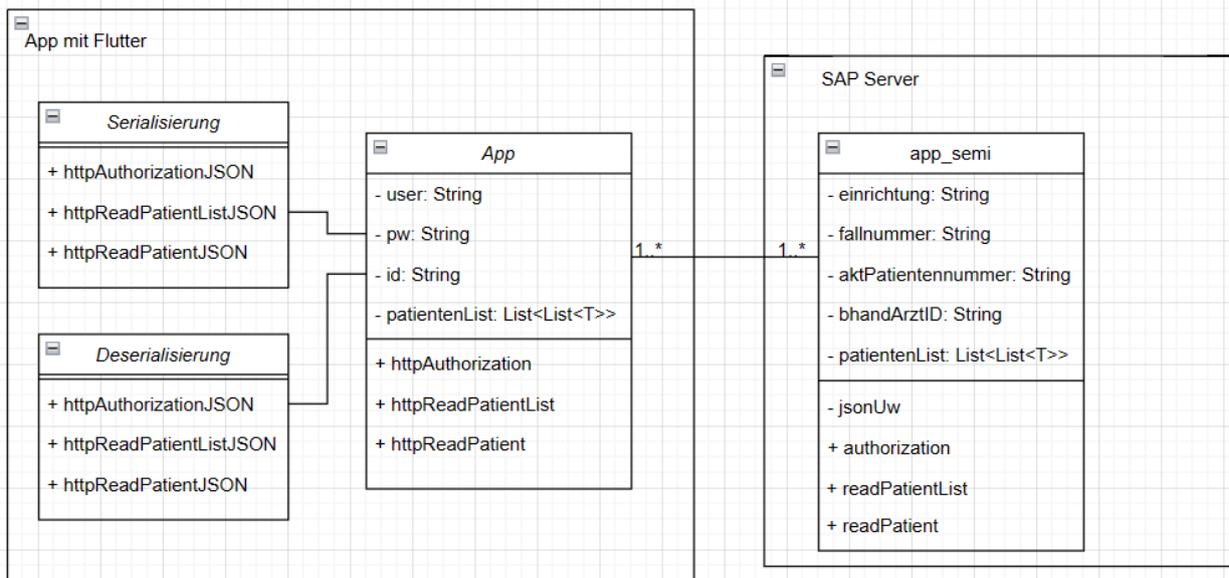


Abbildung 23 min. Klassendiagramm

6 Zusammenfassung und Ausblick

Diese Arbeit entwickelt auf Grundlage einer Analyse ein Konzept für eine mobile Applikation, welche Informationen zu einem Patienten von einem behandelnden Arzt in einem medizinischen Fall nutzerfreundlich in Listen, Tabellen und Graphiken darstellt. Verschiedene Programmiersprachen wurden für die Entwicklung dieser App analysiert und bewertet. Flutter erscheint als beste Programmiersprache für die mobile Applikation, da ein Hybrid den Aufwand zur Entwicklung einer eigenen App für Apple-Nutzer und Android-Nutzer vermeidet.

Die grundlegenden Informationen aus der Organisation, Medizin und der Informatik werden erläutert, um ein Verständnis hierfür entwickeln zu können.

Für die Visualisierung werden Diagramme dargestellt und Diagrammtypen begründet ausgewählt.

Empfehlenswert ist angesichts der persönlichen Daten eine stärkere Fokussierung auf die Sicherheit. Eine ausführliche Würdigung der Datensicherheit einer mobilen Applikation übersteigt den Umfang dieser Seminararbeit.

Die konzipierte mobile Applikation kann nur Daten und Informationen anzeigen. Eine erste Erweiterung könnte das Erfassen von Daten und deren Übermittlung an den Server sein. Auch sollte der Umfang der anzeigbaren Daten über den Allgemein-, Ernährungs- und Entwicklungszustand hinausgehen.

Die Grundkonzeption ist vielfältig erweiterbar und ermöglicht ein umfangreiches Informationsangebot und Datenerfassungsmöglichkeiten.

Hier wurde ein grundlegendes Konzept entwickelt, um die Effizienz ärztlicher Behandlungen durch eine mobile Applikation zu steigern.

7 Literaturverzeichnis

- [Kra19] J.-D. Kranz, „it-talents.de,“ 22 Dezember 2019. [Online]. Available: <https://it-talents.de/it-wissen/programmieren/was-ist-kotlin/>.
- [Dier21] A. Diermann, „www. adesso-mobile.de,“ Competence Center, 05 Februar 2021. [Online]. Available: <https://www.adesso-mobile.de/android-app-programmierung/kotlin-vorteile/>.
- [Gesä20] „gesund.bund.de,“ Bundesministerium für Gesundheit, 20 August 2020. [Online]. Available: <https://gesund.bund.de/patientenakte>.
- [Rei18] „reimbursement.institute,“ Reimbursement Institute, 15 Mai 2018. [Online]. Available: <https://reimbursement.institute/blog/stationaer-teilstationaer-oder-ambulant/#>.
- [MNSF] M. Habben, N. Van den Höfel, S. Merz und F. Dr. Antwerps, „flexikon.doccheck.com,“ 11 Januar 2022. [Online]. Available: <https://flexikon.doccheck.com/de/Allgemeinzustand>.
- [Wik21] „de.wikipedia.org,“ 16 November 2021. [Online]. Available: <https://de.wikipedia.org/wiki/Allgemeinzustand>.
- [Wik22] „de.wikipedia.org,“ 02 Oktober 2022. [Online]. Available: [https://de.wikipedia.org/wiki/Anamnese_\(Pflege\)](https://de.wikipedia.org/wiki/Anamnese_(Pflege)).
- [Neu10] J. Neuhaus, „www.aerzteblatt.de,“ 2010. [Online]. Available: <https://www.aerzteblatt.de/archiv/74001/Elektronische-Fallakte-und-Patientenakte-Auf-dem-Weg-zu-einer-einheitlichen-Loesung>.
- [Bür22] Bürgerliches Gesetzbuch (BGB), § 630f Dokumentation der Behandlung, Abschnitt 3, NRW, 2022.
- [Kra18] J.-D. Kranz, „it-talents.de,“ 15 März 2018. [Online]. Available: <https://it-talents.de/it-wissen/prozedurale-vs-objektorientierte-programmierung/>.
- [Kra+18] J.-D. Kranz, „it-talents.de,“ 12 September 2018. [Online]. Available: <https://it-talents.de/it-wissen/was-ist-funktionale-programmierung/>.
- [Wik+22] „de.wikipedia.org,“ 01 Februar 2022. [Online]. Available: https://de.wikipedia.org/wiki/Java_Virtual_Machine.
- [Ant14] F. Dr. Antwerpes, „flexikon.doccheck.com,“ 29 Juli 2014. [Online]. Available: <https://flexikon.doccheck.com/de/Ern%C3%A4hrungszustand>.
- [Gra08] K. Grashoff, „ÄrzteZeitung.de,“ 11 September 2008. [Online]. Available: <https://www.aerztezeitung.de/Medizin/Den-Ernaehrungszustand-richtig-beurteilen-357380.html>.
- [StiNe] „www.stiftungnetz.ch,“ stiftungNetz, [Online]. Available: <https://www.stiftungnetz.ch/entwicklung-des-kindes/>.

- [Jtp21] „www.javatpoint.com,“ javaTpoint, 2021. [Online]. Available: <https://www.javatpoint.com/flutter-vs-java>.
- [Wikb21] „de.wikibrief.org,“ 25 Juni 2021. [Online]. Available: https://de.wikibrief.org/wiki/Read%E2%80%93eval%E2%80%93print_loop.
- [Kri] „krify.co,“ [Online]. Available: <https://krify.co/advantages-and-disadvantages-of-kotlin/>.
- [Sac18] S. Augsten und C. , „www.dev-insider.de,“ 17 August 2018. [Online]. Available: <https://www.dev-insider.de/was-ist-apple-swift-a-735481/>.
- [Wika22] „de.wikipedia.org,“ 12 April 2022. [Online]. Available: <https://de.wikipedia.org/wiki/Boilerplate-Code>.
- [Dmrz] „www.dmrz.de,“ Deutsches Medizinrechenzentrum GmbH, [Online]. Available: <https://www.dmrz.de/wissen/lexikon/was-ist-eine-elektronische-fall-akte-efa-genau>.
- [Kva21] A. Kvasnevska, „www.miquido.com,“ 08 Dezember 2021. [Online]. Available: <https://www.miquido.com/blog/flutter-vs-kotlin-vs-swift/>.
- [Sbl17] S. Block, „www.inztitut.de,“ 08 April 2017. [Online]. Available: <https://www.inztitut.de/blog/glossar/java/>.
- [Ach22] P. Achilles, „praxistipps.chip.de,“ 20 Mai 2022. [Online]. Available: https://praxistipps.chip.de/was-ist-ein-widget-einfach-erklaert_42003.
- [Ion20] „www.ionos.de,“ 28 Juli 2020. [Online]. Available: <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-flutter/>.
- [Ble22] J. von Bleichert, „www.experte.de,“ 14 Februar 2022. [Online]. Available: <https://www.experte.de/vpn/vpn-sicherheit>.
- [Pub22] „pub.dev,“ 23 September 2022. [Online]. Available: https://pub.dev/packages/fl_chart.
- [Sca22] „scand.de,“ 5 Oktober 2022. [Online]. Available: <https://scand.de/unternehmen/blog/flutter-hauptvorteile/>.