

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät für Design, Medien und Information
Department Medientechnik
Media Systems



Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Konzeption und prototypische Implementierung einer Anwendungsschnittstelle zur Ausschreibung eines verschreibungspflichtigen Rezeptes in digitaler Form

Vorgelegt von

Mona Patricia Röttger

2220740

Erstprüfer: Prof. Dr. Andreas Pläß

Zweitprüfer: Prof. Dr.-Ing. Sabine Schumann

Hamburg, 11.03.2019

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	III
Listings.....	IV
1 Einleitung.....	1
1.1 Aktualität und Relevanz des Themas	1
1.2 Motivation und Zielsetzung.....	2
1.3 Gliederung der Arbeit	3
2 Theoretische Grundlagen	4
2.1 Arzneimittel	4
2.2 Rezeptarten.....	6
2.3 Aufbau des Kassenrezeptes.....	8
2.4 Aktueller Ablauf der Rezept-Ausstellung	10
3 Anforderungsanalyse.....	12
3.1 Digitalisierung des Gesundheitswesens	12
3.1.1 Telematikinfrastruktur	12
3.1.2 E-Rezept	13
3.1.3 Der Einsatz von Applikationen im Gesundheitswesen	14
3.2 Anwendungsfälle	15
4 Konzeption.....	17
4.1 Ablauf des Digitalen Rezeptes	17
4.2 Technische Rahmenbedingungen	19
4.3 Aufbau und Bedienung der Anwendungen	22
4.3.1 App	22
4.3.2 Web-Anwendung.....	23
4.3.3 Kassen-Anwendung	24
4.4 Prozesse	26
4.4.1 Rezept-Anforderung	26
4.4.2 Rezept-Ausstellung	27
4.4.3 Rezept-Einlösung	28

5	Realisierung des Prototyps	31
5.1	Technisches Konzept	31
5.1.1	Projektübersicht	31
5.1.2	Datenbank	32
5.1.3	App	36
5.1.4	Web-Anwendung	41
5.2	Implementierung	45
5.2.1	API	45
5.2.2	App	49
5.2.3	Web-Anwendung	52
6	Fazit und Ausblick	54
	Literaturverzeichnis	V

Abbildungsverzeichnis

Abbildung 1: Kassenrezept Muster, https://www.meine-hautapotheke.de/hautchecker.html , letzter Zugriff am 07.01.2019.....	8
Abbildung 2: Ablauf des Digitalen Rezeptes, eigene Darstellung.....	17
Abbildung 3: Visualisierung der App, eigene Darstellung.....	22
Abbildung 4: Visualisierung der Web-Anwendung, eigene Darstellung.....	23
Abbildung 5: Visualisierung der Kassen-Anwendung, eigene Darstellung	24
Abbildung 6: Visualisierung der Rezept-Anforderung, eigene Darstellung	26
Abbildung 7: Visualisierung der Rezept-Ausstellung, eigene Darstellung	27
Abbildung 8: Visualisierung der Rezept-Einlösung, eigene Darstellung	28
Abbildung 9: Visualisierung der Abrechnung eines Digitalen Rezeptes, eigene Darstellung	29
Abbildung 10: Zusammenspiel der technischen Komponenten, eigene Darstellung	31
Abbildung 11: ERM, eigene Darstellung	34
Abbildung 12: Sequenz-Diagramm der App: Login und grober Ablauf, eigene Darstellung...	37
Abbildung 13: Sequenz-Diagramm der App: Ablauf der Rezept- und Anforderung- Darstellung, eigene Darstellung.....	39
Abbildung 14: Komponenten-Hierarchie der Web-Anwendung, eigene Darstellung.....	42
Abbildung 15: Sequenz-Diagramm der Web-Anwendung, eigene Darstellung	43

Listings

Listing 1: API: CORS-Einstellungen in getRequires.php	45
Listing 2: API: Datenbankverbindung in dbconnection.php	46
Listing 3: API: JSON-Daten in getRequires.php.....	46
Listing 4: API: SQL-Befehl in getRequires.php	47
Listing 5: API: SQL-Befehle in releaseRecipe.php.....	47
Listing 6: API: mysqli_result()-Methode in getRequires.php.....	48
Listing 7: App: AndroidManifest.xml.....	49
Listing 8: App: ShowRecipeActivity, onCreate().....	49
Listing 9: App: BackgroundHandler, doInBackground()	50
Listing 10: App: BackgroundHandler, setAPIConnection().....	50
Listing 11: App: BackgroundHandler, postData()	51
Listing 12: App: BackgroundHandler, receivingData()	51
Listing 13: App: BackgroundHandler, onPostExecute()	51
Listing 14: App: ShowRecipeActivity, getAsyncResult().....	52
Listing 15: Web-Anwendung: RequireList.js	53
Listing 16: Web-Anwendung: RequireList.js, render()	53

Abstract

In this bachelor's thesis a concept for the prescription of drugs is created and a corresponding prototype is developed. Beginning with the definition of the relevant terms and an explanation of the common routines concerning the German health system this paper combines information about e-health and digitalization. The whole process of the drug prescription up to its redeem is described. For an efficient use of the developed concept the provided software functions are recorded and prioritized for their practical implementation. This is followed by the implementation of the prototype. Therefor only the recipe exhibition is taken into consideration. For this, the implementation of an Android app, a website based on React and a relational database, as well as their connection in the form of an API are discussed. This thesis was created in the period from 01.01.2019 to 12.03.2019.

Kurzfassung

In dieser Bachelorarbeit wird ein Konzept für die digitale Verordnung von verschreibungspflichtigen Medikamenten und die Umsetzung eines Prototyps ausgearbeitet.

Als Grundlage für das Konzept werden zuerst aktuelle Begrifflichkeiten, Fachausdrücke, Abläufe im Gesundheitssystem erläutert und Informationen zum Thema E-Health und Digitalisierung zusammengetragen.

Im Konzeptteil wird der gesamte Ablauf, von der Rezept-Ausstellung bis zur -einlösung, erläutert. Für eine optimale Nutzung der Konzeption werden die zur Verfügung gestellten Software-Funktionen festgehalten und für die praktische Umsetzung priorisiert.

Im Anschluss erfolgt die Implementierung des Prototyps. Hier wird ausschließlich die Rezept-Ausstellung realisiert. Dazu wird die Umsetzung einer Android App, einer Website basierend auf React und eine relationale Datenbank, so wie ihre Anbindung in Form einer API, behandelt. Diese Arbeit wurde im Zeitraum vom 01.01.2019 bis 12.03.2019 erstellt.

1 Einleitung

1.1 Aktualität und Relevanz des Themas

„Die Zahl der Ärztinnen und Ärzte in Deutschland steigt, aber wer nur Köpfe zählt, macht es sich zu einfach. Die Realität ist komplexer. Uns fehlen Arztstunden. Und wenn wir nicht endlich entschieden gegensteuern und mehr Ärzte ausbilden, dann wird sich dieser Mangel verschärfen.“, merkt der Präsident der Bundesärztekammer (BÄK) Prof. Dr. Frank Ulrich Montgomery bezogen auf die Ärztestatistik für das Jahr 2017 an.¹ Bereits aus den im Jahre 2016 erhobenen Daten geht hervor, dass die Zahl der gemeldeten Ärzte² im Gegensatz zu den hohen Behandlungszahlen um nur 2,1% geringfügig gestiegen ist. Allein im Rahmen der ärztlichen Versorgung außerhalb der Kliniken kommt es jedes Jahr zu über einer Milliarde Arzt-Patienten-Kontakten.³ Des Weiteren ist zu vermuten, dass die demografische Entwicklung innerhalb Deutschlands zu einer zusätzlichen Erhöhung des Versorgungsbedarfs führen könnte.

Durch die hohe Belastung der Ärzte und die daraus resultierenden überfüllten Wartezimmer, sehen sich viele Praxen bereits dazu gezwungen, einen Patienten-Aufnahmestopp zu verhängen. Für den Patienten führt dies zu einem erhöhten Aufwand bezüglich der Suche nach einem geeigneten Arzt.

Viele Menschen mit chronischen Erkrankungen sind zur Medikamentenverschreibung auf regelmäßige Arztbesuche angewiesen. Diese Patienten benötigen nicht jedes Mal eine erneute ärztliche Untersuchung. Trotzdem ist auch ohne erneute Untersuchung der zeitliche Aufwand gleichzusetzen, weil der Patient das Rezeptformular in Papierform vor Ort anfordern und dann persönlich abholen muss.

Um die Praxen und Patienten auf lange Sicht zu entlasten und eine effiziente Versorgung zu gewährleisten, werden immer mehr Prozesse im Gesundheitswesen durch telemedizinische Technologien ausgetauscht oder erweitert.

¹ Bundesärztekammer, 31.12.2017, Ärztestatistik 2017, <https://www.bundesaerztekammer.de/ueber-uns/aerztestatistik/aerztestatistik-2017/>, letzter Zugriff am 18.01.2019

² In dieser Arbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

³ Bundesärztekammer, 31.12.2016, Ärztestatistik 2016: Die Schere zwischen Behandlungsbedarf und Behandlungskapazitäten öffnet sich, <https://www.bundesaerztekammer.de/ueber-uns/aerztestatistik/aerztestatistik-2016/>, letzter Zugriff am 18.01.2019

1.2 Motivation und Zielsetzung

Aus diesem Grund möchte ich im Rahmen dieser Bachelorarbeit die Beschaffungsvorgänge rund um das ärztliche Rezept digitalisieren. Dafür wird die *Rezept-Ausschreibung* komplett betrachtet. Hierzu umfasst dieser Begriff den gesamten Prozess beginnend mit der Rezept-Anforderung, über die Rezept-Ausstellung bis hin zur Rezept-Einlösung. Für die Realisierung der jeweiligen Teil-Prozesse werden verschiedene Software-Anwendungen benötigt.

Zunächst eine Smartphone-Applikation, im weiteren Verlauf als App bezeichnet, über die der Patient sein Rezept anfordern kann. Hinzu kommt eine webbasierte Anwendung für die Arzt-Praxen, um die Rezept-Anforderungen zu bearbeiten und das Rezept in digitaler Form dem Patienten in der App zur Verfügung zu stellen. Für eine sichere Einlösung dieses Rezeptes brauchen auch die Apotheken eine passende Anwendungsschnittstelle.

Der Einsatz des sogenannten "Digitalen Rezeptes" soll in Zukunft für alle Beteiligten die bisherigen Abläufe erleichtern und den Zeitaufwand verringern.

Ziel dieser Arbeit ist es, ein Konzept rund um den Beschaffungsvorgang eines verschreibungspflichtigen Arzneimittels in digitaler Form zu entwerfen und eine Schnittstelle des erarbeiteten Konzeptes prototypisch zu implementieren. Unter Berücksichtigung der aktuellen Gesetze und Technologien im Gesundheitswesen wird neben der Rezept-Ausstellung auch die Rezept-Einlösung konzeptionell erarbeitet. Am Ende soll ein lauffähiger Prototyp in Form einer Android-App vorhanden sein, über den die Patienten das sogenannte "Digitale Rezept" auf dem Smartphone anfordern und anschließend in der Apotheke vorzeigen können.

Für die Umsetzung der prototypischen Rezept-Ausstellung, wird neben der App eine auf React basierende Web-Anwendung realisiert, über die der Arzt das Rezept bearbeiten kann. Für eine optimale Speicherung der Daten, wird eine relationale Datenbank realisiert. Für den Informationsaustausch zwischen Arzt und Patient wird eine Anbindung an die Datenbank in Form einer API (Application Programming Interface) zur Verfügung gestellt.

Die Digitalisierung der ärztlichen Verordnung soll in Zukunft das herkömmliche ärztliche Rezept in Papierform vollständig ersetzen und den Ablauf für alle Beteiligten vereinfachen.

1.3 Gliederung der Arbeit

Die vorliegende Arbeit ist in sechs Kapitel unterteilt. Neben der *Einleitung*, die einen ersten Überblick in das behandelte Thema gewährt und die Ziele der Arbeit definiert, werden auf den folgenden Seiten die *Grundlagen* (siehe Kapitel 2. *Grundlagen*) zur Ausschreibung eines Rezeptes beschrieben. Der Einsatz von verschiedenen Rezeptarten und der aktuelle Ablauf, von der Ausstellung bis hin zur Einlösung eines Rezeptes, werden für den Einstieg festgehalten. Im dritten Kapitel *Anforderungsanalyse* (siehe Kapitel 3. *Anforderungsanalyse*) wird über die gegenwärtige Digitalisierung im Gesundheitswesen informiert. Die Arztpraxen werden technisch immer besser ausgestattet, was neue Möglichkeiten für den Einsatz digitaler Technologien schafft. Anschließend werden die spezifischen Anforderungen zusammengetragen und priorisiert.

Im darauffolgenden Kapitel wird das *Konzept* (siehe Kapitel 4. *Konzeption*) rund um die Rezept-Ausschreibung erläutert. Anhand von Wireframes werden Aufbau und Funktion der benötigten Software-Anwendungen beschrieben und Prozess-Abläufe dargestellt.

Innerhalb des fünften Kapitels, der *Realisierung des Prototyps* (siehe Kapitel 5. *Realisierung des Prototyps*), wird ausschließlich die Rezept-Ausstellung behandelt. Die technische Konzeption der App und der Web-Anwendung sowie die Implementierung der Datenbankanfragen werden dabei detailliert erklärt.

Abschließend werden im sechsten und letzten Kapitel *Fazit und Ausblick* (siehe Kapitel 6. *Fazit und Ausblick*) Erkenntnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche Erweiterungen des Prototyps gegeben.

2 Theoretische Grundlagen

Hat eine Person gesundheitliche Beschwerden, dann sucht sie in der Regel einen Arzt auf. Der Arzt erstellt eine Diagnose und entscheidet dann, welches Arzneimittel zur Behandlung benötigt wird und stellt in den meisten Fällen eine ärztliche Verordnung, auch Rezept genannt, aus. Doch in welchen Fällen wird ein Rezept benötigt und warum?

2.1 Arzneimittel

Arzneimittel, auch Medikament genannt, sind der Definition nach *“insbesondere Stoffe und Zubereitungen aus Stoffen, die dazu dienen, durch Anwendung bei Mensch oder Tier Krankheiten, Leiden, Körperschäden oder Beschwerden zu heilen, zu lindern, zu verhüten oder zu erkennen sowie Stoffe, die der Diagnose dienen oder seelische Zustände beeinflussen.”*⁴

*“Das seit 1978 geltende "Gesetz über den Verkehr mit Arzneimitteln" (kurz "Arzneimittelgesetz" – AMG) regelt unter anderem, wie ein Arzneimittel beschaffen sein muss, welche Voraussetzungen es erfüllen muss und wie seine Sicherheit zu garantieren ist.”*⁵

In Deutschland wird in der Regel zwischen vier Arten von Arzneimitteln unterschieden⁶:

- Frei verkäufliche Arzneimittel: Diese Arzneimittel, auch oft OTC-Arzneimittel (“Over the Counter”) genannt, sind rezeptfrei und können auch außerhalb von Apotheken z.B. in Drogerie- und Supermärkten erworben werden. Zu diesen Medikamenten zählen Vitaminpräparate und bestimmte Arzneimittel (z.B. Heilerde, Pflanzenpresssäfte usw.), die im Arzneimittelgesetz (AMG § 44) explizit aufgelistet sind.⁷
- Apothekenpflichtige Arzneimittel: Diese Arzneimittel sind nur in der Apotheke erhältlich. Es ist ausreichend, wenn der Apotheker den Patienten über die Anwendung und Nebenwirkungen des Medikaments informiert. Dazu zählen oft Schmerz- und Erkältungsmittel, bei denen die Rezeptpflicht entfällt.⁸
- Verschreibungspflichtige Arzneimittel: Diese Arzneimittel unterliegen einer Verschreibungspflicht, auch Rezeptpflicht genannt, gemäß dem Arzneimittelgesetz (AMG § 48). Dazu gehören Arzneimittel, die im AMG gelistet sind und eine regelmäßige Überwachung und Anpassung der Dosierung benötigen und Medikamente mit unbekannten

⁴ Vgl. AOK Die Gesundheitskassen, 2016, Arzneimittel, https://aok-bv.de/lexikon/a/index_00197.html, letzter Zugriff am 11.01.2019.

⁵ Vgl. Stiftung Warentest, 2018, Arzneimittel - Definition, <https://www.test.de/medikamente/begriffe/>, letzter Zugriff am 11.01.2019.

⁶ Vgl. AOK Die Gesundheitskassen, 2016, Arzneimittel, https://aok-bv.de/lexikon/a/index_00197.html, letzter Zugriff am 11.01.2019.

⁷ vgl. Bundesministerium der Justiz und für Verbraucherschutz, Gesetz über den Verkehr mit Arzneimitteln (Arzneimittelgesetz - AMG) § 44 Ausnahme von der Apothekenpflicht, https://www.gesetze-im-inter-net.de/amg_1976/___44.html, letzter Zugriff am 11.01.2019

⁸ Vgl. Ebenda

Wirkungen, die die Gesundheit auch bei bestimmungsgemäßem Gebrauch gefährden können. Sie sind nur mit einer ärztlichen Verordnung in einer Apotheke erhältlich, also gleichzeitig immer apothekenpflichtig und benötigen ärztliche Überwachung.⁹

- Betäubungsmittel: Arzneimittel, die im Betäubungsmittelgesetz (BtMG) aufgelistete Stoffe beinhalten, benötigen eine besondere Verordnung und sind nur in Apotheken erhältlich. Sie gelten demnach als verschreibungs- und apothekenpflichtig. Hier ist eine strenge Überwachung nötig, da eine missbräuchliche Verwendung zur unmittelbaren Gefährdung der Gesundheit führen kann.¹⁰

Ob ein Medikament zugelassen und in welche Kategorie es nach der Zulassung eingeordnet wird, entscheidet das Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM).¹¹

Für die Zulassung eines Arzneimittels müssen von einem pharmazeutischen Unternehmen Unterlagen an das BfArM eingereicht werden, mit denen die Qualität, die Unbedenklichkeit und die Wirksamkeit des Arzneimittels belegt wird. Das BfArM muss die Unterlagen prüfen und alle Informationen und Hinweise für eine sichere Anwendung in so genannte Gebrauchs- (für Patienten) und Fachinformationen (für Ärzte) zusammenfassen. Dabei steht die Gewährleistung einer sicheren Anwendung der Arznei immer im Vordergrund.

Wurde dem Hersteller eine Zulassung erteilt, gilt diese vorläufig für fünf Jahre.¹² Anschließend wird das Arzneimittel anhand des medizinischen Nutzens, den möglichen Risiken und Nebenwirkungen in eine der oben genannten Kategorien zugeordnet.

⁹ Vgl. Bundesministerium für Gesundheit, 13.02.2018, Zugang zu Arzneimitteln, <https://www.bundesgesundheitsministerium.de/themen/krankenversicherung/online-ratgeber-krankenversicherung/arznei-heil-und-hilfsmittel/zugang-zu-arzneimitteln.html>, letzter Zugriff am 11.01.2019.

¹⁰ Vgl. Bundesinstitut für Arzneimittel und Medizinprodukte, 2013 und 2018, Betäubungsmittel, https://www.bfarm.de/DE/Bundesopiumstelle/Betaeubungsmittel/_node.html, letzter Zugriff am 11.01.2019.

¹¹ Vgl. Stiftung Warentest, 01.01.2018, Zulassung, <https://www.test.de/medikamente/begriffe/zulassung/>, letzter Zugriff am 11.01.2019.

¹² Vgl. Bundesinstitut für Arzneimittel und Medizinprodukte, 2013, Arzneimittelzulassung, https://www.bfarm.de/DE/Buerger/Arzneimittel/Arzneimittelzulassung/_node.html, letzter Zugriff am 11.01.2019.

2.2 Rezeptarten

Für die vielen verschiedenen Arzneimittel und deren Gebrauch und Erwerb, kann der Arzt die dazugehörigen Rezepte ausstellen. Diese verschiedenen Rezeptarten werden hauptsächlich durch unterschiedliche Farben gekennzeichnet und machen damit auf die spezielle Bedeutung aufmerksam. Der Patient kann das Rezept in einer Apotheke vorlegen und das verordnete Medikament kaufen bzw. abholen.¹³

- Rosa Rezept: Das sogenannte Kassenrezept ist rosa bzw. rot. Der Arzt verwendet dieses Rezeptformular, wenn der Patient gesetzlich krankenversichert ist und die Krankenkasse den Großteil der Kosten übernimmt. Der Versicherte muss mindestens 5 Euro und maximal 10 Euro pro Medikament zuzahlen. Grundsätzlich ist das klassische Kassenrezept drei Monate gültig. Doch nur bei einer Einlösung innerhalb der ersten vier Wochen, nach dem Tag der Ausstellung, werden die Kosten durch die Krankenversicherung erstattet.
- Blaues Rezept: Das blaue Rezeptformular wird meistens zur Arzneimittel-Ausschreibung für privat versicherte Patienten verwendet. Der Aufbau ist dem Kassenrezept nachempfunden. Doch auch gesetzlich Versicherte bekommen das blaue Formular, sollte das geforderte Arzneimittel nicht unter den Leistungskatalog der jeweiligen Versicherung fallen. In beiden Fällen muss der Patient das verordnete Medikament zunächst selbst bezahlen. Sobald der Privat-Patient die Quittung bei seiner Krankenkasse einreicht, werden ihm die Kosten zurückerstattet. In der Regel ist auch dieses Rezeptformular drei Monate einlösbar.
- Grünes Rezept: Für Empfehlungen für nicht verschreibungspflichtige Arzneimittel stellt der Arzt ein grünes Rezeptformular aus. In der Regel übernehmen gesetzliche Krankenkassen seit der Gesundheitsreform 2004 keine Kosten mehr für rezeptfreie Medikamente, somit muss der Patient selbst die Kosten tragen. Dieses Rezept ist unbegrenzt gültig und einlösbar.
- Gelbes Rezept: Auch Betäubungsmittelrezept, kurz BtM-Rezept, genannt wird bei der Verschreibung von starken Medikamenten, die dem Gesetz über den Verkehr mit Betäubungsmitteln (Betäubungsmittelgesetz-BtMG) unterliegen, verwendet. Dieses Rezept besteht aus drei Teilen, dem Original und zwei Durchschlägen, und ist nummeriert. Die Arztpraxis und die Apotheke erhalten jeweils einen Durchschlag zur Dokumentation. Auch hier übernimmt die Krankenkasse einen Großteil der Kosten. Das BtM-Rezept ist nur sieben Tage gültig.

¹³ Vgl. Die Techniker, 2018, Welche Rezepte gibt es?, <https://www.tk.de/techniker/service/leistungen-und-mitgliedschaft/leistungen/arzneimittel/arzneimittel-auf-rezept/rezepte-rezept-unterschiede-2002740>, letzter Zugriff am 11.01.2019.

- Weißes Rezept: Zuletzt gibt es noch das zweiteilige T-Rezept. Dieses wird verwendet bei der Verordnung von Medikamenten zur Behandlung von Knochenmarkserkrankungen, die Wirkstoffe wie Thalidomid, Lenalidomid oder Pomalidomid enthalten. Diese Medikamente sind potentiell fruchtschädigend (teratogen, daher die Bezeichnung) und dürfen nicht während einer Schwangerschaft eingenommen werden. Das T-Rezept ist bis zu sechs Tage nach der Ausstellung gültig. Die Apotheke muss den Durchschlag an das Bundesinstitut für Arzneimittel und Medizinprodukte (BfArM) schicken.¹⁴

¹⁴ Vgl. Offizielles Gesundheitsportal der deutschen ApothekerInnen, 2019, Rezepte haben verschiedene Farben und Bedeutungen, <https://www.aponet.de/wissen/arzneimitteldatenbank/arzneimittel/rezeptarten.html>, letzter Zugriff am 11.01.2019 & Tarif-Testsieger, 2019, Kassenrezept: Gültigkeit und Zuzahlung beachten, <https://www.tarif-testsieger.de/gesetzliche-krankenversicherung/kassenrezept/>, letzter Zugriff am 11.01.2019.

2.3 Aufbau des Kassenrezeptes

Damit ein Rezept gültig ist, gibt es laut AMG gewisse Angaben, die auf keinem Rezept fehlen dürfen. Die meisten Deutschen sind gesetzlich krankenversichert.¹⁵ Deshalb fokussiert die Implementierung der vorliegenden Arbeit lediglich die Umsetzung des Kassenrezeptes. Die anderen Formen von Rezepten können zukünftig einbezogen werden. Die Angaben dieser Rezepte können teilweise abweichen. Die grundlegenden Angaben und Erweiterungen werden exemplarisch anhand des Kassenrezeptes erklärt.

1	X	Krankenkasse bzw. Kostenträger AOK Rheinland-Pfalz	2	8	9	
	X	Name, Vorname des Versicherten Mustermann Erika		6	7	8
	X	geb. am 12.08.1964		9	0	1
	X	Heidestraße 17		2	3	4
	X	51147 Köln		5	6	7
	X	10/14		8	9	0
	X			1	2	3
	X			4	5	6
	X			7	8	9
	X			0	1	2
	X			3	4	5
	X			6	7	8
	X			9	0	1
	X			2	3	4
	X			5	6	7
	X			8	9	0
	X			1	2	3
	X			4	5	6
	X			7	8	9
	X			0	1	2
	X			3	4	5
	X			6	7	8
	X			9	0	1
	X			2	3	4
	X			5	6	7
	X			8	9	0
	X			1	2	3
	X			4	5	6
	X			7	8	9
	X			0	1	2
	X			3	4	5
	X			6	7	8
	X			9	0	1
	X			2	3	4
	X			5	6	7
	X			8	9	

Abbildung 1: Kassenrezept Muster, <https://www.meine-hautapotheke.de/hautchecker.html>, letzter Zugriff am 07.01.2019

Das rosafarbene verschreibungspflichtige Rezept ist das sogenannte Kassenrezept. Damit der Patient das Rezept in der Apotheke einlösen kann, müssen bestimmte Daten angegeben sein.¹⁶

1. Grundsätzlich ist eine Verordnung gebührenpflichtig und der Arzt muss das Feld "Geb.-pfl." kennzeichnen. Das Feld "Gebühr frei" wird nur angekreuzt, wenn der Versicherte das 18. Lebensjahr noch nicht vollendet hat, bei Schwangerschaftsbeschwerden sowie

¹⁵ Vgl. statista, 2019, Anzahl der Mitglieder und Versicherten der gesetzlichen und privaten Krankenversicherung in den Jahren 2012 bis 2018 (in Millionen), <https://de.statista.com/statistik/daten/studie/155823/umfrage/gkv-pkv-mitglieder-und-versichertenzahl-im-vergleich/>, letzter Zugriff am 11.01.2019.

¹⁶ Vgl. Kassenärztliche Bundesvereinigung, 2019, Arzneimittel-Verordnung, <https://www.kbv.de/html/27760.php>, letzter Zugriff am 06.01.2019.

in Fällen, “in denen eine Befreiung von der Zuzahlungspflicht (z.B. Härtefallregelung) nachgewiesen wird”¹⁷

2. Die Angaben über den Patienten dürfen nicht fehlen. Hier müssen die zuständige Krankenkasse, der vollständige Name, Geburtsdatum und Versichertennummer des Patienten eingetragen werden.
3. Durch das Ankreuzen des Feldes “noctu” wird der Patient von der Zahlung der Notdienstgebühr befreit, wenn er z.B. das Medikament dringend benötigt und außerhalb der allgemeinen Öffnungszeiten abholt.
4. Ist die Verordnung Folge eines Haus-, Sport- oder Verkehrsunfalls wird das Feld “Unfall” angekreuzt. Handelt es sich allerdings um einen Arbeitsunfall, wird das Feld “Arbeitsunfall” stattdessen markiert und es müssen zusätzlich Unfalltag und Unfallbetrieb in den dafür vorgesehenen Feldern eingetragen werden.
5. Hier wird die neunstellige lebenslange Arztnummer (LANR) angegeben. Jeder Arzt erhält eine vom Tätigkeitsort unabhängige Nummer durch die zuständige Kassenärztliche Vereinigung (KV) und behält sie “für die gesamte Dauer der vertragsärztlichen Tätigkeit”¹⁸.
6. Im Verordnungsfeld werden die verschriebenen Arzneimittel aufgelistet. Hier sind Name, Stärke, Menge und Form des Medikamentes angegeben. Es dürfen maximal drei Arzneimittel auf einem Kassenrezept eingetragen werden.¹⁹
7. Darf die Apotheke das verordnete Mittel nicht durch ein preisgünstigeres, wirkstoffgleiches Arzneimittel austauschen, wird das Aut-Idem-Feld markiert.
8. *“Bei Verordnungen für Anspruchsberechtigte nach dem Bundesentschädigungsgesetz ist wie bei Anspruchsberechtigten nach dem Bundesversorgungsgesetz das Feld 6 zu kennzeichnen.”*²⁰
9. Wird ein Hilfsmittel (7), ein Impfstoff (8) oder Sprechstundenbedarf (9) verordnet, muss das entsprechende Feld mit der zugehörigen Nummer gekennzeichnet werden.
10. Im oberen rechten Teil des Rezeptes werden nach der Rezept-Einlösung in der Apotheke die Abrechnungsfelder (Apotheken-Nummer, Zuzahlung, Gesamtbrutto, Faktor, Taxe, Arzneimittel-/Hilfsmittel-Nummer) ausgefüllt.

¹⁷ Vgl. DeutschesApothekenPortal, 2019, Betriebsstättennummer (BSNR), <https://www.deutschesapothekenportal.de/rezept-retax/dap-lexikon/betriebsstaettennummer-bsnr/>, letzter Zugriff am 06.01.2019.

¹⁸ Vgl. DeutscheApothekenPortal, 2019, LANR (Lebenslange Arztnummer), <https://www.deutschesapothekenportal.de/rezept-retax/dap-lexikon/lanr-lebenslange-arztnummer/>, letzter Zugriff am 06.01.2019.

¹⁹ Vgl. Steinbeißer, Kathrin, 2017, Der Rezept-Code: Das verrät ein Arztrezept wirklich, https://www.focus.de/gesundheitspraxistipps/kasse-oder-privat-rosa-oder-blau-der-rezept-code-das-verraet-ein-arztrezept-wirklich_id_4603036.html, 06.01.2019.

²⁰ Vgl. DeutschesApothekenPortal, 2019, Betriebsstättennummer (BSNR), <https://www.deutschesapothekenportal.de/rezept-retax/dap-lexikon/betriebsstaettennummer-bsnr/>, letzter Zugriff am 06.01.2019.

11. Im Vertragsarztstempel muss Name und Fachbezeichnung des Arztes, Adresse und Nummer der Betriebsstätte angegeben werden. Erst durch die Unterschrift des Arztes wird das Kassenrezept gültig.
12. Am unteren Ende wird in der Codierleiste die Betriebsstätten-Nummer (BSNR) eingetragen, die angibt wo der Arzt zur Zeit der Rezept-Ausstellung arbeitet. Die BSNR ist eine neunstellige Nummer, die durch ihre Zusammensetzung *“den Ort der Leistungserbringung (Betriebsstätte) eindeutig identifiziert”*²¹.

2.4 Aktueller Ablauf der Rezept-Ausstellung

Nach der medizinischen Untersuchung und der Diagnose verschreibt der zuständige Arzt oftmals zur Behandlung der Symptome ein spezielles Medikament. Der Arzt gibt über eine Software auf dem Computer alle wichtigen Daten über das Medikament an und händigt dem Patienten das unterschriebene noch unausgefüllte Rezept aus. Der Patient überreicht das leere Rezept dem Assistenten am Empfang. Der Assistent wählt die vom Arzt eingetragenen Daten ebenfalls über den Computer aus und der Patient bekommt das Rezept in Papierform maschinell ausgefüllt. Anschließend kann der Patient das Rezept in einer Apotheke seiner Wahl einmalig einlösen und erhält das benötigte Medikament.

Der Apotheker füllt die noch leeren Abrechnungsfelder aus und bewahrt das Rezept sicher auf. In regelmäßigen Abständen werden die Rezepte der jeweiligen Apotheke eingesammelt und zum Rechenzentrum abtransportiert. Dort werden die Rezeptstapel mit Hochgeschwindigkeitsscannern maschinell eingelesen und für die Abrechnung mit den Krankenkassen, Apotheken, Herstellern usw. in digitale Daten umgewandelt. Abschließend werden die Rezepte ein weiteres Mal gescannt und für den Versand an die Kostenträger, in den meisten Fällen die Krankenkassen, sortiert.²²

Seit dem Jahr 2004 ist der Versandhandel sämtlicher in Deutschland erlaubter Arzneimittel über verifizierte Online-Apotheken erlaubt.²³ Bestellt der Patient das verschriebene Medikament über einen solchen Versandhandel, muss er das Originalrezept in Papierform an die Online-Apotheke per Post senden. Das pharmazeutische Personal prüft die Angaben und der Patient bekommt das Medikament nach Hause geliefert.²⁴

²¹ Vgl. DeutschesApothekenPortal, 2019, Betriebsstättennummer (BSNR), <https://www.deutschesapothekenportal.de/rezept-retax/dap-lexikon/betriebsstaettennummer-bsnr/>, letzter Zugriff am 06.01.2019.

²² Vgl. Ditzel, Peter, 27.08.2015, Wie ein Rezept zu Geld wird, <https://www.deutsche-apotheker-zeitung.de/daz-az/2015/daz-35-2015/wie-ein-rezept-zu-geld-wird>, letzter Zugriff am 07.01.2019.

²³ Vgl. Bundesvereinigung Deutscher Apothekerverbände e. V., 2019, Internet und Versandhandel, <https://www.abda.de/themen/recht/apotheken-wettbewerb/versandhandel/>, letzter Zugriff am 06.03.2019.

²⁴ Vgl. DocMorris N.V., 2018, Rezept einlösen, <https://www.docmorris.de/rezepteinloesen?navnode=redeemp-rescription>, letzter Zugriff am 06.03.2019.

Benötigt der Patient ein Medikament mit den gleichen Wirkstoffen und unveränderter Dosierung dauerhaft oder über einen längeren Zeitraum (z.B. aufgrund einer chronischen Erkrankung), dann wird ein *Folgerezept* verschrieben. Folgerezepte sind Rezepte, die in den meisten Fällen keine weitere Untersuchung durch den Arzt benötigen.²⁵

Auch wenn keine Untersuchung durch den Arzt notwendig ist, so muss der Patient das gewünschte Rezept beispielsweise telefonisch anfordern und es persönlich in der Praxis abholen. Bei Untersuchungen oder Rezept-Ausstellungen, welche in einem neuen Quartal stattfinden, muss die Versicherungskarte in der Arztpraxis erneut vorgelegt werden. Der Patient muss einige Zeit warten, bis das Rezept bearbeitet wurde und bekommt es dann ausgehändigt und kann es in der Apotheke einlösen.

Auch ohne Untersuchung muss der Patient den Weg und die Wartezeit auf sich nehmen, um das Stück Papier abzuholen. Diese aufwändige Vorgehensweise kann durch das Digitale Rezept umgangen werden.

²⁵ Vgl. vitabook GmbH, 2019, Folgerezept, <https://www.vitabook.de/gesundheitslexikon/folgerezept.php>, letzter Zugriff am 07.01.2019.

3 Anforderungsanalyse

In diesem Kapitel wird der aktuelle Stand der verwendeten Technologien im Gesundheitswesen festgehalten. Daraus ergeben sich Anforderungen an das Konzept und die benötigte Software.

3.1 Digitalisierung des Gesundheitswesens

Neue Technologien und deren Vernetzung halten Einzug in nahezu allen Lebensbereichen. Der technische Fortschritt macht auch vor dem Gesundheitswesen nicht Halt.

Der Begriff *E-Health* fasst alle Anwendungen zusammen, die mit Hilfe moderner Informations- und Kommunikationstechnologien (IKT) die Behandlung und Betreuung von Patienten ermöglichen und unterstützen. Informationen und Daten sollen in Zukunft elektronisch verarbeitet und ausgetauscht werden.²⁶

Um IKT-gestützte Anwendungen und Abläufe im Gesundheitswesen zu optimieren und die Qualität der medizinischen Versorgung zu verbessern, wurde im Januar 2016 das E-Health-Gesetz erlassen. In diesem Gesetz wird der Einsatz der Telematikinfrastuktur behandelt, damit unter Beachtung der Datensicherheit die sensiblen Gesundheitsdaten zur Verfügung gestellt werden können.²⁷

3.1.1 Telematikinfrastuktur

Die Telematikinfrastuktur (TI) ist ein Intranet und soll für einen sicheren Datenaustausch zwischen allen Akteuren im Gesundheitswesen sorgen. Um diese Art der modernen Kommunikation nutzen zu können, müssen unter anderem die Arztpraxen technisch aufrüsten.

Für die Anbindung an das neue System mussten die Praxen bis Ende 2018 mit Internetzugang, eventuell VPN-Verbindung, Konnektoren und neuen Kartenlesegeräten - beispielsweise einem E-Health-Kartenterminal - ausgestattet sein. Jede Praxis authentifiziert sich mit einem elektronischen Praxisausweis und jeder Arzt benötigt einen elektronischen Heilberufsausweis, der als qualifizierte elektronische Signatur (QES) dient. Fester Bestandteil der Telematikinfrastuktur ist die neue elektronische Gesundheitskarte (eGK), die jeder Patient innerhalb der

²⁶ Vgl. Bundesministerium für Gesundheit, 17.05.2018, E-Health, <https://www.bundesgesundheitsministerium.de/service/begriffe-von-a-z/e/e-health.html>, letzter Zugriff am 22.01.2018.

²⁷ Vgl. Bundesärztekammer, 2018, E-Health-Gesetz - neue Anwendungen für Ärzte und Versicherte kommen, <https://www.bundesaerztekammer.de/aerzte/telematiktelemedizin/earztausweis/e-health-gesetz/>, letzter Zugriff am 22.01.2019.

Umstrukturierung (2011-2015) erhalten hat.²⁸ Jede eGK hat einen individuellen Schlüssel gespeichert, mit dem personenbezogene Daten verschlüsselt abgerufen werden können. Der Konnektor dient als Schnittstelle zwischen dem Kartenterminal und dem zentralen Datenserver, um die Personen- und Versicherungsdaten über eine verschlüsselte VPN-Verbindung auszutauschen.²⁹

Mit der Einführung der Telematikinfrastruktur und dem E-Health-Gesetz werden die Prozessabläufe im Gesundheitswesen schrittweise digitalisiert.

3.1.2 E-Rezept

Durch den Fortschritt der Telemedizin plant jetzt auch das Bundesgesundheitsministerium, das ärztliche Rezept in Papierform durch neue Technologien zu ersetzen.

Angekündigte gesetzliche Neuregelungen sollen den Einsatz des elektronischen Rezeptes, dem sogenannten E-Rezept, spätestens bis Frühjahr 2020 ermöglichen. *“Erst das elektronische Rezept macht Telemedizin zu einem Erfolgsprojekt. [...] Die Telemedizin spart Ärzten und Patienten Zeit und Wege – vor allem auf dem Land und außerhalb der üblichen Praxisöffnungszeiten.”*³⁰

Für das technische Grundkonzept der Datenspeicherung und des Datenzugriffs sollen Ärzte, Apotheker und Kassen zusammenarbeiten und ihre Anforderungen abstimmen und kombinieren.

Die Bundesvereinigung Deutscher Apothekerverbände (ABDA) hat umgehend nach der Ankündigung einen Konzeptentwurf an das Bundesministerium für Gesundheit (BMG) eingereicht. Für die Realisierung des Konzeptentwurfs sollen die elektronischen Verordnungen in zwei Phasen eingeführt und getestet werden. In der ersten Phase soll der Arzt den Patienten einen elektronischen Datensatz oder eine Bilddatei des Rezeptes über eine Plattform zukommen lassen, als eine Art digitales Abbild. Dieses Abbild zeigt der Patient in der Apotheke wie gewohnt vor. Dort wird dann ein Datensatz zur Abrechnung erzeugt und an die Rechenzentren weitergeleitet.³¹

Im Gegensatz zu einem regionalen Einsatz des Modellprojektes in der ersten Phase, soll in der zweiten Phase das E-Rezept bis 2020 flächendeckend in Deutschland im Einsatz sein.

²⁸ Vgl. GKV-Spitzenverband, 2018, Elektronische Gesundheitskarte (eGK), https://www.gkv-spitzenverband.de/krankenversicherung/telematik_und_datenaustausch/egk/egk.jsp, letzter Zugriff am 22.01.2019.

²⁹ Vgl. Video, Kassenärztliche Bundesvereinigung, 2018, Anbindung an die Telematikinfrastruktur, <https://www.kzbv.de/anbindung-an-die-telematikinfrastruktur.1163.de.html>, letzter Zugriff am 22.01.2019.

³⁰ Vgl. Deutsche Apotheker Zeitung DAZ.online, 13.11.2018, Spahn: Fernverordnungsverbot fällt, E-Rezept bis 2020, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/11/13/spahn-fernverordnungsverbot-faellt-e-rezept-bis-2020/chapter:1>, letzter Zugriff am 23.01.2019.

³¹ Vgl. Deutsche Apotheker Zeitung DAZ.online, 10.08.2018, ABDA legt Plan für eigenes E-Rezept vor, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/08/10/abda-legt-plan-fuer-eigenes-e-rezept-vor>, letzter Zugriff am 23.01.2019.

Hier sollen, anders als in Phase eins, alle Akteure bereits an die Telematikinfrastuktur angebunden sein. Die Voraussetzungen hierzu wurden bereits in Kapitel 3.1.1 *Telematikinfrastuktur* beschrieben. Der Arzt speichert dann den Datensatz der ärztlichen Verordnung mit seiner elektronischen Signatur auf einem Server und leitet ihn zusätzlich an den Patienten weiter. Der Patient geht anschließend wieder wie gewohnt zur Apotheke und zeigt das Rezept vor. Der Apotheker ruft den gespeicherten Datensatz vom Server ab und schickt den zugehörigen Abrechnungsdatensatz an das zuständige Rechenzentrum.³²

Der ABDA ist es dabei besonders wichtig, dass die internen Prozesse auf der Telematikinfrastuktur basieren und mit ihr kompatibel sind. Der Patient soll jederzeit die Option der freien Apothekenwahl haben und über seine eigenen Daten verfügen können.

Neben dem Zusammenschluss der verschiedenen Akteure und den Testphasen, ist für die Umsetzung des E-Rezeptes von großer Wichtigkeit, dass die Politik entsprechende Gesetze erweitert oder erlässt, um das Projekt zu unterstützen und voran zu bringen.

3.1.3 Der Einsatz von Applikationen im Gesundheitswesen

Auch wenn für Außenstehende eine Anbindung an die Telematikinfrastuktur bisher nicht vorgesehen ist, so wirkt sich die Umstrukturierung doch innovativ auf neue Technologien im Gesundheitswesen aus.

Neben Fitnesstrackern und digitalen Ernährungstagebüchern, gibt es noch weitere Applikationen, die den digitalen Weg in ein gesundes Leben erleichtern sollen und für einen effizienten Informationsaustausch zwischen Patient und Arzt in der Praxis angewendet werden:

Das Programm "Doctolib" unterstützt die Patienten, aber auch die Arztpraxen, bei einer Terminvereinbarung. Der Patient kann sich, von zuhause oder unterwegs, auch außerhalb der Sprechstunden über eine Arztpraxis informieren und online Termine buchen. "Doctolib" wirbt mit einer automatischen Nachrückliste, die beiden Parteien bei der Terminvergabe Zeit spart und für einen optimalen Patientenfluss sorgt.³³

Weitere Apps unterstützen nach der Terminvereinbarung einen reibungslosen Einstieg in die Behandlung. Mit Apps wie "LifeTime" und "Vivy" kann der Patient seine bisherigen Befunde sammeln und mit seinen verschiedenen Fachärzten teilen.³⁴ Somit können auch bei einem

³² Vgl. Deutsche Apotheker Zeitung DAZ.online, 10.08.2018, ABDA legt Plan für eigenes E-Rezept vor, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/08/10/abda-legt-plan-fuer-eigenes-e-rezept-vor/chapter:2>, letzter Zugriff am 23.01.2019.

³³ Vgl. Doctolib, 2019, <https://info.doctolib.de/?origin=homepage-website-desktop>, letzter Zugriff am 22.01.2019.

³⁴ Vgl. Tagesschau, Wenn "Vivy" ans Impfen erinnert, 17.09.2018, <https://www.tagesschau.de/inland/gesundheits-app-101.html>, letzter Zugriff am 22.01.2019.

Arztwechsel die bisherigen Symptome und Diagnosen zügig erfasst, weitergeleitet und behandelt werden. Durch eine Lockerung des Fernbehandlungsverbots Anfang Mai 2018 wurden Software-Anwendungen für einen virtuellen Arztbesuch umgesetzt und getestet, die in einigen Fällen eine ärztliche Behandlung über eine sogenannte "Video Sprechstunde" ermöglichen.³⁵

Auch nach der Diagnose wird die Beschaffung und Einnahme des Medikamentes durch Software erleichtert. Apps wie z.B. "DocMorris" sind Online-Apotheken, über die der Patient die benötigten Medikamente bestellen kann und nach Hause geliefert bekommt.³⁶

Die oben genannte App "LifeTime" bietet auch nach der Untersuchung weitere nützliche Funktionen, wie z.B. eine Erinnerungsfunktion zur regelmäßigen Medikamenteneinnahme.

Grundsätzlich ist anzumerken, dass keine der Anwendungen an die Telematikinfrastruktur angebunden ist. So bestehen Sicherheitsbedenken, da einige Anwendungen, z.B. "Vivy", Benutzerangaben speichern und sammeln können und diese an amerikanische Analysefirmen weitergeben.³⁷

3.2 Anwendungsfälle

Unter Verwendung dieser Technologien werden auch im Gesundheitswesen Prozesse digitalisiert und können somit zu deren maßgeblichen Beschleunigung und Vereinfachung beitragen. Durch das Digitale Rezept soll nun auch der Prozess der Rezept-Ausstellung im Rahmen der vorliegenden Arbeit an die digitale Zukunft angepasst werden. Mit Hilfe der Rezept-Digitalisierung sollen Patienten und Arztpraxen entlastet werden. Natürlich können Arztbesuche nicht ganz entfallen, doch gerade bei der Ausstellung von Folgerezepten, die oftmals keine erneute Untersuchung benötigen, können die Beratungs- und Beschaffungszeiten effizient verringert werden.

Im Folgenden werden innerhalb der Anforderungsanalyse die unterschiedlichen Anwendungsfälle und ihre Benutzergruppen festgehalten. Die daraus resultierenden Anforderungen bilden das Fundament des Konzeptes und definieren die Aufgaben der Software-Entwicklerin zur Realisierung des Prototyps.

³⁵ Vgl. <https://www.presseportal.de/pm/103509/3480877>

³⁶ Vgl. DocMorris, <https://www.docmorris.de/>, letzter Zugriff am 22.01.2019.

³⁷ Vgl. Horchert, Judith, 19.09.2018, Neue Gesundheits-App Vivy, <http://www.spiegel.de/netzwelt/apps/vivy-sicherheitsexperte-kritisiert-von-krankenkassen-angebotene-gesundheits-app-a-1228749.html>, letzter Zugriff am 22.01.2019.

Für ein strukturiertes Konzept und dessen Umsetzung ist es notwendig, Benutzerrollen und Anforderungen im Vorfeld klar zu definieren. Diese sollten aus Sicht aller beteiligten Interessengruppen (Stakeholder) betrachtet und praxisgerecht ausformuliert werden, um eventuelle Fehler frühzeitig zu erkennen und beseitigen zu können. Von der Rezept-Anforderung bis zur -Einlösung sind mehrere Akteure, sogenannte Stakeholder, wie Patient, Arzt/Arztpraxis und Apotheke/Apotheker beteiligt.

Zur Veranschaulichung der zu berücksichtigenden Benutzerrollen und benötigten Funktionen, werden die wichtigsten Anforderungen in Form von User Stories aufgelistet.

Der Patient...

- kann sich die mobile Applikation auf einem Smartphone seiner Wahl installieren.
- kann ein Konto erstellen und sich mit seinen Login-Daten anmelden.
- kann alle eigenen Daten abspeichern und jederzeit abrufen.
- kann versendete Rezept-Anforderungen in Form einer Liste einsehen.
- kann neue Rezept-Anforderungen über die App erstellen und die Daten zur Datenbank hinzufügen und dort speichern.
- kann ausgestellte Rezepte in Form einer Liste einsehen.
- kann einzelne Rezepte anwählen und sich somit das Rezeptformular mit den Detaildaten anzeigen lassen.
- kann über eine Navigationsleiste zwischen verschiedenen Inhalten wählen.

Der Arzt...

- kann über einen Browser seiner Wahl die Web-Anwendung aufrufen.
- hat ein Konto, in das er sich mit seinen Benutzerangaben einloggen kann.
- kann über die Web-Anwendung alle an ihn gesendeten Rezept-Anforderungen einsehen.
- hat die Möglichkeit, jede Anforderung einzeln zu bearbeiten und das Rezept auszustellen.
- kann jederzeit die geänderten Daten speichern und wieder abrufen.

Der Apotheker...

- kann das Rezept in der Apotheke über die Kassen-Anwendung einlösen.
- kann das Rezeptformular einsehen.
- kann jederzeit die zugehörigen Daten des Rezeptes abrufen.
- kann über die Kassen-Anwendung Abrechnungsfelder erstellen und ausfüllen.
- kann den Datensatz der Abrechnung an das zuständige Rechenzentrum senden.

4 Konzeption

Der Ablauf von der Rezept-Anforderung, über -Ausstellung, bis hin zur -Einlösung und der Einsatz der zugehörigen Software-Anwendungen werden kombiniert und miteinander in Verbindung gesetzt. Anhand von schematischen Darstellungen (Wireframes) werden anschließend die Benutzeroberflächen einzelner Seitenvorlagen und ihre Funktionalität erläutert.

4.1 Ablauf des Digitalen Rezeptes

Der aktuelle Stand der Rezept-Ausschreibung und die damit verbundenen Nachteile wurden bereits im Kapitel 2.4 *Aktueller Ablauf der Rezept-Ausschreibung* beleuchtet. Im gesamten Ablauf der digitalisierten Rezept-Ausstellung, kommen entsprechend aller internen Prozesse zwischen den Interessengruppen, geeignete Software-Elemente zum Einsatz.

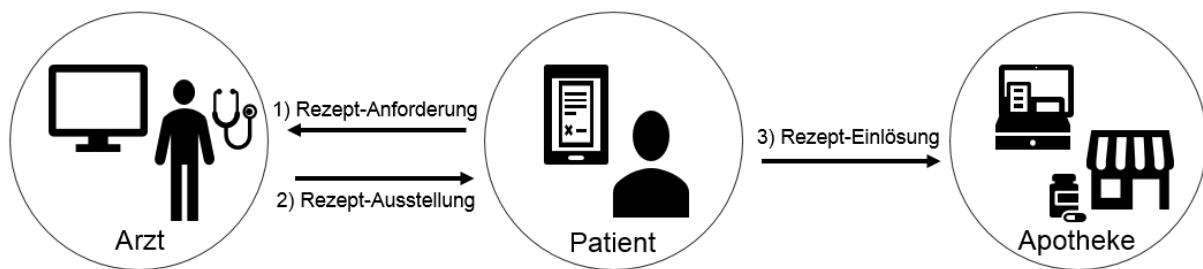


Abbildung 2: Ablauf des Digitalen Rezeptes, eigene Darstellung

Der erste Prozess, der für den Einsatz des Digitalen Rezeptes umstrukturiert werden muss, ist die Rezept-Anforderung. An dem Anforderungs-Prozess sind der Patient und der zuständige Arzt beteiligt. Über eine mobile App kann der Patient jederzeit und an jedem Ort das Rezept für das benötigte Medikament anfordern. In der App füllt der Patient ein Formular aus, welches er dann abschickt. Der Arzt kann die angeforderten Rezepte über eine Web-Anwendung einsehen und bearbeiten.

Im zweiten Prozess, der Rezept-Ausstellung, stehen wieder der behandelnde Arzt bzw. die Arztpraxis und der Patient in Verbindung. Der Arzt kann die Rezept-Anforderungen über die Web-Anwendung einsehen und entscheidet, ob er die Anfrage bestätigt. Durch die Bestätigung der Anforderung, schaltet er diese frei und die Daten werden an eine Datenbank übertragen und dort gespeichert. Nachdem der Arzt das Rezept bearbeitet und ausgestellt hat, kann der Patient das verschreibungspflichtige Rezept über seine App einsehen.

Am letzten Prozess im Ablauf, der Rezept-Einlösung, sind Patient und Apotheker bzw. Apotheke beteiligt. Der Patient kann das Digitale Rezept über seine App in der Apotheke vorzeigen und einlösen. Zudem soll es die Möglichkeit geben, dass der Patient das Digitale Rezept direkt an eine Online-Apotheke weiterleiten kann und das Medikament anschließend zu ihm nach Hause geliefert wird.

Für die Abrechnung muss die Apotheke in beiden Fällen den Datensatz des Rezeptes übermittelt bekommen. Der Apotheker kann mittels einer Kassen-Anwendung direkt vor Ort die Rezeptdaten einsehen und die Abrechnungsfelder ausfüllen. Der erstellte Datensatz wird anschließend an das zuständige Rechenzentrum übermittelt.

4.2 Technische Rahmenbedingungen

Zwischen den Software-Anwendungen werden kontinuierlich Daten ausgetauscht. Dabei kommunizieren diese jedoch nicht direkt miteinander.

Datenbank

Für einen Datenaustausch untereinander fungiert die Datenbank und ihre Anbindung als zentrale Schnittstelle zwischen den Anwendungen. Damit alle Software-Komponenten Zugriff auf die Daten erhalten, werden diese zentral in einer Datenbank abgelegt und gespeichert. Für einen strukturierten Aufbau der Daten, soll ein relationales Datenbankmodell zum Einsatz kommen (*siehe Kapitel 5.1.3 Datenbank*).

Sicherer Datenaustausch

Sensible Daten, wie Passwort und Patientendaten werden innerhalb der Prozesse zwischen den Software-Komponenten ausgetauscht und in einer Datenbank gespeichert. Der regelmäßige Datenaustausch zwischen den verschiedenen Systemkomponenten kann eine Schwachstelle des Systems sein. Damit Dritte nicht auf die Nachricht zugreifen oder sie manipulieren können, muss ein sicherer Datentransfer gewährleistet sein.

Zu diesem Zweck sollten die Daten für alle weiteren Übertragungsstationen zwischen Absender und Empfänger verschlüsselt sein. Die Anfragen aus der App, Web-Anwendung und Kassen-Anwendung an den Server werden über ein abhörsicheres Protokoll, eine sogenannte Hyper Text Transfer Protocol Secure (HTTPS)-Verbindung, übertragen. Die HTTPS-Verbindung basiert auf Secure Socket Layer (SSL)-Verschlüsselung und -Zertifikaten, welche die Daten mit einem nur für sie erkennbaren Schlüssel codieren und die Domain der Serveranfragen authentifizieren.³⁸

App

Für die Verwendung des Digitalen Rezeptes, sind folgende Bedingungen zu erfüllen. Zunächst benötigt der Patient ein internet- und NFC-fähiges Smartphone. Die mobile App kann er dann über den jeweiligen App-Store herunterladen und auf dem Smartphone installieren. Der Nutzer hat die Möglichkeit, zwischen einer App-Version für das Android- oder iOS-Betriebssystem zu wählen. Somit ist die App mit Android- und Apple-Geräten kompatibel. Über den Internetzugang kann die App auf die Datenbank zugreifen und Daten übertragen. Die Verwendung der NFC-Funktion wird zu einem späteren Zeitpunkt innerhalb dieser Arbeit geklärt.

³⁸ Vgl. Rödiger, Jana, 07.12.2017, HTTPS UND SSL VERSCHLÜSSELUNG - WAS IST DAS UND WARUM IST DAS FÜR WEBSEITEN UNVERZICHTBAR?, <https://www.media-company.eu/blog/allgemein/https-und-ssl-verschlueselung/>, letzter Zugriff am 22.02.2019.

Web-Anwendung

Damit Arztpraxen das Digitale Rezept ausstellen können, müssen diese mit spezieller Hardware ausgestattet sein. Die Arztpraxis bzw. der zuständige Arzt benötigt einen Zugang zu einem Computer, der eine sichere Verbindung zum Internet herstellen kann.

Die Web-Anwendung läuft auf einem öffentlichen Server und wird von den meist verbreiteten Browsern (Google Chrome, Firefox und Internet Explorer) unterstützt. Über den Internetanschluss werden Datenbankabfragen von der Web-Anwendung gesendet.

Digitale Signatur

Durch die Unterschrift des Arztes wird das Rezept gültig. Da der Arzt das Digitale Rezept nicht in herkömmlicher Weise unterschreiben kann, muss die Echtheit des Digitalen Rezeptes durch eine digitale Signatur sichergestellt werden.

Unter Verwendung asymmetrischer Verschlüsselungsverfahren, wird der Datensatz des auszustellenden Rezeptes in einen Hash-Wert (Prüfsumme) umgewandelt, mit einem privaten Schlüssel des Arztes codiert und über eine sichere Verbindung weitergeleitet. Kann der Patient oder Apotheker diesen Datensatz mit dem öffentlichen Schlüssel des Arztes decodieren, ist sichergestellt, dass die Nachricht nicht manipuliert wurde und die Identität des Absenders nachgewiesen ist.³⁹

Kassen-Anwendung

In den meisten Apotheken werden die Zahlungsvorgänge bereits mit einem auf den Dienst zugeschnittenen Point-of-Sale-System (POS) geregelt. Das POS ist ein Kassensystem, in welchem die Eingaben der Scannerkassen mit Hilfe einer Software verarbeitet werden.⁴⁰ Auf den Scannerkassen des verbreiteten POS wird meistens "Windows" als Betriebssystem genutzt. Durch die häufige Verwendung von Windows ist es möglich, weitere Programme an das Kassensystem anzupassen.

Um aufwändige technische Umrüstungen zu vermeiden, sollten alle Geräte mit dem Windows-Betriebssystem kompatibel sein.⁴¹ Für die Verbindung zur Datenbank und die Abfrage von Kundendaten, muss der Apotheker über das Kassensystem Zugang zum Internet haben.

Für die Realisierung der Einlösearten können die Kassen zusätzlich über Schnittstellen mit Barcode-Scannern und NFC-fähigen Karten-Terminals erweitert werden.⁴²

³⁹ Vgl. Rouse, Margaret, 2002-2019, Digitale Signatur (elektronische Signatur), <https://www.searchsecurity.de/definition/Digitale-Signatur-elektronische-Signatur>, letzter Zugriff am 22.02.2019.

⁴⁰ Vgl. Deutsche Apotheker Zeitung 2019, 12.01.2003, Apotheken-EDV - besser als ihr Ruf (EDV-Umfrage, Teil 1), <https://www.deutsche-apotheker-zeitung.de/daz-az/2003/daz-3-2003/uid-9024>, letzter Zugriff am 20.02.2019.

⁴¹ Vgl. EHI Retail Institute, 2018, Heute und zukünftig im Handel auf den Kassen eingesetzte Betriebssysteme im Jahr 2018 (Basis: Anzahl Installationen der befragten Unternehmen), <https://www.handelsdaten.de/deutschsprachiger-einzelhandel/im-handel-auf-den-kassen-eingesetzte-betriebssysteme-2018>, letzter Zugriff am 20.02.2019.

⁴² Vgl. Deutsche Apotheker Zeitung 2019, 12.01.2003, Apotheken-EDV - besser als ihr Ruf (EDV-Umfrage, Teil 1), <https://www.deutsche-apotheker-zeitung.de/daz-az/2003/daz-3-2003/uid-9024>, letzter Zugriff am 20.02.2019.

NFC

Near Field Communication (NFC) basiert auf der Radio-Frequency-Identification (RFID)-Technik und ermöglicht das drahtlose Übertragen von Daten per Funk. Für die Kommunikation zwischen zwei Elementen und die einwandfreie Datenübertragung, dürfen die Geräte nur einen Abstand von wenigen Zentimetern haben, was es Dritten (Angreifern) schwer macht, Daten unbemerkt abzufangen.

Das NFC-Schreib- und -Lesegerät baut ein elektromagnetisches Feld auf und versorgt den NFC-Transponder beim Eintreten in das elektromagnetische Feld mit Strom und die auf dem Transponder hinterlegten Daten können ausgetauscht werden.⁴³

Im Einzelhandel wird NFC bereits oft verwendet und mit einer NFC-fähigen Kreditkarte oder dem Smartphone (Transponder) kann der Kunde über das Karten-Terminal (Lesegerät), bargeldlos bezahlen.⁴⁴

QR-Code

Quick Response (QR)-Codes sind im Gegensatz zum herkömmlichen Strichcode zweidimensional und können somit mehr Informationen einbetten. Diese quadratische Matrix, bestehend aus weißen und schwarzen Strichen und Punkten, kann durch einen Scanner eingelesen werden. Der Scanner orientiert sich an drei Ecken des Quadrates, die durch Positionsmarkierungen gekennzeichnet sind, und kann die digital codierten Informationen einlesen, decodieren und weiterleiten. Der QR-Code enthält eine Fehlerkorrektur, durch die Informationen lesbar sind, auch wenn das Muster bis zu 30 Prozent beschädigt ist. Die gelesenen Daten können zu einer Website weitergeleitet werden oder durch andere kompatible Software verarbeitet werden.

Ein ähnlicher Code-Aufbau wird bereits im Personennahverkehr zur Fahrkartenkontrolle von Online-Tickets eingesetzt.⁴⁵

⁴³ Vgl. Tec GmbH & Co. KG, 2019, RFID-Technologie, <https://www.smart-tec.com/de/auto-id-welt/rfid-technologie>, letzter Zugriff am 22.02.2019.

⁴⁴ Vgl. Bauer, Curt, 18.05.2018, NFC: Einfach erklärt, was hinter Near Field Communication steckt, https://praxistipps.chip.de/nfc-einfach-erklart-was-hinter-near-field-communication-steckt_12294, letzter Zugriff am 22.02.2019.

⁴⁵ Vgl. Lingner, Margrit, 29.08.2014, QR-Code selbst erstellen - So geht's, <https://www.pc-magazin.de/ratgeber/qr-code-erstellen-selber-machen-how-to-2525646.html>, letzter Zugriff am 22.02.2019.

4.3 Aufbau und Bedienung der Anwendungen

Um eine einfache und effiziente Bedienbarkeit der jeweiligen Software zu ermöglichen, muss die grafische Benutzeroberfläche (GUI, von graphical user interface) für die Anwender übersichtlich und verständlich aufgebaut sein. Dafür sollen die funktionalen Aufgaben der App über Menüs angeordnet werden. Für eine optimale Verwendung und gute Performanz der jeweiligen Software muss diese an das jeweilige Betriebssystem angepasst werden. Mögliche Fehler im Prozess-Ablauf müssen berücksichtigt und behandelt werden. In diesem Fall muss der Anwender eine Rückmeldung über die Benutzeroberfläche erhalten.

4.3.1 App

Nach dem ersten Starten der App muss der Patient mit seinen persönlichen Angaben ein Konto erstellen und sich somit registrieren. Neben Angaben zur Person, wie Name, Geburtsdatum und Adresse, sind hier zur Identifikation ein Lichtbild seiner eingescannten Versicherungskarte zu hinterlegen. Nach der Erstellung des Kontos ist die Anmeldung mit Angabe des Nutzernamen und Passwortes schließlich über eine Login-Maske möglich.

Beim Öffnen der App wird der Nutzer automatisch zur Startseite weitergeleitet. Die Startseite zeigt drei Menüpunkte, welche der Nutzer über eine Navigationsleiste am unteren Ende des Displays einzeln auswählen kann.

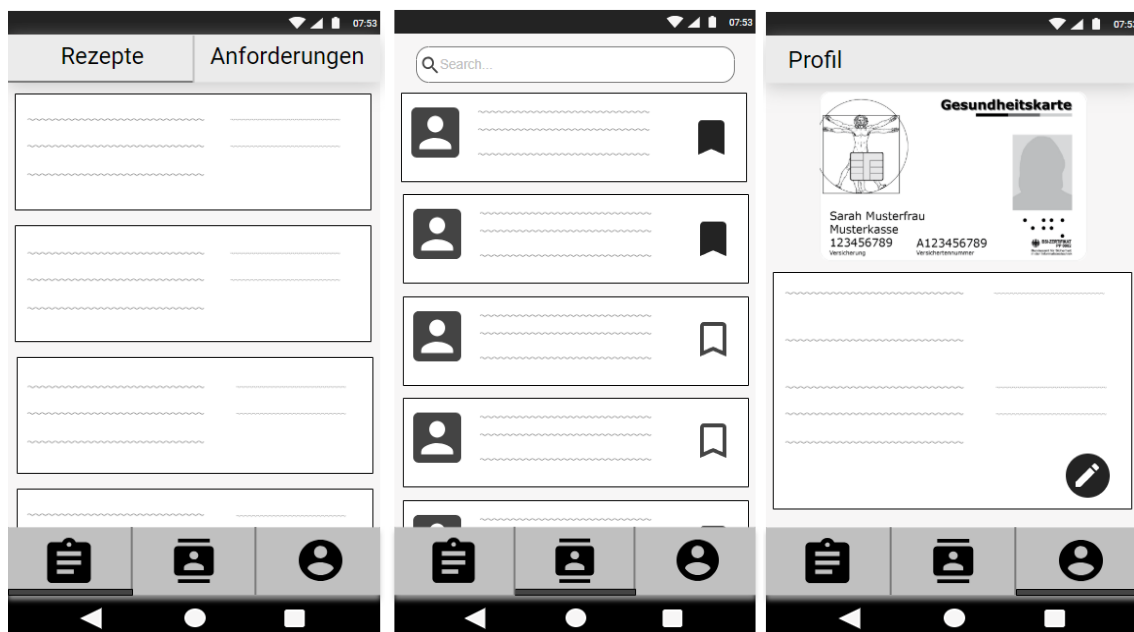


Abbildung 3: Visualisierung der App, eigene Darstellung

Die Startseite ist in zwei Tab-Ansichten aufgeteilt. Die erste Tab-Ansicht "Rezepte" wird standardmäßig nach dem Öffnen der App angezeigt. Dort werden später die ausgestellten Rezepte aufgelistet. Beim ersten Aufruf ist diese Seite noch leer und weist den Nutzer darauf hin, eine Rezept-Anforderung über den Tab "Anforderungen" zu erstellen. In dieser Tab-Ansicht werden

später auch die versendeten Anforderungen in einer Liste angeordnet. Über einen Button oberhalb der Navigationsleiste kann der Nutzer eine neue Rezept-Anforderung erstellen. Wie genau dieser Prozess innerhalb der Applikation abläuft, wird in Kapitel 4.4.1 *Rezept-Anforderung* beschrieben.

Die mittlere Darstellung der Benutzeroberfläche zeigt eine Übersicht der Ärzte bzw. Arztpraxen, die den Dienst des Digitalen Rezeptes anbieten. Der Nutzer kann die Auflistung der Ärzte manuell durchsuchen oder die Suchfunktion am oberen Rand des Displays verwenden. Dafür gibt er den Namen des Arztes in eine Suchleiste ein und der gesuchte Arzt erscheint ganz oben in der Liste. Jedes Element der Liste ist gleich aufgebaut und gibt eine kurze Übersicht mit den wichtigsten Angaben über den Arzt, z.B. Name, Adresse, Telefonnummer und Öffnungszeiten der Arztpraxis. Einzelne Ärzte können als Favoriten markiert und gespeichert werden. Für zukünftige und schnellere Suchanfragen werden alle markierten Ärzte anschließend oben in der Liste angezeigt.

Der rechte Menüpunkt zeigt die Profilseite des Nutzers. Dort werden alle personenbezogenen Angaben, die bei der Registrierung angegeben wurden, dargestellt. Dort kann der Nutzer seine Profildaten jederzeit ändern oder über einen "Logout"-Button die App verlassen und die Übermittlung von Daten abbrechen.

4.3.2 Web-Anwendung

Neben dem Arzt sind auch die medizinischen Fachangestellten einer Praxis an der Ausstellung eines Rezeptes beteiligt und benötigen daher als weiterer Nutzer ebenfalls den Zugang zur Web-Anwendung.

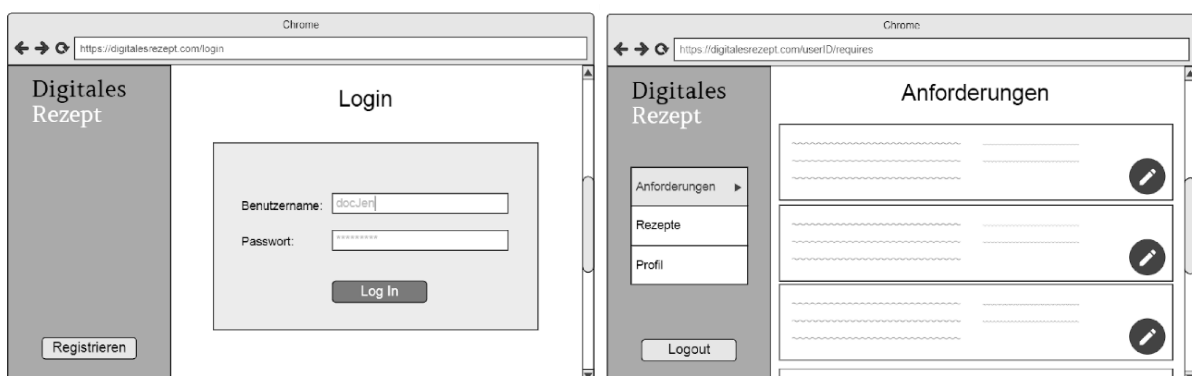


Abbildung 4: Visualisierung der Web-Anwendung, eigene Darstellung

Nach Eingabe der URL wird der Nutzer zunächst zur Login-Seite weitergeleitet. Beim ersten Starten der Web-Anwendung muss der Nutzer zunächst ein Konto erstellen und sich registrieren. Dazu wird er von der Login-Seite zu einer Registrierungsmaske weitergeleitet. In der

Maske muss er Angaben über seine Person, Betriebsstätte, zukünftiger Nutzernamen und Passwort machen. Mit Angabe seiner LANR, lebenslangen Arztnummer, kann der Arzt identifiziert werden. Nach der Registrierung gelangt der Arzt wieder zum Login-Formular, wo er sich mit Benutzernamen und Passwort anmeldet.

Ist der Nutzer angemeldet, wird er direkt zur Auflistung der eingegangenen Rezept-Anforderungen weitergeleitet. Dort erhält er eine erste Übersicht der Patientendaten, die er prüfen und mit bisherigen Behandlungs-Dokumentationen abgleichen kann. Dies könnte ebenfalls mittels digitaler Verfahren parallel zur Web-Anwendung erfolgen. Für die Bearbeitung der Daten kann der Nutzer nun ein Rezeptformular erstellen. Auf die Erstellung eines Digitalen Rezeptes wird in Kapitel 4.4.2 *Rezept-Ausstellung* genauer eingegangen. Sind alle Felder ausgefüllt, werden diese durch einen Button bestätigt und das Digitale Rezept wird ausgestellt.

Neben der Bearbeitung der Rezept-Anforderungen kann der Nutzer über eine seitliche Menüleiste zu den Seiten "Rezepte" und "Profil" navigieren. Unter dem Menüpunkt "Rezepte", kann der Arzt die bereits ausgestellten Rezepte mit einer Detailübersicht und Statusmeldung "gültig/abgelaufen/eingelöst" einsehen.

Auf der Profilseite werden die Kontodaten des Arztes angezeigt, die er dort jederzeit bearbeiten kann. Zum Beispiel können Informationen über die Öffnungszeiten und Sprechstunden angepasst werden. Über einen Logout-Button kann der Nutzer jederzeit die Web-Anwendung verlassen und die Übermittlung von Daten abbrechen.

4.3.3 Kassen-Anwendung

Auch für die Nutzung der Kassen-Anwendung muss sich die Apotheke bzw. der Apotheker vor dem ersten Gebrauch registrieren. Beim Erstellen des Kontos muss die Apotheke Angaben über ihren Standort machen und kann sich anschließend über ein Login authentifizieren.



Abbildung 5: Visualisierung der Kassen-Anwendung, eigene Darstellung

Nach der Registrierung und Anmeldung gelangt der Apotheker auf eine Startseite. Ähnlich wie in der Web-Anwendung für den Arzt, kann der Apotheker auf der Startseite über ein seitliches Menü zwischen den Funktionen der Kassen-Anwendung wählen. Bevor der Apotheker das vorgezeigte Rezept einlöst und abrechnet, muss er die Art der Datenübertragung einstellen. Neben der Hauptfunktion zur Einlösung und Abrechnung des Digitalen Rezeptes (*siehe Kapitel 4.4.3 Rezept-Einlösung*), kann sich der Apotheker über das Menü alle ihm zugeordneten Rezepte zur Kontrolle auflisten lassen, benutzerspezifische Einstellungen vornehmen und seine Profilseite einsehen. Auf der Profilseite kann er seine Profilangaben anpassen und speichern. Über den Logout-Button kann der Nutzer jederzeit die Kassen-Anwendung verlassen und die Übermittlung von Daten abbrechen.

4.4 Prozesse

In diesem Kapitel wird der Einsatz der Software-Anwendungen den Prozessen im Ablauf zugeordnet. Anhand detaillierter Wireframes wird die Vorgehensweise der Prozesse erklärt und die damit verbundenen Funktionen der Anwendungen.

4.4.1 Rezept-Anforderung

Um ein Rezept zu erhalten, muss der Patient eine Anfrage an den Arzt stellen. Dazu nutzt er die Startseite der App. Im "Anforderungen"-Tab kann er über einen Button eine neue Anforderung erstellen und wird zu einem Formular weitergeleitet.



Abbildung 6: Visualisierung der Rezept-Anforderung, eigene Darstellung

Jede Anforderung benötigt Angaben über die Beschwerden und das gewünschte Medikament. Bevor der Patient die Anforderung versenden kann, muss er den Namen des zuständigen Arztes in ein Textfeld eintragen. Während der Eingabe werden ihm passende Treffer aus der Ärztekartei eingeblendet. Hat der Patient bereits einige seiner Ärzte aus der Kartei markiert, werden ihm diese zur Auswahl in einem Drop-Down Menü angezeigt. Vor dem Versenden des Datensatzes werden personenbezogene Daten, die der Arzt benötigt, automatisch aus dem Profil des Nutzers ergänzt. Sind alle Textfelder ausgefüllt, kann er seine Angaben über einen Button am unteren Ende der Seite bestätigen und abschicken. Die Eingabemaske schließt sich und der Nutzer wird zum "Anforderungen"-Tab zurückgeleitet, wo die aktuell gesendete Anforderung zur Liste hinzugefügt wird.

Jede aufgelistete Anforderung enthält eine kleine Statusanzeige, die den Patienten informiert, ob die Anforderung gelesen, abgelehnt oder bearbeitet wurde.

Abgelehnte und bearbeitete Anforderungen werden in der Liste ausgegraut angezeigt und können bei wiederholtem Bedarf, z.B. aufgrund von chronischen Krankheiten, ohne weitere Eingaben erneuert werden.

Hat der Patient seine Rezept-Anforderung versendet, bekommt der zuständige Arzt diese anschließend über die Web-Anwendung unter dem Menüpunkt "Anforderungen" angezeigt. Die Anforderungen sind einzelne Elemente einer Liste. In jedem Element sind die Angaben des Patienten für eine erste Übersicht angeordnet.

4.4.2 Rezept-Ausstellung

In der Liste, die in der Web-Anwendung sichtbar wird, werden zu jeder Anforderung die zugehörigen Angaben, wie Name, Geburtsdatum, Versicherung, Krankenversicherungsnummer des Patienten, Beschwerden und gewünschtes Medikament, dargestellt. Der Arzt kann die einzelnen Rezept-Anforderungen einsehen und nach Prüfung der Angaben ein Rezeptformular ausfüllen. Sollte für die Ausstellung eines Rezeptes doch eine erneute Untersuchung nötig sein, kann der Arzt die Rezept-Anforderung ablehnen, den Bearbeitungsprozess überspringen und dem Patienten über einen Button-Klick eine Einladung zu einem neuen Untersuchungstermin senden.

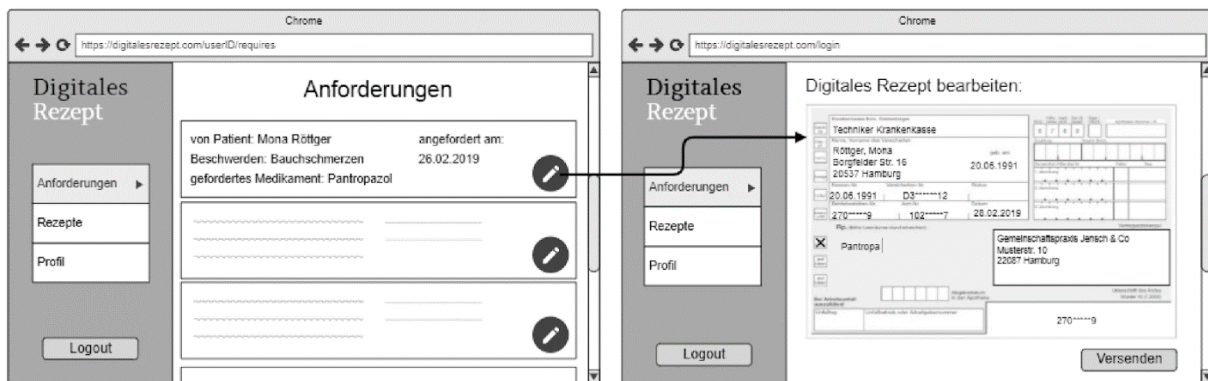


Abbildung 7: Visualisierung der Rezept-Ausstellung, eigene Darstellung

Zur Bestätigung der Rezept-Anforderung muss der Arzt ein Rezept erstellen. Über einen Button gelangt der Arzt in eine Bearbeitungsmaske, wo er erste Einstellungen vornehmen kann. In den Voreinstellungen kann der Arzt die Art des Rezeptes (siehe Kapitel 2.2 Rezeptarten) auswählen, welches anschließend in der Bearbeitungsmaske im gewohnten Format grafisch dargestellt wird. Bereits vorhandene Angaben, wie z.B. Name und Adresse des Patienten, Name des behandelnden Arztes und die Adresse seiner Praxis, werden aus der zugehörigen Anforderung und dem eigenen Profil übernommen und als Standardwerte voreingestellt. Bevor das Digitale Rezept ausgeschrieben werden kann, muss der Nutzer die restlichen Felder, z.B. Medikamentenform und -menge, manuell in die Maske eintragen. Hier kann der Arzt, anstelle

des angeforderten Medikaments, nach eigenem Ermessen natürlich auch ein alternatives Arzneimittel verschreiben.

Die bisherigen Rezepte in Papierform wurden erst durch Stempel und Unterschrift des Arztes gültig. Für das Digitale Rezept werden diese Angaben durch eine digitale Signatur ersetzt, die für jeden Arzt generiert wird und das Rezept verifiziert. Über einen Bestätigungs-Button wird der Datensatz für das Digitale Rezept verarbeitet. Für ein gültiges und somit einlösbares Rezept wird der Datensatz als QR-Code generiert und zugleich für die NFC-Übertragung vorbereitet. Anschließend wird das Digitale Rezept ausgestellt und die Daten werden an den Patienten weitergeleitet.

Nach dem Generieren des Rezeptes wird der Patient durch die App auf seinem Smartphone benachrichtigt und gelangt über die Startseite der App auf den "Rezepte"-Tab. Dort wird das Rezept in einer Liste sichtbar und kann zum Einlösen geöffnet werden.

4.4.3 Rezept-Einlösung

Auf dem Smartphone des Patienten wird jedes bestätigte Rezept als ein Element der Liste angezeigt und enthält eine kurze Info über die Art des Rezeptes und das Medikament. Für einen besseren Überblick wird in der unteren Ecke jedes Elementes eine kleine Statusanzeige eingeblendet, die den Patienten informiert, wie lange das Rezept gültig ist. Wurde ein Rezept in einer Apotheke bereits eingelöst, zeigt die Statusmeldung "eingelöst" an und das Rezept-Element wird inaktiv.

Alle noch aktiven Rezept-Elemente kann der Patient in einer Apotheke einlösen. Dafür wählt er das Rezept-Element in der Liste aus und alle enthaltenen Angaben des Rezeptes werden auf einer Detailseite, dem eigentlichen Digitalen Rezept, in einer mobilen Ansicht angeordnet.



Abbildung 8: Visualisierung der Rezept-Einlösung, eigene Darstellung

Ganz oben befindet sich der generierte QR-Code. Darunter befinden sich die aktuellen Rezeptangaben, wie vollständiger Name, Geburtsdatum, Adresse, zuständige Krankenkasse und Versicherungsnummer des Patienten, Bezeichnung, Form und Menge des Medikamentes, sowie Name, Adresse und elektronische Signatur des Arztes.

In der Apotheke zeigt der Patient das Digitale Rezept in der Detailansicht auf seinem Smartphone vor. Nachdem der Apotheker alle Angaben überprüft hat, kann der Einlöse-Vorgang beginnen.

Wie in der ersten Phase zur Einführung des E-Rezeptes, kann der Apotheker auch das Digitale Rezept über die Einsicht der Detailangaben prüfen und einlösen.

Um die Rezept-Einlösung abzusichern und unverfälschte Daten zu erhalten, sollen dem Apotheker mehrere Einlösearten zur Verfügung stehen. Über den Menüpunkt "Einlösung" in der Kassen-Anwendung wählt der Apotheker die Einlöseart, abhängig vom Kundenwunsch und der eigenen technischen Ausstattung und kann dann entweder den QR-Code des Rezeptes scannen oder den zugehörigen Datensatz über eine NFC-Übertragung an die Kassen-Anwendung senden.

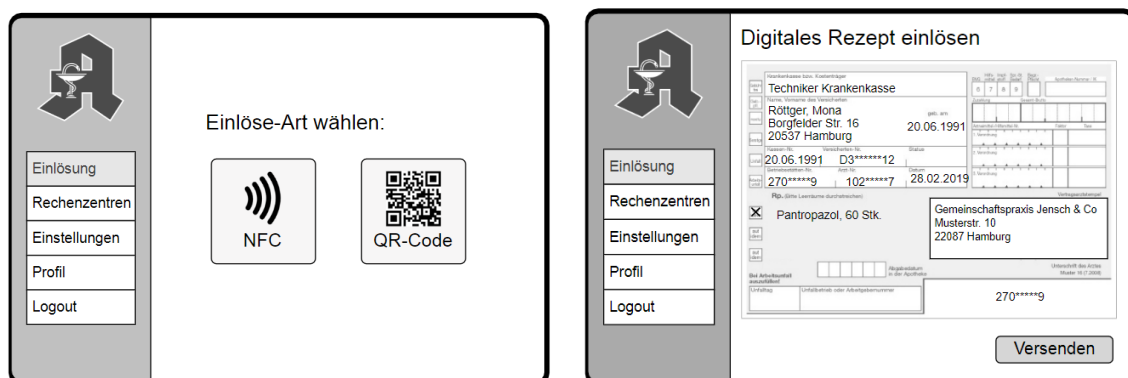


Abbildung 9: Visualisierung der Abrechnung eines Digitalen Rezeptes, eigene Darstellung

Nach der Übertragung des Datensatzes öffnet sich eine neue Maske. Im System der Apotheke wird das Digitale Rezept mit allen Angaben zur erneuten Kontrolle im freien Bereich neben dem Menü dargestellt. Die grafische Darstellung des Rezeptformulars ist teilweise mit den übertragenen Daten über Patient, Arzt und das verschriebene Medikament ausgefüllt. Die Abrechnungsfelder in der oberen rechten Ecke sind noch frei. Hier muss der Apotheker über die Kasse die Felder ausfüllen. Nach der Eingabe von Informationen über die jeweilige Apotheke und das verschriebene Medikament kann der Nutzer die Eingaben über einen Button bestätigen und das komplett ausgefüllte Rezept an das zuständige Rechenzentrum weiterleiten. Abschließend erhält der Patient sein Medikament.

Möchte der Patient das Digitale Rezept in einer Online-Apotheke einlösen, kann er den Datensatz über eine "Teilen"-Funktion übertragen. Dafür muss eine Anbindung an die jeweilige Online-Apotheke gewährleistet sein, um den Datensatz im passenden Format zu übertragen. Zusätzlich kann der Privatpatient für die Rückerstattung der entstandenen Kosten den Datensatz des eingelösten Rezeptes optional an seine Krankenkasse schicken.

5 Realisierung des Prototyps

In diesem Kapitel wird erläutert, mit Hilfe welcher Software-Anwendungen der Prototyp eines digitalen, ärztlichen Kassenrezeptes realisierbar ist. Von der Anfrage bis hin zur Bestätigung und Einlösung eines Digitalen Rezeptes ist ein Zusammenspiel von mehreren technischen Komponenten notwendig, insbesondere App, Web-Anwendung und Datenbank mit Anbindungsschnittstelle.

5.1 Technisches Konzept

Für einen strukturierten Aufbau und eine problemlose Realisierung des Prototyps wird in diesem Abschnitt zunächst das technische Konzept erarbeitet und anhand von Diagrammen erklärt. Zur besseren Verständlichkeit können die folgenden Diagramme von dem realisierten Konzept teilweise abweichen.

5.1.1 Projektübersicht

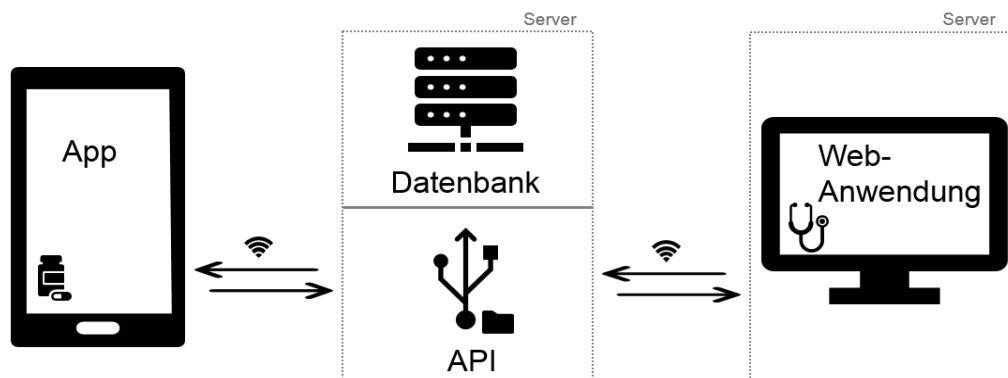


Abbildung 10: Zusammenspiel der technischen Komponenten, eigene Darstellung

Der Patient benutzt die App auf seinem Smartphone, um eine Rezept-Anforderung für ein benötigtes Medikament an die zuständige Arztpraxis zu senden. Der Arzt kann die Anforderung über die Web-Anwendung einsehen und bearbeiten. Wird die Anforderung akzeptiert und bestätigt, erhält der Patient sein Rezept über die App und kann das Medikament in der Apotheke abholen.

Für die Kommunikation und den Datenaustausch zwischen Arzt und Patient laufen im Hintergrund jeweils die Datenbank und das Application Programming Interface (API) auf einem, im Moment lokalen, Apache-Server. Die App und die Website können nur über das API auf die Daten der Datenbank zugreifen. Von der API werden dann über Structured Query Language (SQL)-Befehle Daten eingetragen, gelesen oder geändert. Die Anfragen der App und Website und Rückgabewerte der API, werden in Form von JSON-Objekten übermittelt.

Das JSON-Format wird häufig zum Datenaustausch zwischen Client und Server eingesetzt. Obwohl JavaScript Object Notation (JSON) auf JavaScript basiert, ist es *“unabhängig von der Programmiersprache einsetzbar”*⁴⁶. Für eine gute Lesbarkeit und die Verarbeitung durch einen Parser ist ein gut strukturierter Aufbau der Informationen wichtig:

```
{ "name": "Mona", "age": 27, "isUser": true }
```

Das JSON-Objekt wird durch geschweifte Klammern gekennzeichnet. Innerhalb des Objektes werden sogenannte “Schlüssel:Wert”- Paare durch Kommata getrennt. In diesem Beispiel sind “name”, “age” und “isUser” die Schlüssel. Den Schlüsseln können unter anderem Zahlen, Boolesche Werte, Strings und Arrays als Wert zugewiesen werden. Über den Schlüssel kann dann auf den zugehörigen Wert zugegriffen werden.

In Kapitel 5.2 *Implementierung* wird der genaue Austausch innerhalb der jeweiligen Komponente vorbereitet und ausgeführt.

5.1.2 Datenbank

Benötigte Mittel

Zur Implementierung des Digitalen Rezeptes müssen zwischen der App und Web-Anwendung Daten ausgetauscht werden. Für die zentrale Speicherung wird ein relationales Datenbanksystem verwendet. Das relationale Datenbanksystem besteht aus einem Datenbankmanagementsystem (DBMS) und aus einer Datenbank (DB). Das DBMS organisiert und verwaltet die Datensätze und ist für die unabhängige Lese- und Schreibzugriffe zuständig.

Innerhalb der Implementierung des Prototyps wird MySQL als DBMS verwendet. Über die XAMPP-Software wird ein lokaler Apache-Webserver verwendet, auf welchem MySQL ausgeführt wird.⁴⁷

Für eine übersichtliche Struktur der Daten wird die Datenbank basierend auf dem relationalen Datenbankmodell definiert. In der Datenbank werden alle Daten tabellarisch angeordnet und gespeichert. Um eine konsistente und redundanzfreie Datenbank zu erstellen, werden die Tabellen normalisiert und miteinander verknüpft.⁴⁸

Zur Daten-Administration wird mit einer Datenbanksprache, in diesem Fall über SQL, auf die gespeicherten Daten des DBMS zugegriffen. Im Rahmen der Prototyp-Umsetzung werden die

⁴⁶ Vgl. Karlstetter, Florian, 20.04.2018, Was ist JSON?, <https://www.cloudcomputing-insider.de/was-ist-json-a-704909/>, letzter Zugriff am 29.01.2019.

⁴⁷ Vgl. Apache Friends, 2019, Was ist XAMPP?, <https://www.apachefriends.org/de/index.html>, letzter Zugriff am 26.01.2019.

⁴⁸ Vgl. Kösterke, Jessy, 18.08.2017, Was sind eigentlich relationale Datenbanken?, <https://t3n.de/news/eigentlich-relationale-datenbanken-683688/>, letzter Zugriff am 26.01.2019.

SQL-Komponenten Data Definition Language (DDL) zur Herstellung der Tabellenstruktur und Data Manipulation Language (DML) zur Bearbeitung der Daten verwendet.⁴⁹

Aufbau

Das Relationale Datenbankmodell zählt als das weitverbreitetste Datenbankmodell.⁵⁰ Ein Relationales Datenbankmodell basiert auf einer Ansammlung von Tabellen. Die Daten werden in den Reihen der Datenbank eingetragen. Über die Spalten werden diese Daten ihrer Eigenschaft nach zugeordnet. Die sogenannten Attribute definieren einen Datensatz (Tupel) der Tabelle. Mit Hilfe von Primär- und Fremdschlüsseln werden die Datensätze eindeutig gekennzeichnet und die Tabellen in Beziehungen gesetzt.⁵¹

Für eine optimale Umsetzung und Nutzung der benötigten relationalen Datenbank wird zuerst ein Datenbankentwurf vor der Implementierung angefertigt. Für den Entwurf dient das Entity-Relationship-Modell (ERM) als Grundlage. Der Entwurf sorgt zunächst für einen gut strukturierten Überblick der Daten. Unter Verwendung des ERMs und der anschließenden Normalisierung, werden Redundanz und Inkonsistenz der Daten verhindert und eindeutige Verknüpfungen zwischen diesen Daten hergestellt.

In der Datenbank werden personenbezogene Daten über Patienten und Ärzte, sowie Anforderungs- und Rezeptdaten (*siehe Kapitel 2.3 Aufbau eines Kassenrezeptes*), gespeichert. Um Überschneidungen und Wiederholungen zu verhindern, werden diese in Entität-Typen (durch Rechtecke dargestellt) eingeteilt und über Beziehungen (durch Rauten dargestellt) miteinander verbunden. Über die Kardinalitäten wird die Art der Beziehung genauer definiert.

Für einen besseren Überblick werden die Attribute in *Tabelle 1* aufgelistet. Im folgenden ERM entfallen diese.

⁴⁹ Vgl. Litzel, Nico, 08.06.2017, Was ist ein Datenbankmanagementsystem?, <https://www.bigdata-insider.de/was-ist-ein-datenbankmanagementsystem-a-615034/>, letzter Zugriff am 26.01.2019.

⁵⁰ Vgl. Begerow, Markus, © 2010-2018, Relationales Datenbankmodell, <http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/relationales-datenbankmodell/>, letzter Zugriff am 28.01.2019.

⁵¹ Vgl. ebenda.

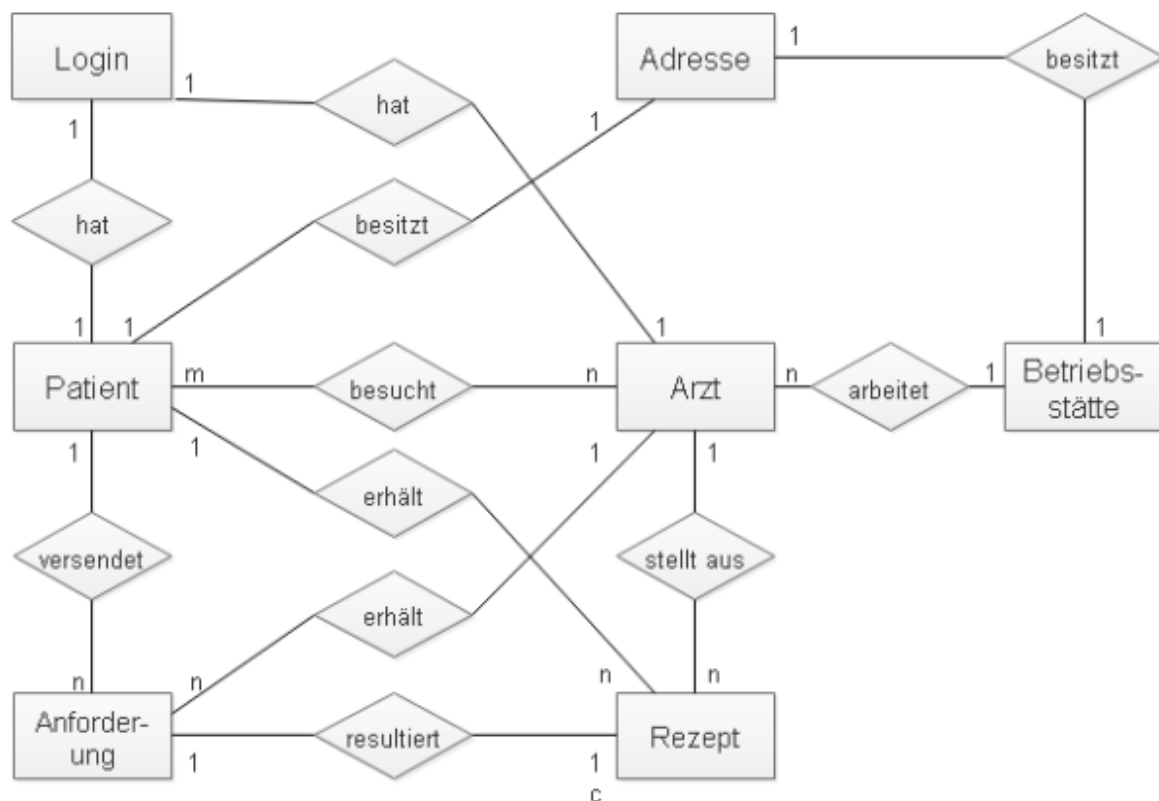


Abbildung 11: ERM, eigene Darstellung

In der Abbildung des ERMs hat z.B. jeder Patient und jede Arztpraxis, im Folgenden Betriebsstätte genannt, eine Adresse. Es kann durchaus vorkommen, dass mehrere Patienten⁵² die gleiche Adresse haben. Auch benötigt jeder Patient und jeder Arzt Logindaten. Diese werden in der Tabelle "Login" gesammelt und zugewiesen. Der Patient kann eine oder mehrere Anforderungen an den Arzt senden. Der Arzt wiederum erhält diese und kann darauf die Rezepte ausstellen.

Für den zukünftigen Einsatz des Prototyps sollte die Datenbank angepasst werden. Es wäre z.B. besser, wenn jede Betriebsstätte ein Konto mit mehreren angelegten Nutzern besitzt, so dass alle Angestellten mit unterschiedlichen Zugriffsrechten die Rezepte bearbeiten können. Denn nicht nur der Arzt, sondern auch die Arzthelfer, sind an dem Ausstellungs-Prozess beteiligt.

Nachdem das ERM erstellt wurde, muss das ERM normalisiert werden. D.h. das ERM muss in die Dritte Normalform (3NF) umgewandelt werden.⁵³

⁵² Familienmitglieder/-angehörige

⁵³ Vgl. Kösterke, Jessy, 18.08.2017, Was sind eigentlich relationale Datenbanken? Das Prinzip der Normalisierung, <https://t3n.de/news/eigentlich-relationale-datenbanken-683688/2/>, letzter Zugriff am 28.01.2019.

Nach der Optimierung des ERMs und Umwandlung in die Dritte Normalform entstehen folgende Relationen:

Relations-name	Attribute	Primär-schlüssel	Fremd-schlüssel
Adresse	id_adresse, Straße, Hausnummer, Zusatz, PLZ, Ort	id_adresse	-
Anforderung	id_anforderung, Status, Beschwerde, Medikament, VersichertenNr, LANR, id_rezept,	id_anforderung	VersichertenNr, LANR, id_rezept
Arzt	LANR, Name, Vorname, Bezeichnung, Signatur, BetriebsNr, id_login	LANR	BetriebsNr, id_login
Betriebsstätte	BetriebsNr, Praxisname, TelefonNr, id_adresse	BetriebsNr	id_adresse
Login	id_login, Nutzernamen, Passwort, Nutzerrolle	id_login	-
Patient	VersichertenNr, Name, Vorname, Geburtsdatum, Krankenkasse, id_adresse, id_login	VersichertenNr	id_adresse, id_login
Patient_besucht_Arzt	id_besuch, VersichertenNr, LANR	id_besuch	VersichertenNr, LANR
Rezept	id_rezept, Medikament, gebührenfrei, Form, Menge, Menge/Einheit, Ausstellungsdatum, noctu, autidem, VersichertenNr, LANR	id_rezept	VersichertenNr, LANR

Tabelle 1: Optimierte Relationen, eigene Darstellung

Jede entstandene Relation wird als eine Tabelle in der Datenbank angelegt. Hier werden zur Vollständigkeit alle Attribute in Klammern aufgelistet.

Die unterstrichenen Attribute bilden den Primärschlüssel (PK) der Relation zur Identifikation eines Datensatzes. Die Fremdschlüssel (FK) werden durch einen Strich über dem Attribut gekennzeichnet. Die FKs verweisen auf das jeweilige Attribut in einer anderen Tabelle.

5.1.3 App

An dieser Stelle wird der interne Aufbau der mobilen Applikation erläutert. Der Patient benötigt die App, um seinem zuständigen Arzt eine Rezeptanfrage zu senden. Nach der Ausschreibung des Rezeptes steht dieses in digitaler Form auf seiner Smartphone-App zum Einlösen in der Apotheke zur Verfügung.

Benötigte Mittel

Im Rahmen dieser Arbeit wurde eine Android-App mit der Entwicklungsumgebung Android Studios als Prototyp umgesetzt.⁵⁴ Die Software basiert auf der Programmiersprache Java. Zusätzlich zu Java gilt seit 2016 auch Kotlin als offizielle Sprache für Android.⁵⁵ Jedoch wurde von der Umsetzung einer dem Autor noch unbekannten Programmiersprache Kotlin abgesehen, da dies bei der Realisierung des Prototyps keine Vorteile hat. Doch von einem zukünftigen Einsatz zur Erweiterung der App mit Kotlin steht durch die hohe Kompatibilität mit Java nichts im Wege. Das Design der Benutzeroberflächen wird in Android Studio über einen (Visual layout) Editor und getrennte Extensible Markup Language (XML) -Dateien gestaltet. Für einen lauffähigen Prototypen wird ein Smartphone mit dem Betriebssystem Android benötigt. Für den Zugriff auf die Datenbank muss das Smartphone Zugang zum Internet haben. Je nach Einlöseart des Rezeptes in der Apotheke vor Ort, muss das Smartphone zusätzlich NFC-fähig sein.

Aufbau

Für einen strukturierten und übersichtlichen Aufbau der Benutzeroberflächen kommen Activities und Fragmente als Bestandteile der App zum Einsatz. Die meisten Android-Anwendungen, wie auch diese, bestehen in der Regel aus mehreren Activities. Eine Activity durchläuft während der Anwendung einen Lebenszyklus. Durch Lifecycle-Callbacks wird der aktuelle Zustand der Activity beschrieben.⁵⁶ Fragmente sind Bausteine, die in Activities wiederverwendet und kombiniert werden können. Jedes Fragment verfügt über einen eigenen Lebenszyklus und Lifecycle-Callbacks. Der Fragment Lifecycle wird von dem übergeordneten Activity Lifecycle stark beeinflusst. Wird bspw. eine Activity pausiert oder beendet, so werden auch die eingebetteten Fragmente pausiert oder beendet.⁵⁷

⁵⁴ Vgl. Google Developers, 2019, <https://developer.android.com/studio/>, letzter Zugriff am 07.02.2019)

⁵⁵ Vgl. Kotlin/JETBRAINS, 2019, Using Kotlin for Android Development, <https://kotlinlang.org/docs/reference/android-overview.html>, letzter Zugriff am 07.02.2019 & Schürmann, Tim, 27.09.2017, Kotlin ist auch nur eine Insel, <https://www.golem.de/news/programmiersprache-fuer-android-kotlin-ist-auch-nur-eine-insel-1709-130224.html>, letzter Zugriff am 07.02.2019.

⁵⁶ Vgl. Google Developers, 23.01.2019, Understand the Activity Lifecycle, <https://developer.android.com/guide/components/activities/activity-lifecycle>, letzter Zugriff am 07.02.2019.

⁵⁷ Vgl. Google Developers, 23.01.2019, Fragments, <https://developer.android.com/guide/components/fragments>, letzter Zugriff am 07.02.2019.

In den folgenden Sequenz-Diagrammen wird das Zusammenspiel der Activities und Fragmente grafisch dargestellt. Die vereinfachten Diagramme entsprechen nicht vollständig der aktuellen Implementierung der Anwendung. Für eine bessere Übersicht und zur Verdeutlichung des Ablaufs werden Klassen, wie z.B. Adapter- und Modell-Klassen, nicht mit aufgelistet. Diese Klassen sind ausschließlich für den Aufbau und die Darstellung der ListViews (Auflistungen) zuständig und sind zum Verständnis und für den prinzipiellen Ablauf in der Anwendung zu vernachlässigen.

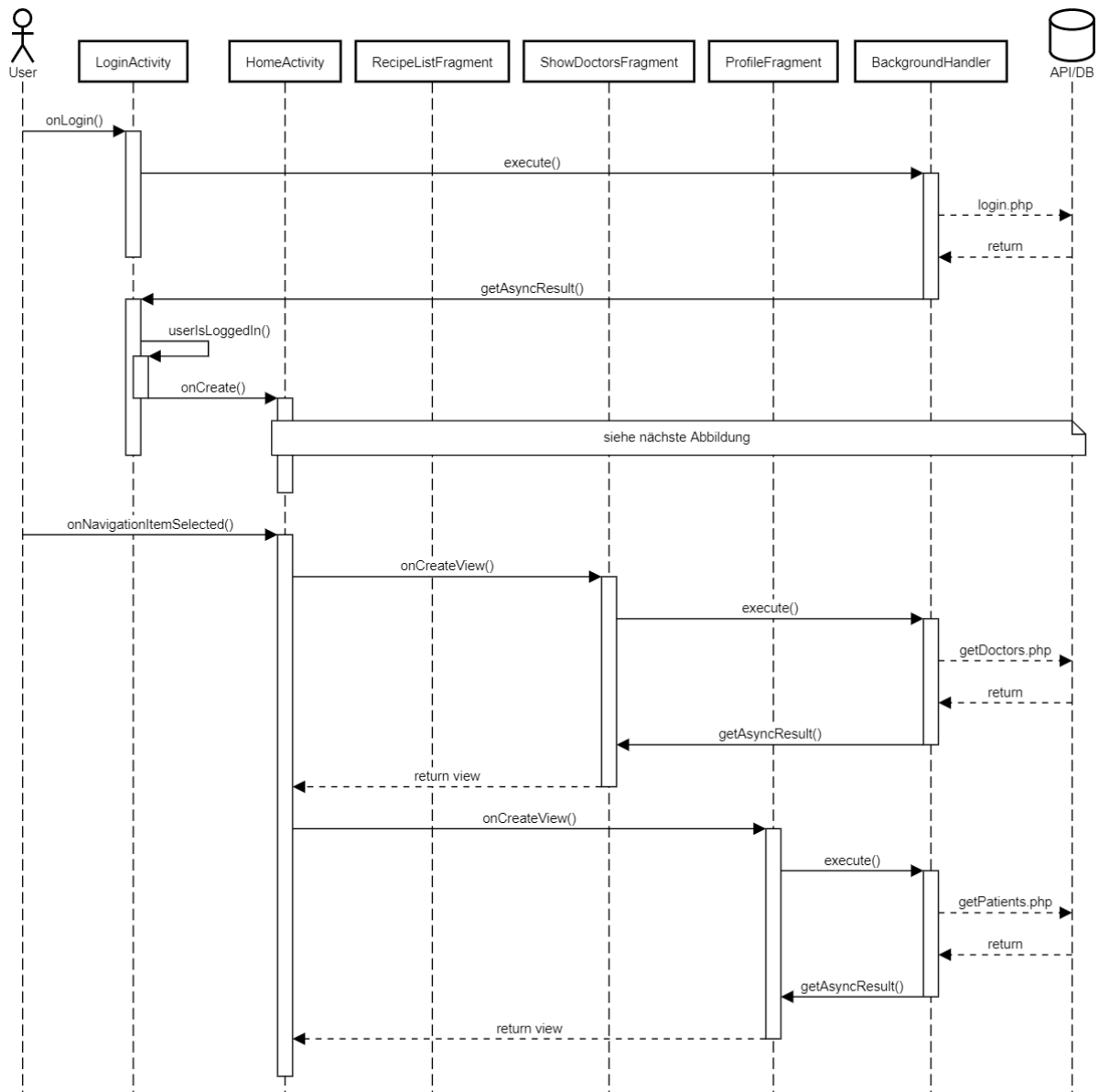


Abbildung 12: Sequenz-Diagramm der App: Login und grober Ablauf, eigene Darstellung

Beim Start der App wird die *LoginActivity* gestartet und auf dem Smartphone angezeigt. Dort können die Nutzerdaten über Textfelder eingegeben werden. Bei der Bestätigung der Eingaben durch einen Button, werden die Einträge über die *BackgroundHandler*-Klasse an die API *login.php*-Datei weitergeleitet. Dort werden die übermittelten Daten mit den bei der Registrierung gespeicherten Daten aus der Datenbank verglichen. Bei Übereinstimmung der Daten wird die Versichertennummer als User-ID an die App zurückgegeben und der Nutzer ist angemeldet. Die Registrierung wird bei der Prototyp-Umsetzung übersprungen. In der Datenbank wurden Benutzerangaben zum Test-Login eingepflegt. In Zukunft kann der Nutzer über einen Hinweis auf der *LoginActivity* zum Registrierungsvorgang gelangen und mit seinen persönlichen Angaben ein Konto erstellen.

Ist der Login erfolgreich abgeschlossen, wird die *LoginActivity* gelöscht und der Nutzer zur *HomeActivity* weitergeleitet. Im Zuge der Initialisierung der neuen Activity, wird die ID als String über ein Bundle-Objekt übergeben. Die User-ID wird bei den meisten folgenden Datenbank-anfragen mitgeschickt, um personenbezogene Datensätze zu filtern. Die *HomeActivity* setzt sich aus verschiedenen Fragmenten und einer Navigationsleiste am unteren Ende des Bildschirms zusammen. Über die Navigationsleiste kann zwischen den diversen Fragment-Ansichten gewechselt werden.

Als Standard-Fragment wird beim Starten der *HomeActivity* das *RecipeListFragment* geladen und angezeigt. Vor der Darstellung des Fragments greift die *HomeActivity* in ihrer *onCreate()*-Methode wieder über die *BackgroundHandler*-Klasse auf das API zu, um die benötigten Daten aus der Datenbank zu lesen. Die Abläufe innerhalb des Standard-Fragments werden in der Abbildung erklärt.

Neben dem *RecipeListFragment* kann über die Navigationsleiste zu anderen Fragment-Ansichten gewechselt werden. Wird der mittlere Tab der Navigationsleiste angewählt, startet das *ShowDoctorsFragment*. In dieser Ansicht erhält der Nutzer einen Überblick über die Ärzte, die das "Digitale Rezept" anbieten. Hierfür wird beim Initialisieren in der *onCreateView()*-Methode eine Anfrage an die Datenbank geschickt (*getDoctors.php*). Alle Datensätze, die den Ärzten zugeordnet sind, werden dann im Fragment aufgelistet. Für die Darstellung der Liste wird ein *DoctorListAdapter*-Objekt erzeugt, in dem jeder Datensatz einem Modell der *Doctor*-Klasse zugewiesen wird und die Anordnung der Daten bearbeitet wird. Als zukünftige Erweiterung soll hier der Nutzer einzelne Ärzte markieren und speichern können.

Das letzte Fragment innerhalb der *HomeActivity* ist das *ProfileFragment*. Auch hier wird in der *onCreateView()*-Methode auf die Datenbank zugegriffen. Über die *getPatients.php*-Datei und

die User-ID werden alle personenbezogenen Daten angefordert und auf einer Profilseite dargestellt. Als geplante Erweiterung soll der Nutzer seine Profildaten in der Smartphone App ändern und anpassen können.

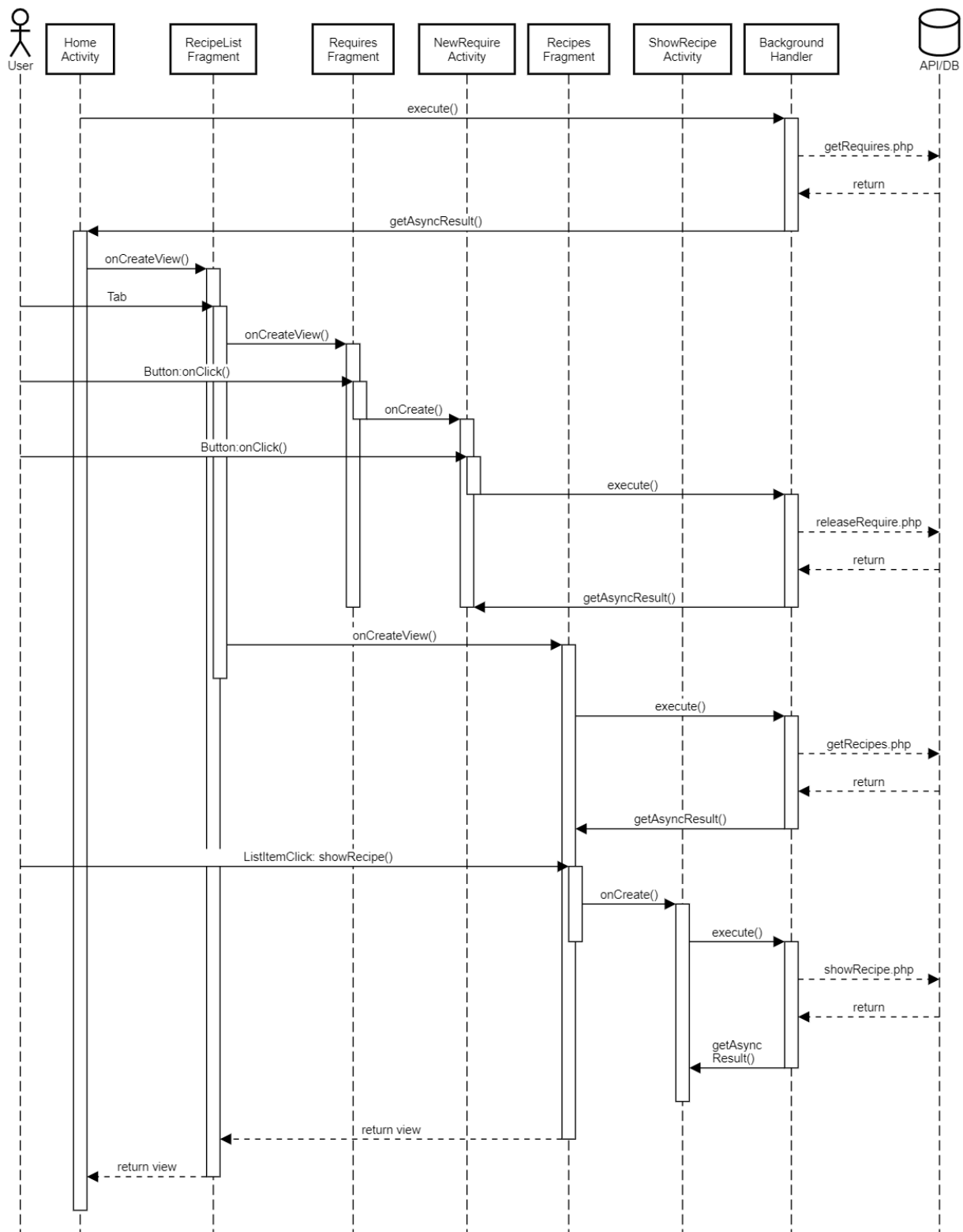


Abbildung 13: Sequenz-Diagramm der App: Ablauf der Rezept- und Anforderung-Darstellung, eigene Darstellung

Das *RecipeListFragment* besteht wiederum aus zwei Fragmenten, die in einem Tab-Layout angeordnet werden. Am oberen Ende des *RecipeListFragments* befindet sich eine Tab-Leiste, mit welcher zwischen der Anzeige der Rezept-Anforderungen und Rezept-Ausstellungen gewählt werden kann. In der *HomeActivity* wird vor dem Aufrufen eines weiteren Fragmentes eine Anfrage an das API versendet. In der *getRequires.php*-Datei werden über die User-ID alle Daten zu den Rezept-Anforderungen des Nutzers aus der Datenbank ausgelesen und an die *HomeActivity* zurückgegeben. Ein Bundle-Objekt leitet die Datensätze an das untergeordnete Fragment weiter. In der *RequiresFragment*-Klasse werden die Datensätze wieder über eine Adapter-Klasse, *RequireListAdapter.java*, aufgeteilt und aufgelistet. Über einen Button kann der User eine neue Rezept-Anforderung ausfüllen und erstellen. Hierfür wird die *NewRequireActivity*-Klasse gestartet. Neben Informationen über Art der Beschwerde und gewünschtem Medikament wird der zuständige Arzt angegeben, welcher die Anforderung bearbeiten soll. Beim Abschicken der Anforderung wird eine Anfrage zur *releaseRequire.php*-Datei der API gesendet und in die Datenbank eingetragen.

Wurde die Anforderung bearbeitet und bestätigt, wird diese für die Einlösung in der Datenbank bereitgestellt. Über das *RecipesFragment* können die bereits ausgestellten Rezepte eingesehen werden. Beim Initialisieren werden die Rezeptdaten wieder über den *BackgroundHandler* bei der API angefragt. In der *getRecipes.php*-Datei werden alle Rezept-Datensätze des Users ausgelesen und an die App zurückgegeben. Die richtige Anordnung und Auflistung der Daten wird über die *RecipeListAdapter*-Klasse behandelt und in dem Fragment angezeigt. Ist ein bestimmtes Rezept der Liste angewählt, wird durch die *showRecipe()*-Methode eine neue Activity gestartet. In der *ShowRecipeActivity* wird das Rezeptformular für die Einlösung dargestellt. Für einen schnelleren Ablauf werden vorher die Daten, die für die Auflistung bereits angefragt wurden, mit Hilfe eines Bundle-Objektes an die Activity übergeben. Die restlichen benötigten Daten werden in der *onCreate()*-Methode über die *showRecipe.php*-Datei der API angefragt. Nach der Übermittlung der Daten an die App werden diese im Rezeptformular angeordnet.

5.1.4 Web-Anwendung

In diesem Kapitel wird die Umsetzung der prototypbasierten Web-Anwendung beschrieben. Über die Web-Anwendung kann der Arzt alle Rezept-Anforderungen einsehen und bearbeiten. Bevor der Arzt die Rezeptdaten aus der Datenbank abrufen kann, muss er sich mit seinen Nutzerdaten anmelden.

Benötigte Mittel

Die Web-Anwendung wurde unter Verwendung von React umgesetzt. React ist eine JavaScript-Bibliothek, die zum Erstellen von grafischen Benutzeroberflächen eingesetzt wird. Die Software wird in trennbare Einheiten, sogenannte Komponenten, aufgeteilt. Komponenten sind JavaScript (JS) Funktionen, die als selbst definierte Hypertext Markup Language (HTML)-Tags eingebettet werden. Innerhalb einer Komponente wird neben dem Verhalten (mit JS) zusätzlich das GUI (mit HTML und CSS) implementiert. HTML-Tags werden im Javascript Syntax Extension (JSX)-Format dargestellt. Die Komponentenarchitektur ist in React hierarchisch aufgebaut.⁵⁸

Für eine möglichst performante Darstellung des GUIs werden Änderungen durch einen virtuellen Document Object Model (DOM) selektiv aktualisiert.⁵⁹

Die Prototyp-Web-Anwendung läuft aktuell ebenfalls auf einem lokalen Server. Für die Programmierung und Ausführung einer solchen serverseitigen Web-Anwendung dient das Framework *Node.js*. Über den Node Package Manager (npm) wurde die JavaScript-Bibliothek React installiert und die Anwendung erstellt. Mit "npm start" kann diese schließlich auf dem *Node.js*-Server lokal ausgeführt werden.⁶⁰

Für den zukünftigen Gebrauch wird die Anwendung auf einem öffentlichen Server installiert und erreichbar sein. Da für diese Anwendung aktuell keine Anbindung an die Telematikinfrastruktur möglich ist, benötigt der Arzt lediglich einen Computer mit Internetzugang.

⁵⁸ Vgl. Facebook Inc., 2018, Tutorial: Intro to React, <https://reactjs.org/tutorial/tutorial.html#what-is-react>, letzter Zugriff am 11.02.2019.

⁵⁹ Vgl. Tillmann, Janna, 22.09.2017, Was ist eigentlich ein (virtuelles) DOM?, <https://t3n.de/news/eigentlich-virtuelles-dom-858160/>, letzter Zugriff am 11.02.2019.

⁶⁰ Vgl. Codecademy, 2019, React Setup, Part 1: React and React DOM, Install React.js On Your Local Computer, <https://www.codecademy.com/articles/react-setup-i>, letzter Zugriff am 11.02.2019.

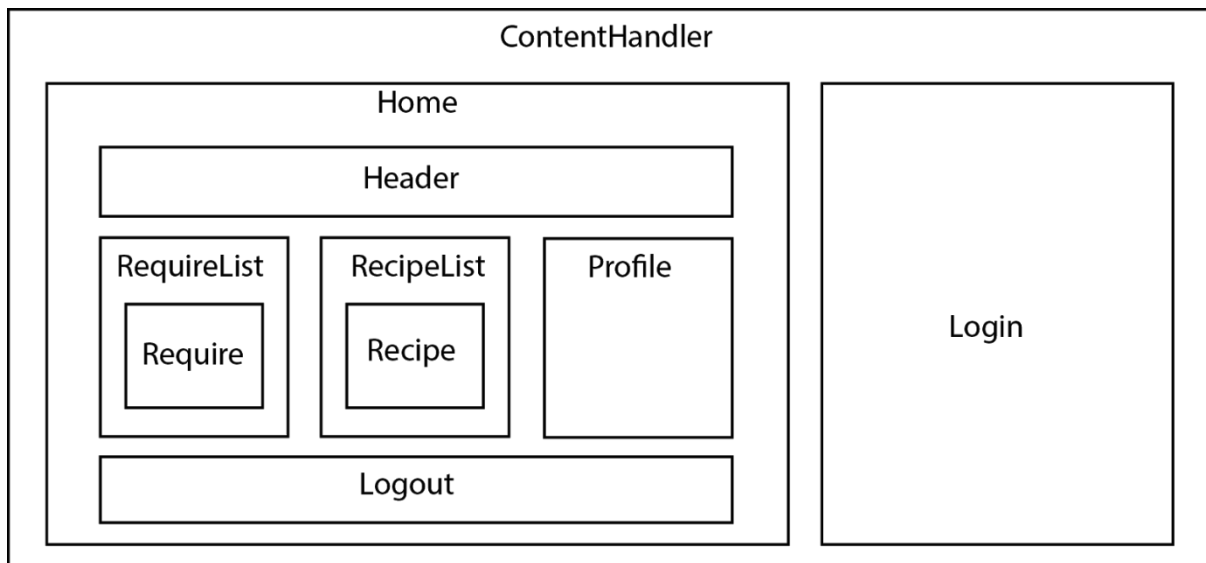
Aufbau

Abbildung 14: Komponenten-Hierarchie der Web-Anwendung, eigene Darstellung

In dieser Grafik stellen die Rechtecke einen Teil der jeweiligen Komponenten dar. Diese sind hierarchisch verschachtelt und bauen aufeinander auf. Durch den Einsatz einzelner Komponenten können Benutzeroberflächen und passende Funktionen zusammengefasst werden und dadurch ein flexibles Design erstellt werden.

Die Web-Anwendung wird über die *index*-Datei im Browser ausgeführt. Über ihre *render()*-Methode werden die Komponenten ihrer Hierarchie nach ausgeführt und dargestellt. In der *Root*-Komponente werden die Routen für die Verlinkungen der Komponenten festgelegt. So können die Komponenten nicht nur über die selbst definierten HTML-Tags, sondern auch über Verlinkungen, eingebettet werden. Über die URL können Parameter an die untergeordnete Komponente übergeben werden.

Die nächste untergeordnete Komponente ist die *ContentHandler*-Komponente. Hier wird geprüft, ob der Nutzer angemeldet ist oder nicht. Ist der Nutzer angemeldet, wird er zur *Home*-Komponente weitergeleitet. Innerhalb der *Home*-Komponente kann der Nutzer auf weitere Funktionen der Web-Anwendung zugreifen.

Die wichtigste Methode innerhalb einer Komponente ist die *render()*-Methode. In der *render()*-Methode können Komponenten, JavaScript-Code und HTML-Tags im JSX-Format implementiert werden. Bei selektiven Änderungen wird diese aufgerufen und aktualisiert die betroffenen Komponenten der Benutzeroberfläche.⁶¹

⁶¹ Vgl. Facebook Inc., 2019, ReactDOM, <https://reactjs.org/docs/react-dom.html#render>, letzter Zugriff am 04.03.2019.

Auch bei diesem Sequenz-Diagramm werden einzelne Komponenten und Methoden zugunsten einer übersichtlichen Darstellung vernachlässigt und nicht grafisch dargestellt.

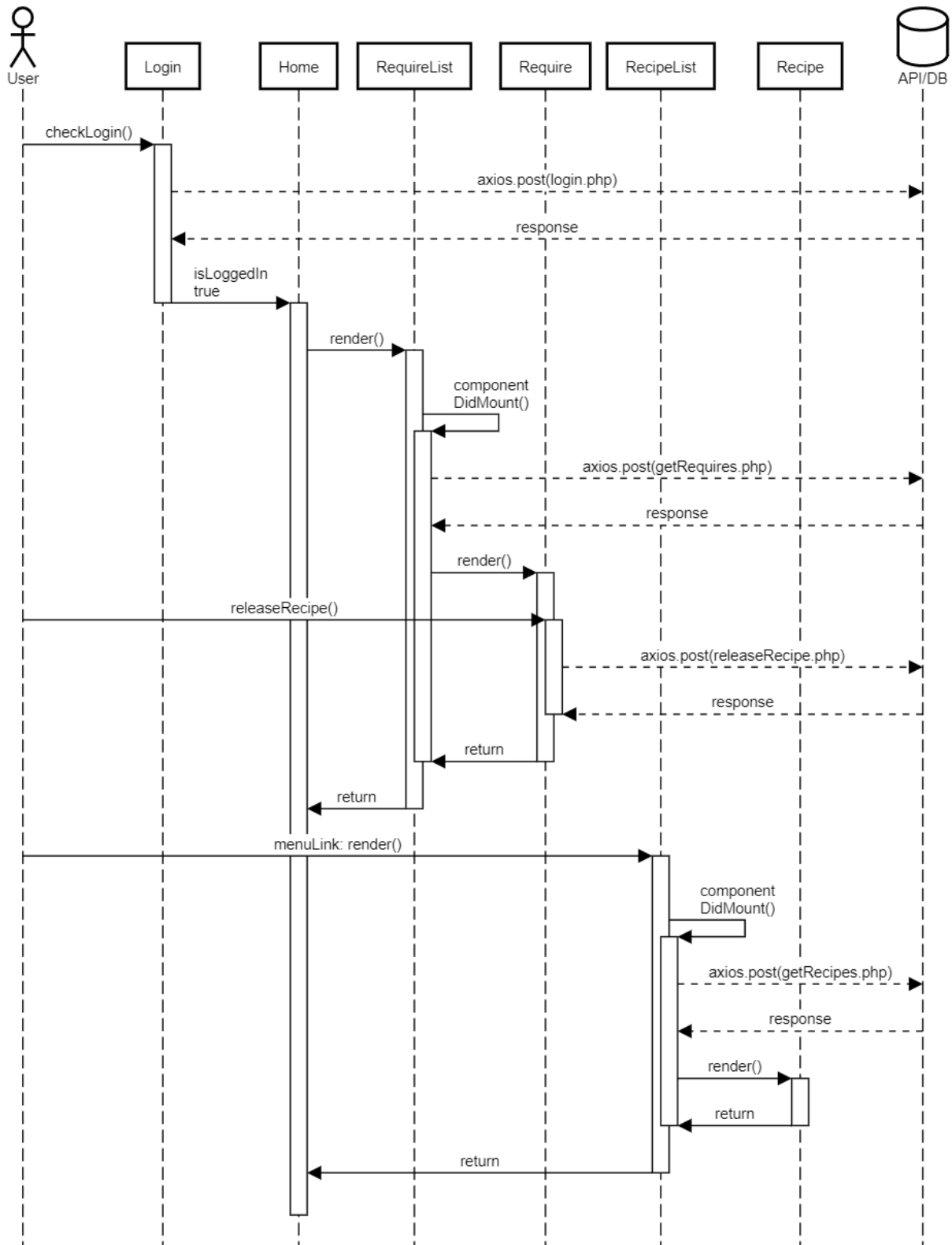


Abbildung 15: Sequenz-Diagramm der Web-Anwendung, eigene Darstellung

Die erste Verlinkung führt zur *ContentHandler*-Komponente. Diese zeigt, je nach Login-Status, die *Login*- oder *Home*-Komponente an. Ist der Nutzer noch nicht angemeldet, wird er zur *Login*-Komponente weitergeleitet und ihm werden Eingabefelder für Nutzernamen und Passwort angezeigt. Wurden die Felder ausgefüllt und mit einem Button bestätigt, wird über die *checkLogin()*-Methode die erste Datenbankabfrage in Form eines Hypertext Transfer Protocol (HTTP)-Requests an die *login.php*-Datei der API ausgeführt. Stimmen die eingegebenen Nutzerdaten mit den Login-Daten in der Datenbank überein, wird innerhalb der Anwendung zur *Home*-Komponente verlinkt. Die *Header*- und *Logout*-Komponente sind fest in der *Home*-Klasse implementiert. Über ein Menü kann der Nutzer zu drei verschiedenen Ansichten gelangen. Mit Klick auf den ersten Menü-Eintrag wird die *RequireList*-Komponente gestartet. In ihrer *componentDidMount()*-Methode werden alle Daten der Rezept-Anforderungen abgerufen, die an den zuständigen Arzt gesendet wurden.

Nachdem die angefragten Daten zurückgesendet wurden, wird in der *render()*-Methode für jeden Datensatz eine *Require*-Komponente erstellt und untereinander aufgelistet. Jedes Element der Liste zeigt spezifische Daten über die jeweilige Rezept-Anforderung. Der Arzt kann diese Informationen bewerten und über einen Button die Anforderung bestätigen und somit das Rezept ausstellen. Bestätigt er die Anforderung, wird die *releaseRecipe()*-Methode ausgeführt, in der über einen weiteren HTTP-Request ein neuer Datensatz für das ausgestellte Rezept in die Datenbank eingefügt wird.

Möchte der Arzt die ausgestellten Rezepte einsehen, gelangt er über den zweiten Menüeintrag zur *RecipeList*-Komponente. Auch hier werden vor dem Rendern innerhalb der *componentDidMount()*-Methode alle Daten der ausgestellten Rezepte angefragt. Sind die Daten an die Komponente zurückgegeben, werden diese über *Recipe*-Komponenten einzeln aufgelistet. Eine mögliche zukünftige Erweiterung ist eine Bearbeitungsfunktion. Der Arzt könnte eventuell fehlerhaft ausgestellte Rezepte zurückziehen oder bearbeiten.

Im letzten Menüeintrag bekommt der Nutzer seine persönlichen Angaben auf einer Profilseite angezeigt. In einer weiteren zukünftigen Erweiterung könnte der Nutzer hier sein Profil bearbeiten und seine Daten aktualisieren.

5.2 Implementierung

Der Austausch von Daten zwischen den Komponenten und deren Speicherung spielt bei der Umsetzung des Prototyps eine wichtige Rolle. In diesem Abschnitt wird die Implementierung der Datenbankbindung und Datenbankabfragen der Client-Anwendungen teilweise anhand von Code-Ausschnitten beschrieben. Zwischen den Code-Ausschnitten und dem tatsächlichen Code kann es zu Abweichungen kommen.

Für jede Software-Anwendung wurde ein Git-Repository erstellt. Git erleichtert die Versionsverwaltung der verschiedenen Anwendungen während der Entwicklungsphase, späteren Erweiterungen und Fehlerkorrekturen.

5.2.1 API

Das im Rahmen des Prototyps implementierte Application Programming Interface (API) fungiert als Programmierschnittstelle zwischen der MySQL-Datenbank und der jeweiligen Software-Anwendung (App/Website). Das API besteht aus mehreren Hypertext Preprocessor (PHP)-Dateien, die auf einem lokalen Server liegen.

Für einen geregelten Informationsaustausch ist die Struktur der definierten Syntax in fast allen Klassen nach einem ähnlichen Grundkonzept aufgebaut.

Die Dateien der API werden über HTTP-Requests der Client-Anwendungen aufgerufen. Als Sicherheitselement werden Zugriffe von clientseitigen Web-Anwendungen durch die Same-Origin-Policy (SOP) eingeschränkt. Anwendungen mit einer unbekannten Domain müssen mit Hilfe von Cross-Origin Resource Sharing (CORS) zuerst zugelassen werden. Damit die Ärzte über die Web-Anwendung Anfragen an die Datenbank senden können, müssen einige Einstellungen im Header vorgenommen werden. Da für den Prototypen alles auf lokalen Servern ausgeführt wird, werden aktuell mit "Access-Control-Allow-Origin: *" alle Domains von anderen Servern zugelassen. Erst wenn die Server mit den installierten Anwendungen mit dem Internet verbunden sind, müssen zum sicheren Betrieb die entsprechenden Domains freigegeben werden.⁶²

```
header('Access-Control-Allow-Origin: *');  
header('Access-Control-Allow-Headers: Content-Type');  
  
require "dbconnection.php";
```

Listing 1: API: CORS-Einstellungen in getRequires.php

⁶² Vgl. Mozilla, 2005-2019, Cross-Origin Resource Sharing (CORS) <https://developer.mozilla.org/de/docs/Web/HTTP/CORS>, letzter Zugriff am 01.02.2019.

Sind alle Klassen der API für die Anwendungen erreichbar, können Anfragen übermittelt werden. Vor jedem Datenaustausch muss eine Verbindung zur Datenbank aufgebaut werden. Dafür wird die Datei *dbconnection.php* eingebunden und ausgeführt.

In der über “*require*” eingefügten Klasse, wird mit der *mysqli_connect()*-Methode eine Verbindung zur Datenbank aufgebaut. Für die Verbindung müssen der Host, die Nutzerangaben und der Name der Datenbank angegeben werden. Da die Prototyp-Anwendung im Moment auf einem lokalen Server ausgeführt wird, muss der Parameter “localhost” übergeben werden. Wird die Anwendung in Zukunft auf einem im Internet frei zugänglichen Server ausgeführt, wird die Domain des Servers eingesetzt. Zur Authentifizierung des Nutzers und zur Bestimmung seiner Zugriffsrechte werden Nutzernamen und Passwörter angegeben. Da es aktuell keine Einschränkungen der Zugriffsrechte gibt, werden die voreingestellten Daten übernommen. Anschließend wird für die Verbindung der Name “*digitales_rezept*”, der im vorherigen Kapitel erstellten Datenbank, angegeben.

```
$conLink = mysqli_connect($dbServer, $dbUsername, $dbPassword, $dbName);
```

Listing 2: API: Datenbankverbindung in dbconnection.php

Nachdem die Verbindung zur Datenbank aufgebaut wurde, wird die Datenbankabfrage vorbereitet. In den meisten Fällen werden für die Abfrage Parameter benötigt, die im JSON-Format von einer der App- oder Web-Anwendungen bei der HTTP-Anfrage auf das API mitgegeben wurden. Über das *\$_POST*-Array und einen Schlüssel, im Beispiel “*userRole*” und “*userID*”, kann der zugehörige Wert ausgelesen und in einer Variablen für die weitere Verwendung zwischengespeichert werden.

```
$user_role = $_POST["userRole"];  
$user_ID = $_POST["userID"];
```

Listing 3: API: JSON-Daten in getRequires.php.

Für ein übersichtliches und einheitlich strukturiertes API wird von der App-Anwendung immer “*Patienten*” und von der Web-Anwendung “*Aerzte*” als Wert für den “*userRole*”-Schlüssel eingesetzt. Dadurch können in jeder Klasse, trotz gleichem Grundverhalten, eventuell vorhandene Unterschiede abgefangen und individuell bearbeitet werden.

Zum Beispiel hat die *getRequires.php*-Datei die Aufgabe, alle Anforderungen aus der Datenbank zu lesen und an die Anwendung zurück zu geben. Der Patient darf jedoch nur die Anforderungen einsehen, die er selbst erstellt hat. Diese Anforderungen kann der Patient an verschiedene Ärzte verschickt haben. Der zuständige Arzt möchte jedoch nur die Anforderungen einsehen, die er von seinen Patienten erhalten hat. Das Grundverhalten der Klasse weist für beide Anwendungsfälle Parallelen auf. Doch die gesuchten Datensätze sind nicht identisch, da sich die Datenbankabfragen voneinander unterscheiden.

Im folgendem Quellcode-Ausschnitt wird mit Hilfe einer *if*-Abfrage zwischen den Nutzerrollen (Patienten, Aerzte) unterschieden und das SQL-Kommando entsprechend angepasst. Innerhalb der SQL-Anfrage werden die Datensätze mit Hilfe von *JOINS* und den Fremdschlüsseln strukturiert und miteinander verknüpft. Anschließend wird über die *mysqli_query()*-Methode die SQL-Anfrage über die bestehende Verbindung an die Datenbank weitergeleitet.⁶³

```
if($user_role == "Patienten"){
    $mysql_qry = "SELECT * FROM (Anforderungen LEFT JOIN Aerzte ON LANR_fk = LANR)
                WHERE versichertenNr_fk LIKE '$user_ID';";
} else if($user_role == "Aerzte"){
    $mysql_qry = "SELECT * FROM (Anforderungen LEFT JOIN Patienten
                ON versichertenNr_fk = versichertenNr) WHERE LANR_fk LIKE '$user_ID';";
}

$result = mysqli_query($conLink, $mysql_qry);
```

Listing 4: API: SQL-Befehl in *getRequires.php*

Neben dem SQL-*SELECT*- Befehl gibt es noch weitere Befehle, wie *INSERT*, *UPDATE* und *DELETE*, die Bestandteile der Data Manipulation Language (DML) sind und innerhalb der API zum Einsatz kommen. Die DML ist ein Teil der SQL-Datenbanksprache, welche verwendet wird, um Daten zu manipulieren (lesen, schreiben, ändern, löschen).⁶⁴ Der *INSERT*-Befehl wird z.B. verwendet, wenn der Arzt die Anforderung eines Patienten bestätigt und ein Rezept ausstellt. Der neue Datensatz über das ausgestellte Rezept wird dann in die Tabelle "Rezepte" eingefügt. Wurde der neue Datenbankeintrag erfolgreich abgeschlossen, muss die bearbeitete Anforderung anschließend abgeändert werden. Dazu wird ein *UPDATE*-Befehl an die Datenbank übermittelt, mit dem ein Attribut der Tabelle "Anforderungen" mit "true" initialisiert wird. Über diesen Wert kann nachträglich festgestellt werden, ob die Anforderung erfolgreich bearbeitet wurde und den Nutzer ggf. darüber informieren.

```
$mysql_qry = "INSERT INTO Rezepte(
            id_rezept, medikament_name, medikament_form,
            medikament_menge, versichertennummer_fk, LANR_fk
        )
        VALUES(
            '$require_ID', '$users_medicine', 'Tabletten', '30 Stück', '$ver_number', '$user_ID'
        );";

$mysql_qry = "UPDATE Anforderungen SET id_rezept_fk = '$require_ID', status = 'true'
            WHERE id_anforderung = '$require_ID' ";
```

Listing 5: API: SQL-Befehle in *releaseRecipe.php*

⁶³ Vgl. The PHP Group. 2001-2019, *mysqli_query*, <http://php.net/manual/de/mysqli.query.php>, letzter Zugriff am 01.02.2019.

⁶⁴ Vgl. Datenbanken-verstehen.de, 2010-2018, DML, DDL, DCL, SQL-Kommandos, <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/dml-ddl-dcl-kommandos/>, letzter Zugriff am 01.02.2019.

Nachdem die Datenbankabfrage vollständig und korrekt ausgeführt wurde und die angefragten Daten gelesen und zurückgegeben wurden, wird in den meisten Klassen der API die selbst angelegte *mysqli_result()*-Methode ausgeführt. Innerhalb dieser Funktion werden mit der *mysqli_fetch_all()*-Methode zuerst die aus der Anfrage resultierenden Daten als Array zurückgegeben.⁶⁵

In einer *for*-Schleife wird jedem Element im Array ein Schlüssel-Wert zugewiesen und in einem neuen Array zwischengespeichert. Auch beim Zuweisen der Werte werden die Nutzerrollen beachtet und die Schlüssel individuell angepasst. Vor der Übertragung der vom Server abgefragten Datensätze zum Client, wird das Array mit den "Schlüssel:Wert"- Paaren mit der *json_encode()*-Methode in einen JSON-kodierten String umgewandelt.⁶⁶

```
for($i=0;$i<count($data);$i++){
    if ($user_role == "Patienten"){
        $db_data[$i] = ['id'=>$data[$i][0], 'complaint'=>$data[$i][1], 'med_name'=>$data[$i][2],
            'ver_number'=>$data[$i][3], 'LANR_fk'=>$data[$i][4],
            'id_recipe_fk'=>$data[$i][5], 'status'=>$data[$i][6],
            'doc_lastName'=>$data[$i][8], 'doc_firstName'=>$data[$i][9]];
    } else if ($user_role == "Aerzte"){
        $db_data[$i] = ['id'=>$data[$i][0], 'complaint'=>$data[$i][1], 'med_name'=>$data[$i][2],
            'ver_number'=>$data[$i][3], 'LANR_fk'=>$data[$i][4],
            'id_recipe_fk'=>$data[$i][5], 'status'=>$data[$i][6],
            'pat_lastName'=>$data[$i][8], 'pat_firstName' => $data[$i][9]];
    }
}

return json_encode($db_data);
```

Listing 6: API: *mysqli_result()*-Methode in *getRequires.php*

Am Ende jeder Klasse wird die Verbindung zu Datenbank mit einer *close()*-Methode beendet.

⁶⁵ Vgl. The PHP Group, 2001-2019, *mysqli_result::fetch_all*, <http://php.net/manual/de/mysqli-result.fetch-all.php>, letzter Zugriff am 01.02.2019.

⁶⁶ Vgl. w3schools.com, 1999-2019, JSON PHP, https://www.w3schools.com/js/js_json_php.asp, letzter Zugriff am 01.02.2019.

5.2.2 App

Alle Funktionen einer Android-App laufen grundsätzlich im Mainthread. Um eine optimale Nutzung der App zu gewährleisten und eine mögliche Blockade des Mainthreads zu vermeiden, werden die Anfragen an die Datenbank in einem Hintergrundthread parallel ausgeführt.

Dafür wird in der Android-App ein asynchroner Task integriert. Die *BackgroundHandler*-Klasse der Anwendung wird durch die Superklasse *AsyncTask* erweitert und erbt von dieser alle Methoden, in welchen die Anfrage schließlich vorbereitet und ausgeführt wird.⁶⁷

Alle Datenbankanfragen zum Apache-Server des *BackgroundHandlers* laufen über die API-Schnittstelle. Damit die App auf das API zugreifen kann, müssen vor dem Zugriff in der App Einstellungen vorgenommen werden. Damit die App Informationen über den Status der Netzwerk-Verbindung anfragen kann und Zugriff zum Internet hat, muss die Berechtigung für den Internetzugang aus Sicherheitsgründen im *AndroidManifest* eingetragen werden.⁶⁸

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Listing 7: App: *AndroidManifest.xml*

Nach Freigabe der Berechtigung kann der HTTP-Request in dem Hintergrundthread ausgeführt werden. Um die *AsyncTask*-Klasse zu starten und somit den HTTP-Request auszuführen, muss ein Objekt der *BackgroundHandler*-Klasse in einer Activity oder Fragment des Mainthreads instanziiert werden. Über die *execute()*-Methode wird der Hintergrundthread gestartet. Über die Methode werden Parameter übergeben, die für die weitere Bearbeitung der Datenbankanfrage benötigt werden.

```
type = "showRecipe";
String userrole = "Patienten";

backgroundHandler = new BackgroundHandler(this);
backgroundHandler.execute(type, userrole, userID, recipeID);
```

Listing 8: App: *ShowRecipeActivity, onCreate()*

Bei einer asynchronen Task-Ausführung werden die *onPreExecute()*-, *doInBackground()*-, *onProgressUpdate()*- und *onPostExecute()*-Methode in dieser Reihenfolge durchlaufen. Die *onPreExecute()*- und *onProgressUpdate()*-Methode sind für den Prototypen zu vernachlässigen. Im Folgenden werden die Datenanfragen und die Verarbeitung der Daten behandelt. Bei den meisten Datenbankanfragen ist der interne Ablauf sehr ähnlich. In den folgenden Code-Ausschnitten wird der Ablauf beispielhaft anhand der Rezept-Darstellung erklärt. In diesem Beispiel sollen alle Daten angefragt werden, die zum Einlösen des Rezeptes benötigt werden.

⁶⁷ Vgl. Google Developers, 2019, *AsyncTask*, <https://developer.android.com/reference/android/os/AsyncTask>, letzter Zugriff am 09.02.2019.

⁶⁸ Vgl. Google Developers, 2019, *Manifest.permission*, <https://developer.android.com/reference/android/Manifest.permission>, letzter Zugriff am 09.02.2019.

In der *doInBackground()*-Methode werden zunächst die übermittelten Parameter ausgelesen und verarbeitet. Der erste Parameter wird mit Verwendung von *if*-Statements verglichen und gibt immer den Typ der Anfrage an, in diesem Fall "showRecipe". Die übrigen Parameter werden in einem String mit dem Variablennamen *post_data* eingebunden und Schlüsseln zugewiesen, über die der Wert dann in der API gelesen werden kann. Vom Anfrage-Typ sind die Zusammensetzung des Strings und die URL der API-Datei abhängig, die mit dem HTTP-Request gesendet werden.

```

if ( ... ){
    [...]
} else if (type.equals("showRecipe")){
    str_userRole = params[1];
    String str_userID = params[2];
    String str_recipeID = params[3];

    try { post_data = URLEncoder.encode("userRole", "UTF-8")+"="
        +URLEncoder.encode(str_userRole, "UTF-8")+"&"
        +URLEncoder.encode("userID", "UTF-8")+"="
        +URLEncoder.encode(str_userID, "UTF-8")+"&"
        +URLEncoder.encode("recipeID", "UTF-8")+"="
        +URLEncoder.encode(str_recipeID, "UTF-8");
    } catch (UnsupportedEncodingException e) {
        //Fehlerbehandlung
    }

    setAPIConnection(showRecipe_url, "POST");

    postingData(post_data);
    return receivingData();
}

```

Listing 9: App: BackgroundHandler, *doInBackground()*

Für die Ausführung eines solchen Requests wird in der *setAPIConnection()*-Methode die Verbindung zur API aufgebaut. Über eine Instanz der *HttpURLConnection*-Klasse werden die URL und die *Request*-Methode gesetzt. Aktuell wird bei allen Anfragen die *POST*-Methode zum Senden und Empfangen von Daten verwendet.⁶⁹ Die vom Typ anhängige URL leitet die Verbindungs-Anfrage an die *showRecipe.php*-Datei der API.

```

httpURLConnection = (HttpURLConnection) url.openConnection();
httpURLConnection.setRequestMethod(request_method);

```

Listing 10: App: BackgroundHandler, *setAPIConnection()*

⁶⁹ Vgl. Google Developers, 2019, Connect to the network, <https://developer.android.com/training/basics/network-ops/connecting>, letzter Zugriff am 09.02.2019.

Über diese Verbindung wird dann mit der *postingData()*-Methode der String mit den enthaltenen Daten übertragen.

```
outputStream = httpURLConnection.getOutputStream();
bufferedWriter = new BufferedWriter( new OutputStreamWriter(outputStream,"UTF-8") );

bufferedWriter.write(post_data);
bufferedWriter.flush();
```

Listing 11: App: BackgroundHandler, postingData()

Nachdem die Datenanfrage bearbeitet wurde, werden diese Daten an die mobile App zurückgesendet. Mit einem *InputStream*- und *BufferedReader*-Objekt wird das Ergebnis der Datenbank gelesen und nach dem Beenden der Verbindung als Rückgabewert der *doInBackground()*-Methode deklariert.

```
inputStream = httpURLConnection.getInputStream();
bufferedReader = new BufferedReader( new InputStreamReader(inputStream,"iso-8859-1") );

String line = "";
while ((line = bufferedReader.readLine()) != null){
    result += line;
}

bufferedReader.close();
inputStream.close();
httpURLConnection.disconnect();

return result;
```

Listing 12: App: BackgroundHandler, receivingData()

Der Rückgabewert der *doInBackground()*-Methode wird anschließend an die *onPostExecute()*-Methode übergeben. Dort werden bekannte Fehlermeldungen abgefangen und das Ergebnis als Array im JSON-Format über eine *Callback*-Methode an den Mainthread transferiert.

```
jsonArray = new JSONArray(result);
asyncCallback.getAsyncResult(jsonArray, type);
```

Listing 13: App: BackgroundHandler, onPostExecute()

Damit die Activities und Fragmente des Mainthreads diese *Callback*-Methode erben und darauf zugreifen können, werden diese durch das *AsyncTaskCallback*-Interface erweitert.

In der überschriebenen *getAsyncResult()*-Methode wird jedes Objekt des JSON-Arrays gelesen und verarbeitet. Die darin enthaltenen Daten werden über die gesetzten Schlüssel-Werte den Textfeldern im Rezeptformular zugewiesen.

```
JSONObject jsonObject = jsonArray.getJSONObject(i);

String patInsuranceNr = (String) jsonObject.get("ver_number");
String patInsurance = (String) jsonObject.get("pat_insurance");
String patLastName = (String) jsonObject.get("pat_lastName");
String patFirstName = (String) jsonObject.get("pat_firstName");
[ ... ]
```

Listing 14: App: ShowRecipeActivity, getAsyncResult()

Nachdem das Rezeptformular ausgefüllt wurde, wird der Hintergrundthread mit der *cancel()*-Methode beendet.

Innerhalb der Anwendungen werden von den unterschiedlichsten Stellen Daten angefragt. Würden alle Daten direkt beim Starten der App abgerufen, könnte es bei großen Datenmengen zu Verzögerungen im Ablauf kommen. Für verteilte Anfragen muss der Nutzer allerdings über eine ständige Internetverbindung verfügen, mit dem Vorteil schnellerer Hintergrundprozesse und aktuellerer Daten.

5.2.3 Web-Anwendung

Die Datenanfrage an den Apache-Server und die Weiterverarbeitung der daraus resultierenden Daten innerhalb der Web-Anwendung werden beispielhaft anhand der Darstellung der Rezept-Anforderungen erklärt.

Möchte der Arzt die aktuellen Rezept-Anforderungen einsehen, gelangt er über das Menü zur *RequireList*-Komponente. Bevor die Anforderungen aufgelistet und gerendert werden können, müssen die Daten in der *componentDidMount()*-Methode angefragt werden.

Die *componentDidMount()*-Methode ist eine Reacts Lifecycle-Hook-Funktion, die clientseitig einmalig vor der *render()*-Methode aufgerufen wird. So wird sichergestellt, dass alle Daten vor der Darstellung komplett vorhanden sind.

Zur Ausführung der Anfrage wird die Axios-Bibliothek importiert. Axios ist eine clientseitige JavaScript Bibliothek, mit der asynchrone HTTP-Requests gesendet werden. In die *post()*-Methode werden die URL und Daten im JSON-Format als Parameter übergeben. Die URL gibt den Endpunkt der Anfrage an, welcher die JSON-Daten verarbeiten soll. Nach der Datenbank-anfrage über das API werden die angefragten Daten zurückgegeben und im *requires*-Array zwischengespeichert.

```
componentDidMount(){
    axios.post('http://localhost/API_Data/getRequires.php', {
        userRole: 'Aerzte',
        userID: this.state.userID
    }).then((response) => {
        this.setState({requires: response.data});
    }).catch((error) => {
        ...
    })
}
```

Listing 15: Web-Anwendung: *RequireList.js*

Für die Auflistung der einzelnen Anforderungen, wird jedes Element im Array mit der *map()*-Methode aufgerufen und für jeden Datensatz wird eine *Require*-Komponente erstellt. Bei der Instanziierung der *Require*-Komponente werden die Werte aus der Datenbank über die zugehörigen Schlüssel-Paare als Parameter übergeben.

```
return (
    <div>
        {this.state.requires.map(require =>
            <Require
                key = {require.id}, patFirstName={require.pat_firstName},
                patLastName={require.pat_lastName}, verNumber={require.ver_number},
                id={require.id}, medicine={require.med_name},
                complaint={require.complaint}, recipeFK={require.id_recipe_fk},
                statusReleased={require.state}, docID={this.state.userID}
            />
        )}
    </div>
);
```

Listing 16: Web-Anwendung: *RequireList.js*, *render()*

Jede *Require*-Komponente ist gleich aufgebaut und ordnet die übergebenen Parameter dementsprechend an. Jetzt kann der Arzt alle Anforderungen einsehen und schließlich bearbeiten.

6 Fazit und Ausblick

Ziel und Ergebnis dieser Bachelorarbeit ist die Ausarbeitung eines Konzeptes zur Digitalisierung des ärztlichen Rezeptes, um den bisherigen Prozess zu optimieren und die beteiligten Personengruppen zu entlasten. Die Transportwege zwischen Patient, Arzt, Apotheke und Rechenzentrum werden auf ein Minimum reduziert, die Arztpraxen können ihre Kapazitäten effektiver einsetzen und Patienten sind unabhängig von den Sprechstunden.

Nach aktuellem Stand der bisherigen Prozesse und verwendeten Technologien wurden die Anforderungen an das Konzept definiert und der Ablauf zwischen der digitalen Rezept-Anforderung und -Einlösung aufeinander abgestimmt. Dazu wurden Aufbau, Funktionen und technische Anforderungen an App, Web-Anwendung und Kassen-Anwendung festgelegt.

Ein weiteres Ziel ist die Realisierung der Rezept-Ausstellung in Form von prototypbasierten Software-Anwendungen. Dafür wurde zur optimalen Speicherung der Daten ein relationales Datenbankmodell konzipiert und eine Datenbank dementsprechend angelegt. Zusätzlich wurden die App und die Web-Anwendung mit einer kompatiblen Datenbankanbindung für den Austausch von Daten implementiert.

Für eine vollständige Verwendung bietet sich eine Anpassung des Prototyps an, um diesen um diverse Funktionalitäten zu erweitern. Für die Rezept-Ausstellung müssen Datenbank und Web-Anwendung auf öffentlichen Servern ausgeführt werden und die Datenbankanfragen über sichere Internet-Verbindungen laufen. Erweiterungen für die Profilbearbeitung und die Generierung des Rezept-Datensatzes für die Rezept-Einlösung müssen noch eingebunden werden. Zusätzlich muss für die Rezept-Einlösung die Kassen-Anwendung mit Anbindung an ein zuständiges Rechenzentrum implementiert werden. Um den großen Anteil der iPhone-Nutzer einzubinden, besteht die Möglichkeit eine iOS-Version der App zu erstellen.

Zur Qualitätssicherung sollten anschließend Usability Test mit verschiedenen Testgruppen durchgeführt werden, um Design, Aufbau und Funktionen der Anwendungen optimieren zu können. Erweiterungen und Anpassungen sollten jedoch kein Problem darstellen, weil bei der Implementierung austauschbare Elemente wie Fragmente und Komponenten verwendet wurden.

Die zukünftige Einführung des Digitalen Rezeptes wird die Digitalisierung im Gesundheitswesen voranbringen und das herkömmliche Rezept in Papierform ersetzen. Durch Gesetzesänderungen und technische Umrüstung der Arztpraxen innerhalb der letzten Jahre gibt es vom Bundesministerium für Gesundheit bereits konkrete Pläne für den Einsatz elektronischer Rezepte ab 2020.

Literaturverzeichnis

AOK Die Gesundheitskassen, 2016, Arzneimittel, https://aok-bv.de/lexikon/a/index_00197.html, letzter Zugriff am 11.01.2019.

Apache Friends, 2019, Was ist XAMPP?, <https://www.apachefriends.org/de/index.html>, letzter Zugriff am 26.01.2019

Bauer, Curt, 18.05.2018, NFC: Einfach erklärt, was hinter Near Field Communication steckt, https://praxistipps.chip.de/nfc-einfach-erklart-was-hinter-near-field-communication-steckt_12294, letzter Zugriff am 22.02.2019

Begerow, Markus, © 2010-2018, Relationales Datenbankmodell, <http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/relationales-datenbankmodell/>, letzter Zugriff am 28.01.2019

Bundesärztekammer, 2018, E-Health-Gesetz - neue Anwendungen für Ärzte und Versicherte kommen, <https://www.bundesaerztekammer.de/aerzte/telematiktelemedizin/earztausweis/e-health-gesetz/>, letzter Zugriff am 22.01.2019

Bundesärztekammer, 31.Dezember 2017, Ärztestatistik 2017, <https://www.bundesaerztekammer.de/ueber-uns/aerztestatistik/aerztestatistik-2017/>, letzter Zugriff am 18.01.2019

Bundesärztekammer, 31.Dezember 2016, Ärztestatistik 2016: Die Schere zwischen Behandlungsbedarf und Behandlungskapazitäten öffnet sich, <https://www.bundesaerztekammer.de/ueber-uns/aerztestatistik/aerztestatistik-2016/>, letzter Zugriff am 18.01.2019

Bundesinstitut für Arzneimittel und Medizinprodukte, 2013 und 2018, Betäubungsmittel, https://www.bfarm.de/DE/Bundesopiumstelle/Betaeubungsmittel/_node.html, letzter Zugriff am 11.01.2019.

Bundesinstitut für Arzneimittel und Medizinprodukte, 2013, Arzneimittelzulassung, https://www.bfarm.de/DE/Buerger/Arzneimittel/Arzneimittelzulassung/_node.html, letzter Zugriff am 11.01.2019

Bundesministerium der Justiz und für Verbraucherschutz, Gesetz über den Verkehr mit Arzneimitteln (Arzneimittelgesetz - AMG) § 44 Ausnahme von der Apothekenpflicht, https://www.gesetze-im-internet.de/amg_1976/_44.html, letzter Zugriff am 11.01.2019

Bundesministerium für Gesundheit, 17.Mai 2018, E-Health, <https://www.bundesgesundheitsministerium.de/service/begriffe-von-a-z/e/e-health.html>, letzter Zugriff am 22.01.2018

Bundesvereinigung Deutscher Apothekerverbände e. V., 2019, Internet und Versandhandel, <https://www.abda.de/themen/recht/apotheken-wettbewerb/versandhandel/>, letzter Zugriff am 06.03.2019

Codecademy, 2019, React Setup, Part 1: React and React DOM, Install React.js On Your Local Computer, <https://www.codecademy.com/articles/react-setup-i>, letzter Zugriff am 11.02.2019

Literaturverzeichnis

- Datenbanken-verstehen.de, 2010-2018, DML, DDL, DCL, SQL-Kommandos, <http://www.datenbanken-verstehen.de/datenbank-grundlagen/dbms/dml-ddl-dcl-kommandos/>, letzter Zugriff am 01.02.2019
- Deutsche Apotheker Zeitung 2019, 12.01.2003, Apotheken-EDV - besser als ihr Ruf (EDV-Umfrage, Teil 1), <https://www.deutsche-apotheker-zeitung.de/daz-az/2003/daz-3-2003/uid-9024>, letzter Zugriff am 20.02.2019
- Deutsche Apotheker Zeitung DAZ.online, 10.08.2018, ABDA legt Plan für eigenes E-Rezept vor, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/08/10/abda-legt-plan-fuer-eigenes-e-rezept-vor>, letzter Zugriff am 23.01.2019
- Deutsche Apotheker Zeitung DAZ.online, 10.08.2018, ABDA legt Plan für eigenes E-Rezept vor, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/08/10/abda-legt-plan-fuer-eigenes-e-rezept-vor/chapter:2>, letzter Zugriff am 23.01.2019
- Deutsche Apotheker Zeitung DAZ.online, 13.11.2018, Spahn: Fernverordnungsverbot fällt, E-Rezept bis 2020, <https://www.deutsche-apotheker-zeitung.de/news/artikel/2018/11/13/spahn-fernverordnungsverbot-faellt-e-rezept-bis-2020/chapter:1>, letzter Zugriff am 23.01.2019
- DeutscheApothekenPortal, 2019, LANR (Lebenslange Arztnummer), <https://www.deutsche-sapotheckenportal.de/rezept-retax/dap-lexikon/lanr-lebenslange-arztnummer/>, letzter Zugriff am 06.01.2019
- DeutschesApothekenPortal, 2019, Betriebsstättennummer (BSNR), <https://www.deutsche-sapotheckenportal.de/rezept-retax/dap-lexikon/betriebsstaettennummer-bsnr/>, letzter Zugriff am 06.01.2019
- Die Techniker, 2018, Welche Rezepte gibt es?, <https://www.tk.de/techniker/service/leistungen-und-mitgliedschaft/leistungen/arzneimittel/arzneimittel-auf-rezept/rezepte-rezept-unterschiede-2002740>, letzter Zugriff am 11.01.2019.
- Ditzel, Peter, 27.08.2015, Wie ein Rezept zu Geld wird, <https://www.deutsche-apotheker-zeitung.de/daz-az/2015/daz-35-2015/wie-ein-rezept-zu-geld-wird>, letzter Zugriff am 07.01.2019
- DocMorris N.V., 2018, Rezept einlösen, <https://www.docmorris.de/rezepteinloesen?navnode=redeemPrescription>, letzter Zugriff am 06.03.2019
- DocMorris, <https://www.docmorris.de/>, letzter Zugriff am 22.01.2019
- Doctolib, 2019, <https://info.doctolib.de/?origin=homepage-website-desktop>, letzter Zugriff am 22.01.2019
- EHI Retail Institute, 2018, Heute und zukünftig im Handel auf den Kassen eingesetzte Betriebssysteme im Jahr 2018 (Basis: Anzahl Installationen der befragten Unternehmen), <https://www.handelsdaten.de/deutschsprachiger-einzelhandel/im-handel-auf-den-kassen-eingesetzte-betriebssysteme-2018>, letzter Zugriff am 20.02.2019
- Facebook Inc., 2018, Tutorial: Intro to React, <https://reactjs.org/tutorial/tutorial.html#what-is-react>, letzter Zugriff am 11.02.2019
- Facebook Inc., 2019, ReactDOM, <https://reactjs.org/docs/react-dom.html#render>, letzter Zugriff am 04.03.2019

Literaturverzeichnis

- GKV-Spitzenverband, 2018, Elektronische Gesundheitskarte (eGK), https://www.gkv-spitzenverband.de/krankenversicherung/telematik_und_datenaustausch/egk/egk.jsp, letzter Zugriff am 22.01.2019
- Google Developers, 2019, AsyncTask, <https://developer.android.com/reference/android/os/AsyncTask>, letzter Zugriff am 09.02.2019
- Google Developers, 2019, Connect to the network, <https://developer.android.com/training/basics/network-ops/connecting>, letzter Zugriff am 09.02.2019
- Google Developers, 2019, Manifest.permission, <https://developer.android.com/reference/android/Manifest.permission>, letzter Zugriff am 09.02.2019
- Google Developers, 23. Januar 2019, Fragments, <https://developer.android.com/guide/components/fragments>, letzter Zugriff am 07.02.2019
- Google Developers, 23. Januar 2019, Understand the Activity Lifecycle, <https://developer.android.com/guide/components/activities/activity-lifecycle>, letzter Zugriff am 07.02.2019
- Horchert, Judith, 19.09.2018, Neue Gesundheits-App Vivy, <http://www.spiegel.de/netzwelt/apps/vivy-sicherheitsexperte-kritisiert-von-krankenkassen-angebotene-gesundheits-app-a-1228749.html>, letzter Zugriff am 22.01.2019
- Karlstetter, Florian, 20.04.2018, Was ist JSON?, <https://www.cloudcomputing-insider.de/was-ist-json-a-704909/>, letzter Zugriff am 29.01.2019
- Kassenärztliche Bundesvereinigung, 2018, Anbindung an die Telematikinfrastruktur, <https://www.kzbv.de/anbindung-an-die-telematikinfrastruktur.1163.de.html>, letzter Zugriff am 22.01.2019
- Kassenärztliche Bundesvereinigung, 2019, Arzneimittel-Verordnung, <https://www.kbv.de/html/27760.php>, letzter Zugriff am 06.01.2019
- Kösterke, Jessy, 18.08.2017, Was sind eigentlich relationale Datenbanken? Das Prinzip der Normalisierung, <https://t3n.de/news/eigentlich-relationale-datenbanken-683688/2/>, letzter Zugriff am 28.01.2019
- Kösterke, Jessy, 18.08.2017, Was sind eigentlich relationale Datenbanken?, <https://t3n.de/news/eigentlich-relationale-datenbanken-683688/>, letzter Zugriff am 26.01.2019
- Kotlin/JETBRAINS, 2019, Using Kotlin for Android Development, <https://kotlin-lang.org/docs/reference/android-overview.html>, letzter Zugriff am 07.02.2019
- Lingner, Margrit, 29.08.2014, QR-Code selber erstellen - So geht's, <https://www.pc-magazin.de/ratgeber/qr-code-erstellen-selber-machen-how-to-2525646.html>, letzter Zugriff am 22.02.2019
- Litzel, Nico, 08.06.2017, Was ist ein Datenbankmanagementsystem?, <https://www.bigdata-insider.de/was-ist-ein-datenbankmanagementsystem-a-615034/>, letzter Zugriff am 26.01.2019
- Mozilla, 2005-2019, Cross-Origin Resource Sharing (CORS) <https://developer.mozilla.org/de/docs/Web/HTTP/CORS>, letzter Zugriff am 01.02.2019

- Offizielles Gesundheitsportal der deutschen ApothekerInnen, 2019, Rezepte haben verschiedene Farben und Bedeutungen, <https://www.aponet.de/wissen/arzneimitteldatenbank/arzneimittel/rezeptarten.html>, letzter Zugriff am 11.01.2019
- Rödiger, Jana, 07.12.2017, HTTPS UND SSL VERSCHLÜSSELUNG - WAS IST DAS UND WARUM IST DAS FÜR WEBSEITEN UNVERZICHTBAR?, <https://www.media-company.eu/blog/allgemein/https-und-ssl-verschluesSELUNG/>, letzter Zugriff am 22.02.2019
- Rouse, Margaret, 2002-2019, Digitale Signatur (elektronische Signatur), <https://www.se-archsecurity.de/definition/Digitale-Signatur-elektronische-Signatur>, letzter Zugriff am 22.02.2019
- Schürmann, Tim, 27.September 2017, Kotlin ist auch nur eine Insel, <https://www.golem.de/news/programmiersprache-fuer-android-kotlin-ist-auch-nur-eine-insel-1709-130224.html>, letzter Zugriff am 07.02.2019
- statista, 2019, Anzahl der Mitglieder und Versicherten der gesetzlichen und privaten Krankenversicherung in den Jahren 2012 bis 2018 (in Millionen), <https://de.statista.com/statistik/daten/studie/155823/umfrage/gkv-pkv-mitglieder-und-versicherten-zahl-im-vergleich/>, letzter Zugriff am 11.01.2019
- Stiftung Warentest, 01.01.2018, Zulassung, <https://www.test.de/medikamente/begriffe/zulassung/>, letzter Zugriff am 11.01.2019.
- Stiftung Warentest, 2018, Arzneimittel - Definition, <https://www.test.de/medikamente/begriffe/>, letzter Zugriff am 11.01.2019
- Tagesschau, Wenn "Vivy" ans Impfen erinnert, 17.09.2018, <https://www.tagesschau.de/inland/gesundheits-app-101.html>, letzter Zugriff am 22.01.2019
- Tarif-Testsieger, 2019, Kassenrezept: Gültigkeit und Zuzahlung beachten, <https://www.tarif-testsieger.de/gesetzliche-krankenversicherung/kassenrezept/>, letzter Zugriff am 11.01.2019
- Tec GmbH & Co. KG, 2019, RFID-Technologie, <https://www.smart-tec.com/de/auto-id-welt/rfid-technologie>, letzter Zugriff am 22.02.2019
- The PHP Group, 2001-2019, mysqli_result::fetch_all, <http://php.net/manual/de/mysqli-result.fetch-all.php>, letzter Zugriff am 01.02.2019
- The PHP Group. 2001-2019, mysqli_query, <http://php.net/manual/de/mysqli.query.php>, letzter Zugriff am 01.02.2019
- Tillmann, Janna, 22.09.2017, Was ist eigentlich ein (virtuelles) DOM?, <https://t3n.de/news/eigentlich-virtuelles-dom-858160/>, letzter Zugriff am 11.02.2019
- vitabook GmbH, 2019, Folgerezept, <https://www.vitabook.de/gesundheitslexikon/folge-rezept.php>, letzter Zugriff am 07.01.2019
- w3schools.com ,1999-2019, JSON PHP, https://www.w3schools.com/js/js_json_php.asp, letzter Zugriff am 01.02.2019