



# Entwicklung eines webbasierten Patientenmoduls für eine internationale medizinische Datenbank

Bachelorarbeit an der Universität Ulm

**Vorgelegt von:**

Dennis Rodriguez  
dennis.rodriguez@uni-ulm.de

**Gutachter:**

Prof. Dr. Manfred Reichert

**Betreuer:**

Michael Stach

2018

Fassung 12. Oktober 2018

© 2018 Dennis Rodriguez

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF- $\LaTeX$  2 $\epsilon$

## **Kurzfassung**

Obwohl digitale Informations- und Kommunikationstechnologien mehr und mehr eingesetzt werden, werden für Umfragen und Studien immernoch fast ausschließlich Fragebögen auf Papier verwendet. Dies verursacht zum einen sehr viel Arbeit bei der Validierung der ausgefüllten Fragebögen und bei der späteren Digitalisierung. Außerdem wird dadurch die Anzahl der Patienten, von welchen Daten erhoben werden können, auf jene, die Kliniken besuchen eingeschränkt. Für viele Krankheiten, wie beispielsweise Tinnitus, ist es auf Grund des sehr hohen Grades an Heterogenität der Krankheit jedoch sehr wichtig, Daten von möglichst vielen Betroffenen aus unterschiedlichen Gruppierungen zu erhalten.

Diese Arbeit soll ein System entwickeln, welches es Patienten ermöglicht Daten über ein webfähiges Endgerät einzugeben. Dies ermöglicht mehr Flexibilität für die Patienten, da die Fragebögen auch zu Hause mittels eines Smartphones oder Tablets ausgefüllt werden können. Außerdem wird die manuelle Validierungs- und Digitalisierungsarbeit überflüssig, da die Daten direkt vom Gerät des Benutzers in die Datenbank gespeichert werden können.



## Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich beim Erstellen dieser Arbeit unterstützt haben.

Besonderer Dank gilt Herr *Prof. Dr. Manfred Reichert* für die Begutachtung dieser Arbeit und die Ermöglichung der Durchführung am *Institut für Datenbanken und Informationssysteme* an der Universität Ulm.

Ebenso möchte ich mich bei *Michael Stach* bedanken, für die engagierte Betreuung, nicht nur bei der Erstellung dieser Arbeit, sondern auch beim Entwurf und der Implementierung des Patientenmoduls während der oftmals auch recht spontanen Treffen.

Abschließend möchte ich mich bei Freunden und Familie bedanken für ihre Unterstützung, sowohl während der Erstellung dieser Arbeit als auch während den restlichen Studienjahren.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Problemstellung . . . . .	2
1.3	Struktur der Arbeit . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Tinnitus . . . . .	5
2.2	TRI . . . . .	6
2.2.1	Tinnitus Database . . . . .	6
2.2.2	CRF . . . . .	7
2.3	AES . . . . .	10
2.3.1	SubBytes . . . . .	12
2.3.2	ShiftRows . . . . .	12
2.3.3	MixColumns . . . . .	13
2.3.4	AddRoundKey . . . . .	13
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>15</b>
3.1	TrackYourTinnitus . . . . .	15
3.2	QuestionSys . . . . .	16
<b>4</b>	<b>Anforderungen</b>	<b>17</b>
4.1	Benutzerprofilanalyse . . . . .	17
4.2	Randbedingungen . . . . .	18
4.3	Funktionale Anforderungen . . . . .	18
4.4	Nicht-funktionale Anforderungen . . . . .	20
<b>5</b>	<b>Architektur</b>	<b>23</b>
5.1	Übersicht . . . . .	23
5.2	Modulbeschreibungen . . . . .	25
5.2.1	Ablauf . . . . .	25

*Inhaltsverzeichnis*

5.2.2	Architektur . . . . .	29
<b>6</b>	<b>Ausgewählte Aspekte der Implementierung</b>	<b>33</b>
6.1	Technologien . . . . .	33
6.2	Auszüge aus der Implementierung . . . . .	34
6.2.1	SurveyJS . . . . .	34
6.2.2	Zwischenspeicherung im Patientenmodul . . . . .	36
<b>7</b>	<b>Anforderungsabgleich</b>	<b>41</b>
7.1	Funktionale Anforderungen . . . . .	41
7.2	Nicht-funktionale Anforderungen . . . . .	42
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>43</b>
8.1	Zusammenfassung . . . . .	43
8.2	Ausblick . . . . .	43

# 1

## Einleitung

Weltweit leiden rund 15% der Bevölkerung an Tinnitus. Die meisten der Betroffenen befinden sich dabei im Alter zwischen 40 und 80 Jahren [1]. Dabei sind die Auswirkungen für jeden betroffenen Patienten unterschiedlich. Manche beschreiben es lediglich als störend, während andere durch den Tinnitus davon abgehalten werden, kognitiv anspruchsvolle Arbeit ausüben zu können. Bei allen Betroffenen jedoch ist die Lebensqualität durch den Tinnitus vermindert. [2]

Die Behandlung des Tinnitus gestaltet sich jedoch häufig als schwierig, da sich das Krankheitsbild bei jedem Patienten anders präsentiert. Dazu kommt eine Vielzahl von unterschiedlichen Auslösern und unterschiedliches Ansprechen auf verschiedene Behandlungsmethoden. Daraus folgt, dass es keine standardisierten Behandlungsmethoden gibt, was oft zu einem Gefühl der Hilflosigkeit bei Ärzten und Patienten führen kann. [3]

Zur Therapiefindung füllen die Patienten Fragebögen aus, nach denen sie dann in verschiedenen Formen und Schweregraden des Tinnitus eingeteilt werden, um dann entsprechende Behandlung zu erhalten.

Um diesen Prozess zu unterstützen, wurde die "Tinnitus Database" [4] entwickelt. Dort werden die Ergebnisse der Fragebögen digitalisiert und den Ärzten übersichtlich zur Verfügung gestellt. Somit wird das Einteilen der Patienten in die verschiedenen Tinnitus Formen und Schweregrade erleichtert.

## 1.1 Motivation

Auch wenn die Tinnitus Database eine große Hilfestellung bei der Behandlung und Untersuchung des Tinnitus leistet, ist ihre Handhabung doch sehr umständlich. Das Ausfüllen der Fragebögen erfolgt nur auf Papier, wodurch jeder Fragebogen zunächst noch digitalisiert werden muss, bevor er im System verwendet werden kann. Dieser Prozess ist sehr aufwändig, erfordert jedoch kein medizinisches Fachwissen. Um diese Arbeit zu erleichtern und den Patienten gleichzeitig mehr Flexibilität beim Ausfüllen der Fragebögen zu bieten, soll das Patientenmodul entwickelt werden.

## 1.2 Problemstellung

Ziel dieser Arbeit ist ein System zu entwickeln, welches die in Kapitel 1.1 beschriebene aufwändige Arbeit der manuellen Digitalisierung überflüssig macht. Hierfür soll das System einerseits die Möglichkeit bieten, Fragebögen direkt in digitaler Form auszufüllen, und die Antworten direkt in der Tinnitus Datenbank speichern zu können. Andererseits bietet das System den Patienten die Möglichkeit die Fragebögen bereits zuhause auszufüllen, was den Vorteil bietet, die Aufenthaltszeit in der Klinik während den regelmäßigen Visiten zu verringern.

Bei der Entwicklung eines solchen Systems gilt jedoch zu beachten, dass ein Großteil, der an Tinnitus leidenden Bevölkerung, sich bereits schon in gehobenem Alter befindet. Es muss also davon ausgegangen werden, dass die meisten der Benutzer des Systems nicht sehr geübt sind im Umgang mit digitalen Systemen. Folglich muss das System sehr intuitiv gestaltet sein und verhindern, dass die Benutzer unbeabsichtigte Funktionen ausführen können. Das gleiche gilt für den Teil des Systems, welcher in die Tinnitus Database integriert ist und von den Ärzten bedient wird. Hierbei handelt es sich ebenfalls um mögliche Laien im Umgang mit IT Systemen, klare Rückmeldungen und intuitive Bedienung sind also wichtig.

## 1.3 Struktur der Arbeit

In den folgenden sieben Kapiteln dieser Arbeit werden die grundlegenden Aspekte sowie allgemeine und spezifische Implementierungsaspekte eines solchen Patientenmoduls dargestellt. Kapitel 2 beschäftigt sich zunächst noch einmal mit den Grundlagen des Tinnitus sowie der grundlegenden Struktur der Tinnitus Database sowie den Fragebögen. Kapitel 3 gibt einen Einblick in bereits bestehende Systeme, welche ähnliche Ziele wie das Patientenmodul verfolgen. In Kapitel 4 werden die Randbedingungen sowie funktionale und nicht-funktionale Anforderungen erläutert. Kapitel 5 bietet eine Beschreibung der zugrundeliegenden Architektur des Patientenmoduls. Dies beinhaltet sowohl eine Übersicht als auch Beschreibungen der einzelnen Module sowie den Ablauf im System. Kapitel 6 gibt einen Überblick über die verwendeten Technologien und gibt konkrete Implementierungsbeispiele einiger Komponenten. In Kapitel 7 erfolgt ein Abgleich der ursprünglich gestellten Anforderungen mit der tatsächlichen Funktionalität des System, gefolgt von einer Zusammenfassung und möglichen Erweiterungen des Systems in Kapitel 8.



# 2

## Grundlagen

In diesem Kapitel werden zunächst Grundlagen erläutert. Dies beinhaltet eine Erläuterung einiger Aspekte des Tinnitus sowie eine kurze Beschreibung der Tinnitus Research Initiative und der Tinnitus Database, für welche das Patientenmodul eine Erweiterung darstellt.

### 2.1 Tinnitus

Betrachtet man Tinnitus im Ganzen, muss man zunächst die zwei grundlegenden Arten des Tinnitus, subjektiv und objektiv, unterscheiden. Bei objektivem Tinnitus hört die betroffene Person Geräusche, die prinzipiell überall im Körper entstehen und auf unterschiedlichsten Wegen das Ohr erreichen können. Dabei kann es sich zum Beispiel um unregelmäßigen Blutfluss oder Muskelkontraktionen handeln. Anders als beim subjektiven Tinnitus können die Geräusche, die einen objektiven Tinnitus verursachen, auch von anderen Personen wahrgenommen werden. Meist liegt dem objektiven Tinnitus eine andere Krankheit zu Grunde, welche die entsprechenden Geräusche verursacht. Die überwältigende Mehrheit der Betroffenen leidet jedoch an subjektivem Tinnitus, welcher durch das Fehlen einer tatsächlichen Krankheit schwerer zu behandeln ist. Subjektiver Tinnitus kann sehr viele verschiedene Formen annehmen: von sehr hochfrequenten Tönen, nicht unähnlich dem Geräusch von Grillen, bis hin zu sehr niedrigen Frequenzen. Ebenso gibt es Unterschiede in der Persistenz, einige Patienten leiden durchgehend am Tinnitus, während er bei anderen lediglich in Phasen auftritt. [2]

Über die Jahre hat die Forschung außerdem gezeigt, dass subjektiver Tinnitus nicht im Ohr entsteht, sondern durch abnormale neurale Aktivität im Gehirn verursacht wird. Die

## 2 Grundlagen

genaue neurale Aktivität, welche den Tinnitus verursacht, ist nicht im Detail bekannt, jedoch haben Studien gezeigt, dass sich die Aktivität von durch Geräuschstimulation hervorgerufene Aktivität unterscheidet, was nahelegt, dass die verschiedenen Formen den Tinnitus auf unterschiedliche Arten erzeugt werden. [5]

Es gibt Hinweise darauf, dass sich chronischer Tinnitus grundlegend von akutem Tinnitus unterscheidet. Diese Abhängigkeit von der Dauer des Tinnitus spielt eine wichtige Rolle in der Behandlung der Patienten. Außerdem gibt es Hinweise, dass die Wirksamkeit von Behandlungen nachlässt, wenn der Tinnitus länger als 5 Jahre anhält. [5]

Dies alles führt dazu, dass um eine effektive Behandlung von Tinnitus zu erreichen, die Therapie an jeden Patienten individuell angepasst werden muss. Dies erfordert wieder viel Forschungsarbeit, um die verschiedenen Klassifizierungsmerkmale leichter erkennen zu können und die Patienten schneller den richtigen Therapien zuweisen zu können. Hierfür wurde die Tinnitus Research Initiative, TRI [6], gegründet.

## 2.2 TRI

Die Tinnitus Research Initiative ist eine non-profit Organisation mit dem Ziel, effektive Behandlungen für alle Arten des Tinnitus zu entwickeln. In der Vergangenheit war die Forschung im Bereich Tinnitus stark von einzelnen Individuen oder kleinen Gruppen aus einer Vielzahl von unterschiedlichen Disziplinen abhängig. Die Forschung wurde meist individuell und unabhängig von einander durchgeführt. Dies soll mit Hilfe der Tinnitus Research Initiative geändert werden, um ein besseres Verständnis des Tinnitus zu erreichen.

### 2.2.1 Tinnitus Database

Teil der Tinnitus Research Initiative ist die Tinnitus Database. Sie dient dem Speichern der Patientendaten und Auswertung der Fragebögen.

Die Funktionalität umfasst alles vom Erstellen der Patienten im System bis hin zur Validation der ausgefüllten Fragebögen. Wurde der Patient im System angelegt und die entsprechenden

Visiten erstellt, erhält man eine Übersicht über die zugehörigen Fragebögen (vgl. Abbildung 2.1).

The screenshot displays the Tinnitus Database web application. On the left is a dark sidebar with navigation options like 'Home', 'Patients', 'Screening', 'Questionnaires', and 'Reports'. The main content area is divided into two sections. The top section, titled 'Record', shows 'Patient Data' with fields for 'Entered Patient ID', 'Consent Database ID', 'Date of Birth', 'Sex', 'Smoking Status', 'Case', and 'Case ID'. Below this is a 'Screening' section with 'Visit Date' and 'Position' fields, and corresponding 'Created at' and 'Updated at' timestamps. The bottom section is titled 'Tinnitus Sample Case History Questionnaire' and contains several numbered questions with radio button and dropdown options. Questions include: 1. State of Birth, 2. Gender, 3. Handedness, 4. Family history of tinnitus complaints, 5. Initial onset: When did you first experience your tinnitus?, 6. How did you perceive the beginning?, 7. Was the initial onset of your tinnitus related to..., 8. Have your tinnitus levels in PULSATED?, 9. Where do you perceive your tinnitus?, and 10. How does your tinnitus manifest itself over time?.

Abbildung 2.1: Tinnitus Datenbank: Übersicht Patientensite [4]

Zusätzlich existieren noch Funktionen, welche bestimmte Fragebögen auswerten und als Metriken oder graphisch darstellen.

## 2.2.2 CRF

Das *Case Report Form* [7] ist ein Erhebungsbogen, in dem Untersuchungsdaten eines Patienten, in normalerweise anonymisierter Form, festgehalten werden. In Abbildung 2.2 ist eine Übersicht über alle Fragebögen, welche Patienten der TRI zu den verschiedenen Zeitpunkten ausfüllen müssen, dargestellt. Dabei erfüllt jeder Fragebogen einen spezifischen Zweck. Neue Patienten füllen zunächst Fragebögen aus, die die Krankheitsgeschichte des Patienten abdecken. Im späteren Verlauf der Studie zielen die Fragebögen dann auf den Krankheitsverlauf und eventuelle Änderungen der Symptome ab. Dazu kommen Fragebögen, die den Depressiongrad des Patienten evaluieren sollen, sowie Fragebögen, die die allgemeine Lebensqualität des Patienten abfragen.

## 2 Grundlagen

	Screening*	Baseline*	Week 2	Week	Week 12 = End** (Vend)
<b>TSCHQ</b>	A				
<b>Otological Examination</b>	A				
<b>Medical history (no CRF form)</b>	A				
<b>Audiometry</b>	A	A			B
<b>Loudness match</b>		B			B
<b>Pitch match</b>		B			B
<b>Maskability</b>		B			B
<b>Residual Inhibition</b>		B			B
<b>THI</b>	A	A	A	A	A
<b>TBF 12</b>	B	B	B	B	B
<b>Tinnitus Severity</b>	A	A	A	A	A
<b>MDI</b>	A	A	A	A	A
<b>WHOQOL</b>	B	B	B	B	B
<b>CGI</b>			A	A	A
<b>Concomittant medication</b>	A	A	A	A	A
<b>Adverse Events</b>			A	A	A
<b>Comorbidity</b>	A	A	A	A	A

### Abbreviations:

A	Essential
B	Highly recommended
TSCHQ	Case History Questionnaire
THI	Tinnitus Handicap Inventory (Newman et al 1996)
TBF 12	Tinnitus Impairment Questionnaire
CGI	Clinical Global Impression
MDI	Major Depression Inventory
WHOQOL	World Health Organization – Quality of Life (Questionnaire)
BREF	Best available technique reference document

Abbildung 2.2: Übersicht über die verschiedenen Fragebögen der CRF [7]

In Abbildung 2.3 sehen wir einige Beispielfragen aus dem TSCHQ (Tinnitus Sample Case History Questionnaire) Fragebogen. Wie zu erkennen ist, ist ein Großteil der Fragen einfach gehalten und als Multiple-Choice-Fragen oder Abstufungen durch Zahlenbereiche modelliert.

<p><b>17. Wieviel Prozent der Zeit im letzten Monat haben Sie sich über Ihren Tinnitus geärgert bzw. waren Sie wegen des Tinnitus unglücklich oder genervt?</b></p> <p><u>  </u> <u>  </u> <u>  </u> <u>  </u> % (Bitte tragen Sie eine Zahl zwischen 1 und 100 ein.)</p>
<p><b>18. Wievielen verschiedenen Behandlungen haben Sie sich aufgrund Ihres Tinnitus unterzogen ?</b></p> <p><input type="checkbox"/> keiner    <input type="checkbox"/> einer    <input type="checkbox"/> 2 - 4    <input type="checkbox"/> 5 und mehr</p>
<p><b>19. Wird die Lautstärke Ihres Tinnitus durch bestimmte Arten von Umgebungsgeräuschen reduziert bzw. überdeckt, wie zum Beispiel durch das Rauschen eines Wasserfalls oder das Geräusch fließenden Wassers, wenn Sie unter der Dusche stehen ?</b></p> <p><input type="checkbox"/> JA    <input type="checkbox"/> NEIN    <input type="checkbox"/> Ich weiß es nicht</p>
<p><b>20. Verschlechtert starker Lärm Ihren Tinnitus?</b></p> <p><input type="checkbox"/> JA    <input type="checkbox"/> NEIN    <input type="checkbox"/> I weiß es nicht</p>
<p><b>21. Beeinflusst eine Bewegung Ihres Kopfes und/oder Ihres Nackens (z.B. das Verschieben des Kiefers oder das Zusammenbeißen der Zähne) oder eine Berührung Ihrer Arme, Hände oder Ihres Kopfes Ihren Tinnitus ?</b></p> <p><input type="checkbox"/> JA    <input type="checkbox"/> NEIN</p>

Abbildung 2.3: Auszug aus dem TSCHQ Fragebogen [7]

## 2.3 AES

Im Folgenden wird ein Blick auf das gewählte Verschlüsselungsverfahren AES geworfen. Es wird erläutert, wie die Verschlüsselung mittels AES [8] funktioniert und was die Vorteile gegenüber Vorgängerverfahren sind.

AES ging aus einem 5 jährigem Auswahlprozess des *National Institute of Standards and Technology* hervor, bei welchem ein Nachfolger für das in die Jahre gekommenen DES Verfahren gesucht wurde. Ein kurzer Vergleich, zu sehen in Abbildung 2.4, der beiden Verfahren zeigt schnell wieso AES das überlegene Verfahren ist.

	DES	AES
Date	1976	1999
Block size	64	128
Key length	56	128, 192, 256
Number of rounds	16	9,11,13
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but accept open public comment
Source	IBM, enhanced by NSA	Independent cryptographers

Abbildung 2.4: Vergleich von AES und DES [9].

Durch die Erhöhung der *Block Size* und *Key length* ist es mit der Rechenleistung aktueller Systeme nicht möglich in brauchbarer Zeit die Verschlüsselung mittels Brute Force Angriffen zu brechen. Ebenfalls ist kein mathematischer Ansatz bekannt, welcher den verwendeten Schlüssel berechnen kann. Somit bietet AES aktuell, solange keine mathematischer Ansatz zur Berechnung des Schlüssels gefunden wird, maximale Sicherheit bis der Zeitpunkt erreicht ist, an welchem durch die ständige Zunahme an erreichbarer Rechenleistung Brute Force

Angriffe möglich machen wird.

Zusätzlich zu den aus Abbildung 2.4 hervorgehenden Verbesserungen in Bezug auf die Sicherheit, welche sich aus der größeren *Block size* und *Key length* ergeben, benötigt AES außerdem weniger Rechenleistung und ist dabei gleichzeitig schneller in der Berechnung als die konkurrierenden Verfahren [9].

Wie in Abbildung 2.5 zu sehen, ist AES in einzelne "Runden" unterteilt. Die Anzahl der Durchläufe hängt dabei von Größe des verwendeten Schlüssels (Cipher Key) ab.

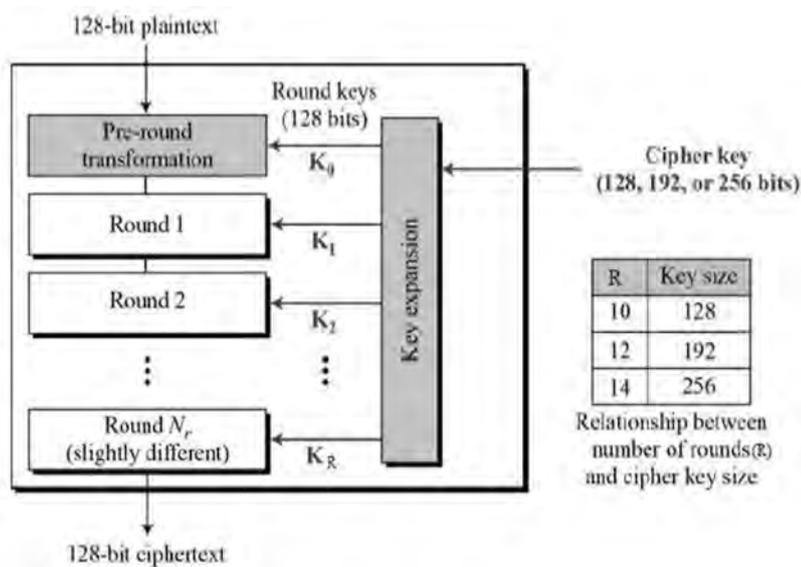


Abbildung 2.5: Übersicht über den Ablauf von AES [10].

Die einzelnen "Runden" des Verfahrens sind wiederum, wie in Abbildung 2.6 zu sehen, in 4 Schritte unterteilt. Dabei werden alle in AES verwendeten Operation auf Bytes ausgeführt. AES behandelt die 128 Bit der Eingabe folglich als 16 Byte Blocks.

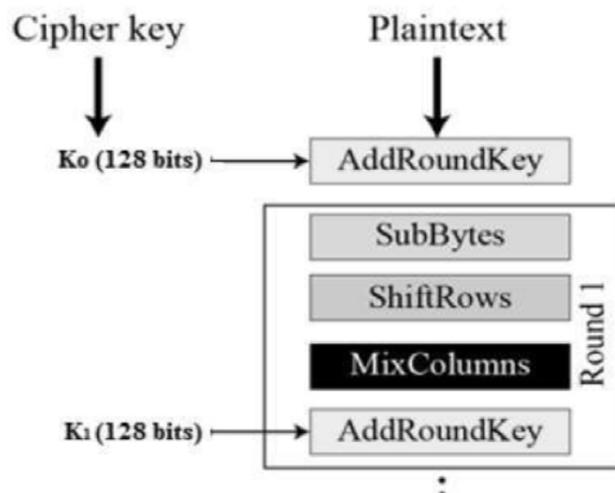


Abbildung 2.6: Darstellung einer "Runde"des AES Verfahrens [10].

Im Folgenden wird nun kurz auf die einzelnen Schritte eingegangen, um die grundlegende Funktionsweise von AES zu erläutern.

### 2.3.1 SubBytes

Beim ersten Schritt jeder Runde handelt es sich um eine Substitution von Bytes mit Hilfe einer vorgegebenen Substitutionstabelle (genannt S-box). Das Ergebnis hierbei bildet eine 4x4 Matrix. [8]

### 2.3.2 ShiftRows

Schritt 2 verschiebt alle Reihen der in Schritt 1 entstandenen Matrix nach links. Die Anzahl der Verschiebungen unterscheidet sich dabei pro Reihe. Die erste Reihe bleibt unberührt, die zweite Reihe wird um eine Byteposition verschoben, die dritte um zwei Positionen und die vierte Reihe um drei Positionen. Folglich erhält man anschließend erneut eine 4x4 Matrix. [8]

### 2.3.3 MixColumns

In Schritt 3 werden die einzelnen Spalten der Matrix ersetzt, indem die ursprünglichen Spalten als Polynom interpretiert werden und anschließend mit einem fest vorgegebenen weiteren Polynom multipliziert werden [8]. Dabei gilt zu beachten, dass dieser Schritt im letzten Durchlauf nicht mehr ausgeführt wird.

### 2.3.4 AddRoundKey

Beim abschließenden Schritt wird lediglich der aktuelle *Round Key* auf die aus Schritt 3 (bzw. Schritt 2 im letzten Durchlauf) hervorgehende Matrix mit Hilfe einer simplen bitweisen XOR-Funktion angewendet. [8]



# 3

## Verwandte Arbeiten

An dieser Stelle werden zu dieser Arbeit existierende vergleichbare Arbeiten und Projekte betrachtet. Dabei wird zunächst das Projekt TrackYourTinnitus betrachtet, welches zum Ziel hat, das Problem der Datenerhebung bei chronischen Erkrankungen zu lösen, indem den Patienten mittels einer Smartphone App eine kontinuierliche Datenerhebung ermöglicht wird. Zusätzlich wird das Projekt QuestionSys vorgestellt, welches dem Benutzer den Entwurf von Fragebögen ermöglicht, um diese dann später auf mobilen Geräten ausfüllen zu können. Außerdem dient das System als Middleware zum sicheren Datenaustausch und Datenmanagement.

### 3.1 TrackYourTinnitus

Mit Hilfe des Projekts TrackYourTinnitus wird eines der Hauptprobleme bei der Behandlung von Tinnitus behoben. Wie in Kapitel 2.1 beschrieben unterscheidet sich das Ansprechverhalten von Patient zu Patient. Dies erfordert eine sehr genaue Erfassung des Tinnitus sowie möglicher Einflussfaktoren. Die aktuell eingesetzten Verfahren zur Datenerhebung betrachten jedoch nicht temporäre Variabilitäten oder Umweltfaktoren, welche die Intensität des Tinnitus beeinflussen können [11].

Ein weiteres Problem ergibt sich, wenn man betrachtet, welche Patientengruppe zu den erhobenen Daten beiträgt. Im Bezug auf den Tinnitus wurden in der Vergangenheit mehrere Datenbanken [12], [13] erstellt, um beim Verstehen der Heterogenität des Tinnitus zu helfen. Diese Datenbanken beziehen ihre Daten jedoch von Patienten, welche eine Klinik aufsuchen um bei Studien mitzuwirken. [11]

An dieser Stelle setzt das TrackYourTinnitus Projekt an, indem mittels der Möglichkeit Pa-

### 3 Verwandte Arbeiten

tientendaten auch außerhalb von Kliniken erheben zu können, Daten aus verschiedenen Patientengruppen erfasst werden können. Die mobile Anwendung präsentiert dem Benutzer hierfür 12 mal am Tag einen Fragebogen. Dies soll gewährleisten, dass aus möglichst vielen verschiedenen Alltagssituationen Daten erhoben werden, um alle möglichen Einflussfaktoren zu erfassen. Zusätzlich kann das System weitere Messungen nehmen, wie Geräuschlautstärke im Umfeld des Benutzers.

## 3.2 QuestionSys

Die weitverbreitete Nutzung von mobilen Geräten zur Datenerhebung, zum Beispiel in klinischen Studien oder Umfragen, bietet einige vielversprechende Perspektiven [14]. Hinzu kommen Systeme, welche im Vergleich zur klassischen Studie, Daten im Alltag der Patienten erfassen, wie das in Kapitel 3.1 erwähnte Projekt TrackYourTinnitus oder Chrodis+ [15] für Diabetes Patienten.

Für solche Systeme bietet QuestionSys ein Framework zur Erstellung, Validierung, Analyse und Verwendung von Fragebögen. Dabei ist das System in drei Teile gegliedert. Der Fragebogen Konfigurator ermöglicht das Erstellen der Fragen, sowie das Zusammenfassen der einzelnen Fragen zu Seiten innerhalb des Fragebogens, welche dann auf dem Endgerät dargestellt werden. Die Fragebogen App erlaubt es den Benutzern die Fragebögen auf ihren mobilen Geräten auszufüllen und sammelt somit die Benutzerdaten.

Der dritte Teil stellt eine Middleware dar, um sicheren Datenaustausch und Datenmanagement gewährleisten zu können. Ein wichtiges Element von QuestionSys ist, dass es Benutzern ohne Programmierkenntnisse ermöglichen soll Anwendungen zur Datenerhebung zu entwerfen [16].

# 4

## Anforderungen

Im folgenden Kapitel werden die Anforderungen an das System genauer betrachtet. Dies beinhaltet eine Benutzerprofilanalyse sowie die Randbedingungen und Anforderungen dieser Arbeit.

### 4.1 Benutzerprofilanalyse

Rund 15% der Weltbevölkerung leiden an Tinnitus, die Mehrheit davon im Alter zwischen 40 und 80 Jahren. Der Großteil der Teilnehmer an klinischen Studien zu Tinnitus leidet an chronischem Tinnitus, welcher die höchste Prävalenz im Alter zwischen 60 und 69 erreicht [1]. Das bedeutet, dass die größte Altersgruppe unter den Benutzern des Patientenmoduls bei ungefähr 60 Jahren beginnt. Dies bedeutet, dass der größte Teil der Benutzer des Patientenmoduls nur sehr wenig, oder womöglich auch gar keine Erfahrung im Umgang mit digitalen Systemen hat. Die zweite Benutzergruppe umfasst die Ärzte, welche ebenfalls das System bedienen. Es muss also davon ausgegangen werden, dass es sich bei den Benutzern dieser zweiten Gruppe nicht um Experten im Umgang mit IT-Systemen handelt.

Dies bedeutet, dass das System so gestaltet sein muss, dass eine möglichst intuitive Bedienung gewährleistet werden kann, welche außerdem dem Benutzer kaum bis gar keine Möglichkeiten bietet das System falsch zu bedienen.

## 4.2 Randbedingungen

Beim Entwurf des Systems gab es einige Randbedingungen zu beachten, welche das Design des Patientenmoduls beeinflusst haben. Da Teile des Systems in die bereits bestehende Tinnitus Database integriert werden mussten, wurde bei der Entwicklung des Patientenmoduls auf das in der Tinnitus Database verwendete Laravel Framework zurückgegriffen. Dies bietet maximale Kompatibilität zwischen den Systemem und ermöglicht einfachen Datenaustausch.

## 4.3 Funktionale Anforderungen

Im Folgenden werden die sich aus den vorangegangenen Kapiteln ergebenden funktionalen Anforderungen nochmals im Detail beschrieben. Dies beinhaltet sowohl Anforderungen die sich speziell auf die in Kapitel 4.1 beschriebenen Benutzergruppen beziehen, als auch Anforderungen welche sich aus der Benutzung des Systems im Umfeld der Medizin ergeben.

Nummer	Titel	Priorität	Seite
FA01	Fortschrittsanzeige	verpflichtend	15
FA02	Ablauf des Fragebogens	verpflichtend	15
FA03	Abgabe eines Fragebogens	verpflichtend	15
FA04	Worklist	verpflichtend	15
FA05	Reihenfolge der Fragebögen	verpflichtend	15
FA06	Datenschutz	verpflichtend	15
FA07	Nutzerfeedback	optional	15
FA08	Zeitmessung	optional	16
FA09	Sprache	verpflichtend	16
FA10	Rückmeldung	verpflichtend	16
FA11	Zwischenspeicherung	verpflichtend	16

Tabelle 4.1: Funktionale Anforderungen

### **FA01 (Fortschrittsanzeige):**

Der Benutzer soll permanent Feedback über seinen Fortschritt bekommen. Dazu sollte er zu jeder Zeit sehen können wie sein Fortschritt aussieht. Dabei sollte unterschieden werden zwischen dem Fortschritt auf der aktuellen Seite als auch innerhalb des Fragebogens.

### **FA02 (Ablauf des Fragebogens):**

Die Bearbeitung des Fragebogens soll einen strikten Prozess darstellen. Der Benutzer sollte zwar die Möglichkeit besitzen innerhalb eines Fragebogens vorherige Seiten nochmals anzuschauen, jedoch sollte er nicht dazu ermutigt werden. Wurde ein Fragebogen abgegeben, darf der Benutzer nicht die Möglichkeit besitzen seine Angaben nochmals zu bearbeiten.

### **FA03 (Abgabe eines Fragebogens):**

Erreicht ein Benutzer das Ende eines Fragebogens, soll vor dem Abschicken keine Zusammenfassung mehr angezeigt werden, um wie bereits in FA2 beschrieben den Benutzer nicht dazu zu ermutigen seine Angaben nochmals zu ändern.

### **FA04 (Worklist):**

Die einzelnen Fragebögen sollen für den Nutzer keinen spezifischen Zeitpunkten zugeordnet werden. Es soll lediglich eine einfache Worklist angezeigt werden, die alle Fragebögen enthält, die der Nutzer noch bearbeiten muss.

### **FA05 (Reihenfolge der Fragebögen):**

Falls nötig soll es möglich sein die Reihenfolge, in der die Patienten die Fragebögen bearbeiten, abändern zu können.

### **FA06 (Datenschutz):**

Die Benutzer müssen ihr Einverständnis geben, bevor die Daten im Patientenmodul zwischengespeichert werden können.

#### 4 Anforderungen

##### **FA07 (Nutzerfeedback):**

Jeder Nutzer soll einmalig einen Feedback-Fragebogen bekommen. Dieser sollte sowohl spezifische Fragen, als auch Freitext anbieten und dient dazu das System an die Nutzer anpassen zu können.

##### **FA08 (Zeitmessung):**

Das System soll die Zeit messen, die der Benutzer braucht, um einen Fragebogen auszufüllen und ebenfalls abspeichern, ob Pausen eingelegt werden (wenn der Nutzer den Fragebogen über mehrere Tage verteilt ausfüllt). So kann bestimmt werden, wie zuverlässig die Angaben der Nutzer sind.

##### **FA09 (Sprache):**

Da das System international verwendet wird, müssen mehrere Sprachen für die Oberfläche und Fragebögen zur Verfügung stehen.

##### **FA10 (Rückmeldung):**

Das System soll dem Benutzer Rückmeldung geben wenn Aktionen erfolgreich ausgeführt wurden, bzw. im Falle von auftretenden Fehlern dem Benutzer leicht verständliche Meldungen anzeigen, wodurch der Fehler verursacht wurde.

##### **FA11 (Zwischenspeicherung):**

Das System soll dem Benutzer die Möglichkeit bieten noch nicht vollständig ausgefüllte Fragebögen zwischenspeichern.

## **4.4 Nicht-funktionale Anforderungen**

Wie bei den funktionalen Anforderungen sind auch die nicht-funktionalen Anforderungen durch die Benutzerprofilanalyse geprägt. Hinzu kommen wiederum noch Anforderungen, welche sich aus der Verwendung des Systems im Kontext der Medizin ergeben.

Nummer	Titel	Seite
NA01	Usability	17
NA02	Fehlerprävention	17
NA03	Wartbarkeit	17
NA04	Robustheit	17
NA05	Effizienz	17
NA06	Datenschutz	18

Tabelle 4.2: Nicht-funktionale Anforderungen

##### **NA01 (Usability):**

Da das System von Benutzern aller Altersgruppen verwendet wird, muss auf eine einfache und intuitive Bedienbarkeit geachtet werden. Dies schließt auch Rückmeldungen über den Zustand des Systems mit ein, wie in FA1 beschrieben.

##### **NA02 (Fehlerprävention):**

Da das System auch von Personen höheren Alters bedient wird (vgl. NA01), die mit dem Umgang von digitalen Systemen unter Umständen nur wenig vertraut sind, muss das System im Stande sein, Fehler von vornherein auszuschließen. Für den Fall dass doch ein Fehler auftritt, muss das System in der Lage sein diesen ohne Zutun des Nutzer selbst beheben zu können.

##### **NA03 (Wartbarkeit):**

Wie in FA5 bereits beschrieben sollten die Fragebögen im System änderbar sein. Dazu sollten diese nicht fest im System eingebettet sein, sondern über eine Datenbank ins System geladen werden können.

##### **NA04 (Robustheit):**

Es dürfen keine Fehler während der Benutzung des Systems auftreten, die dazu führen, dass der Patient einen oder mehrere Fragebögen erneut ausfüllen muss, da dies die Qualität der gegebenen Antworten beeinflussen kann.

#### *4 Anforderungen*

##### **NA05 (Effizient):**

Das System soll effizient und performant sein, um gleichbleibende Bedienqualität unabhängig von der Anzahl der Benutzer zu gewährleisten.

##### **NA06 (Datenschutz):**

Benutzer im System sollen vollständig anonym bleiben. Das System soll so wenig Benutzerdaten wie möglich speichern.

# 5

## Architektur

In diesem Kapitel wird der Aufbau des Patientenmoduls genauer betrachtet. Dabei gilt zunächst zu beachten, dass das System in zwei Module aufgeteilt ist. Wie in Abbildung 5.1 zu sehen, ist ein Teil des Patientenmoduls in die Tinnitus Datenbank integriert.

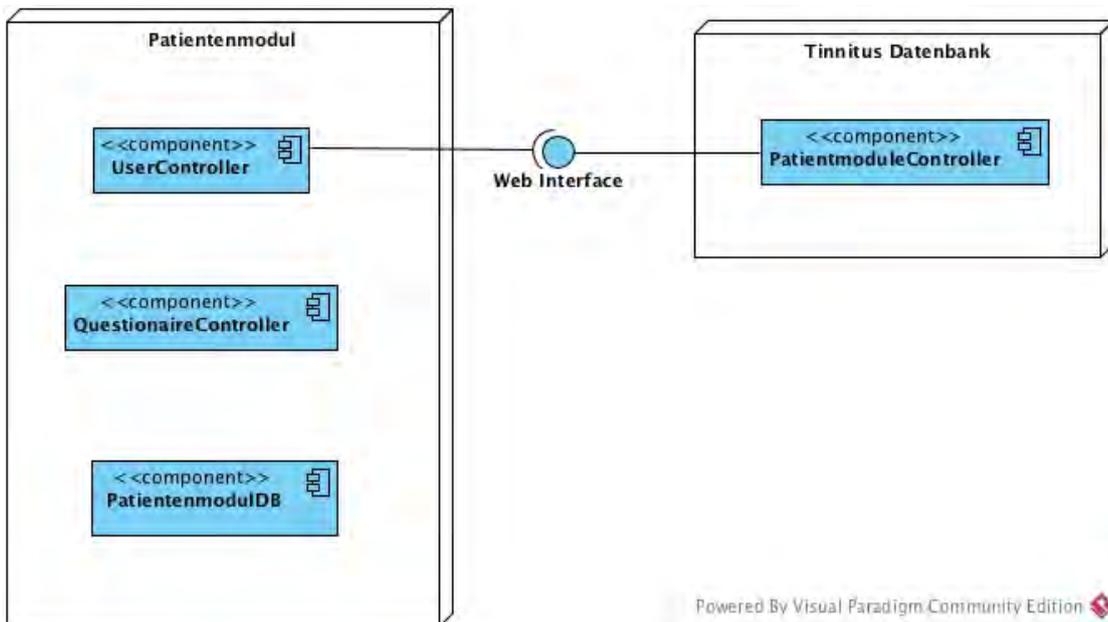


Abbildung 5.1: Übersicht: Patientenmodul

### 5.1 Übersicht

Der Teil des Systems, der von den Ärzten bedient wird, ist in die Tinnitus Database integriert. Über einen Button in der Navigationsleiste der Tinitus Database wird der Benutzer zu einer

## 5 Architektur

Übersicht (siehe Abbildung 5.2) geleitet, welche alle für den Patienten angelegten Visiten anzeigt, sowie welche Fragebögen für den Patienten ausgewählt wurden und welche er davon bereits ausgefüllt hat. Zusätzlich besteht die Möglichkeit über die Übersichtsseite einer Visite (vgl. Abbildung 2.1) zur Übersicht über die Fragebögen zu gelangen.

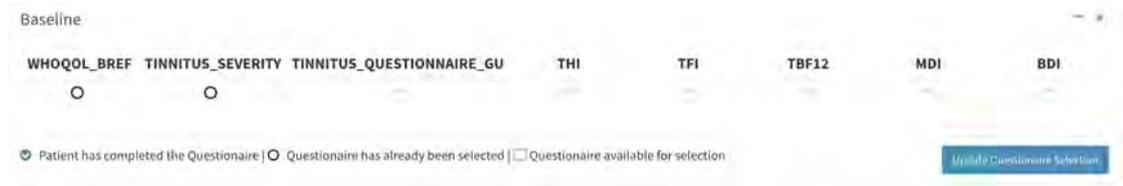


Abbildung 5.2: Auswahlmaske der auszufüllenden Fragebögen

Möchte der Benutzer die Reihenfolge in der der Patient die Fragebögen ausfüllt ändern, kann er dies über die Auswahl wie in Abbildung 5.3 gezeigt tun.

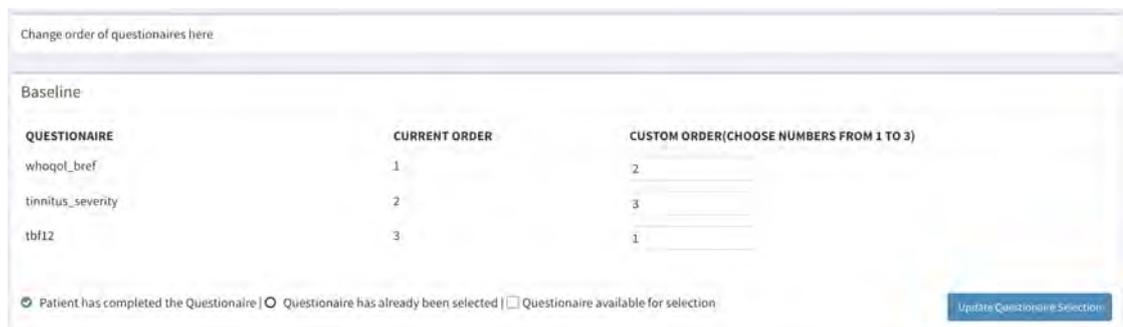


Abbildung 5.3: Anpassung der Reihenfolge der auszufüllenden Frageböge.

Der Teil des System, welcher von den Patienten bedient wird, ist sehr simpel gehalten, um so wenig Möglichkeiten zur falschen Bedienung wie möglich zu bieten. Der Patient loggt sich, wie in Abbildung 5.4 zu sehen, mit seinem Token ein. Dieses Token erhält er von seinem Arzt und gilt immer nur für eine Visite. Dies ermöglicht es, die Zuordnung von Patient und Token lediglich in der Tinnitus Datenbank abbilden zu müssen und sorgt somit für die vollständige Anonymität der Patienten innerhalb des Patientenmoduls.

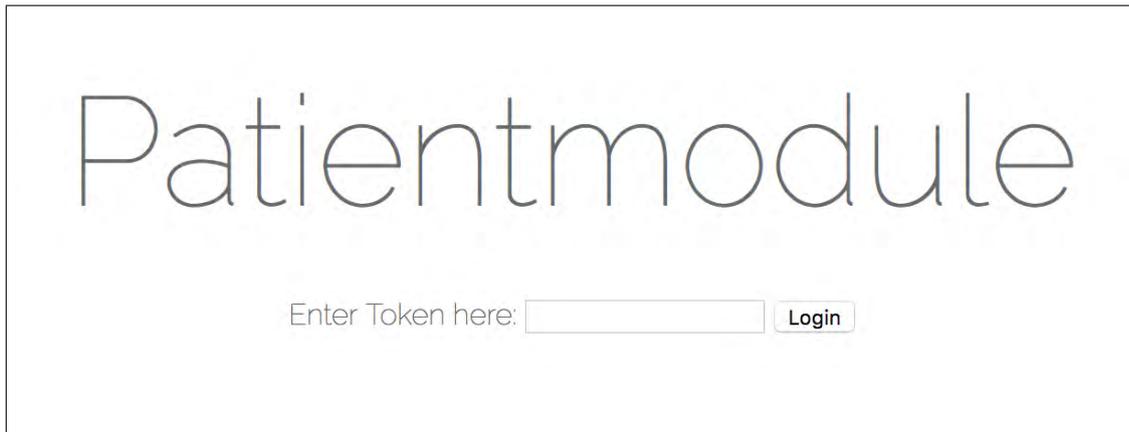


Abbildung 5.4: Login im Patientenmodul mit Token.

Nach dem Login erhält der Patient lediglich eine Übersicht, welche ihm anzeigt, wie viele Fragebögen für die nächste Visite noch auszufüllen sind. Außerdem wird dem Patient die Möglichkeit geboten, die Zwischenspeicherung der Fragebogendaten freizuschalten. Dies ermöglicht dem Patienten Fragebögen nur teilweise auszufüllen und später, auch an einem anderen webfähigen Gerät, fortzusetzen. Dabei werden sämtliche Daten vor dem Speichern mit einem vom Patienten gewählten Passwort verschlüsselt.

## 5.2 Modulbeschreibungen

Im Folgenden erfolgt eine genauere Betrachtung der einzelnen Bestandteile des Patientenmoduls sowie eine vollständige Darstellung der Abläufe im System.

### 5.2.1 Ablauf

Im Folgenden wird einmal der vollständige Ablauf, vom Auswählen der Fragebögen bis hin zur schlussendlichen Abspeicherung der Antworten, erläutert.

Sobald ein Patient in der Tinnitus Datenbank angelegt und wenigstens eine Visite für diesen Patienten erstellt wurde, stehen die Funktionalitäten des Patientenmoduls für diesen Patienten zur Verfügung. Über die Unterseite "Patientenmodul" der Tinnitus Database (vgl. Abbildung

## 5 Architektur

5.5) erhält der Benutzer einen Überblick über alle Visiten des ausgewählten Patienten und kann gleichzeitig von hier aus die Auswahl der auszufüllenden Fragebögen bearbeiten.

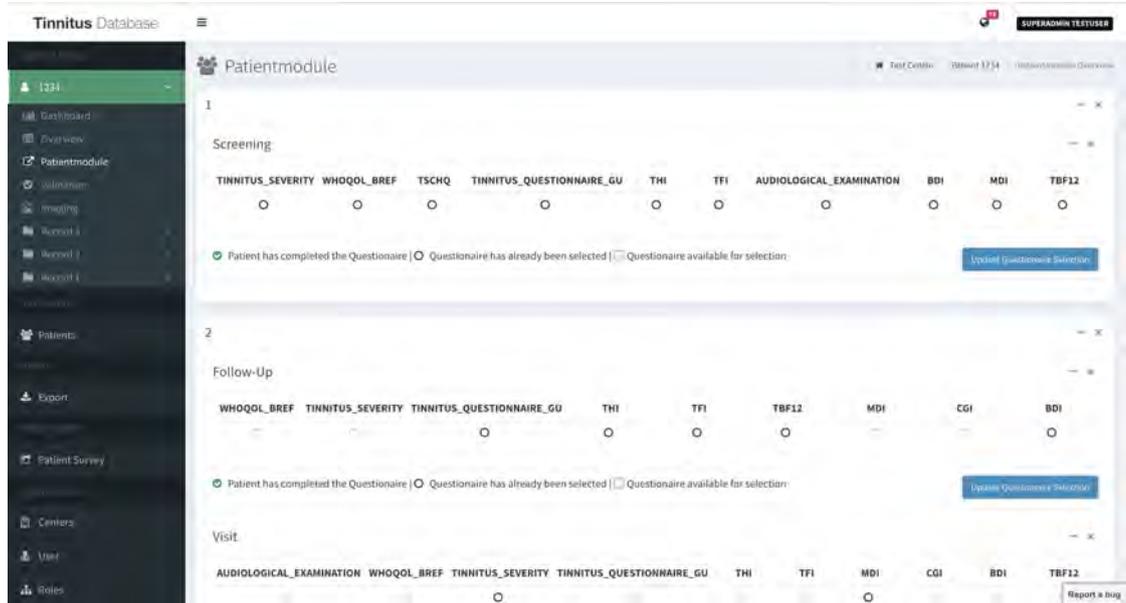


Abbildung 5.5: Übersicht über alle Visiten des Patienten.

Hat der Benutzer die gewünschten Fragebögen ausgewählt, kann er im nächsten Schritt (siehe Abbildung 5.3), falls gewünscht, noch die Reihenfolge, in welcher die Fragebögen dem Patient vorgelegt werden, anpassen. Falls erforderlich kann diese Auswahl jederzeit nachträglich noch erweitert werden. Hierbei gilt jedoch zu beachten, dass für den speziellen Fall indem der Patient alle zuvor ausgewählten Fragebögen bereits ausgefüllt hat, der Patient außerhalb des Systems darauf hingewiesen werden muss, dass neue Fragebögen hinzugefügt wurden. Da im Patientenmodul keinerlei Patientendaten hinterlegt sind, ist es nicht möglich diese Benachrichtigung zu automatisieren.

Der Ablauf des System bis zu diesem Zeitpunkt wird in Abbildung 5.6 als Sequenzdiagramm dargestellt.

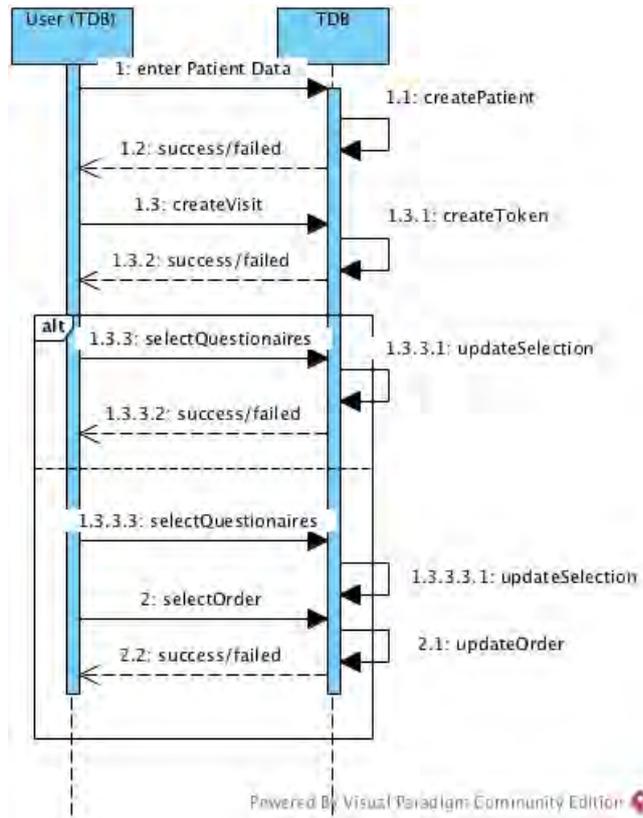


Abbildung 5.6: Sequenzdiagramm zum Ablauf der Funktionen in der Tinnitus Database.

Bevor der Patient beginnen kann, muss ihm jedoch noch das für die Visite entsprechende Token mitgeteilt werden. Dieses wird automatisch beim Anlegen der Visite in der Tinnitus Datenbank generiert und auf der Übersichtsseite angezeigt.

Ab diesem Zeitpunkt kann sich der Patient im eigentlichen Patientenmodul mit seinem Token einloggen und mit dem Ausfüllen der Fragebögen beginnen. Nach dem Login (vgl. Abbildung 5.4) wird der Patient auf eine persönliche Übersichtsseite geleitet. Hier sieht er zum einen wie viele Fragebögen noch ausstehen bis zur nächsten Visite (vgl. Abbildung 5.7).

## 5 Architektur

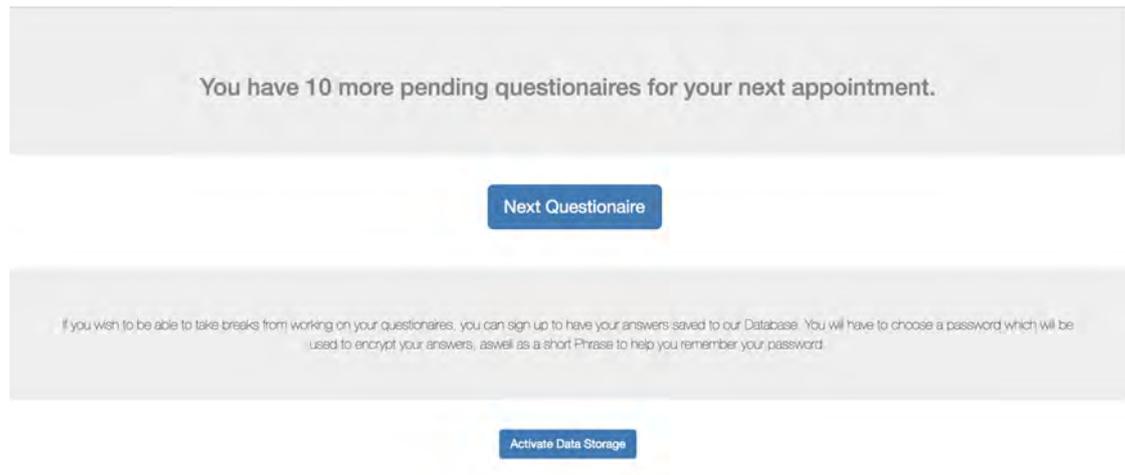


Abbildung 5.7: Startseite des Benutzers

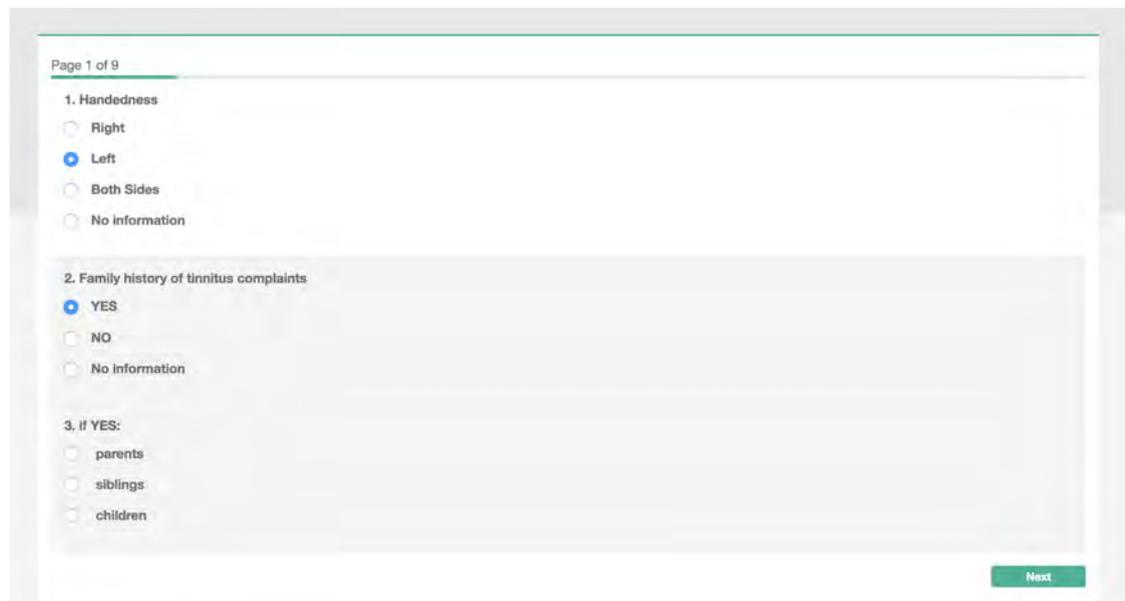


Abbildung 5.8: Beispiel für die Darstellung mittels SurveyJS anhand des TSCHQ Fragebogens.

Außerdem besteht hier die Möglichkeit der verschlüsselten Zwischenspeicherung der nicht vollständig ausgefüllten Fragebögen zuzustimmen, um das Ausfüllen pausieren zu können. Nachdem ein Fragebogen vollständig ausgefüllt wurde, wird der Benutzer automatisch zu

dieser Seite zurückgeleitet. Außerdem wird bei jedem Aufruf dieser Übersicht eine Anfrage an die Tinnitus Datenbank geschickt um zu prüfen, ob neue Fragebögen der Auswahl hinzugefügt wurden. Möchte der Benutzer den nächsten Fragebogen ausfüllen genügt ein Klick auf den Button *Next Questionnaire* und das System wählt den entsprechenden Fragebogen aus, welcher mittels SurveyJS dem Benutzer dargestellt wird (vgl. Abbildung 5.8).

Der Ablauf für den Benutzer des Patientenmoduls wird nochmals als Sequenzdiagramm in Abbildung 5.9 dargestellt.

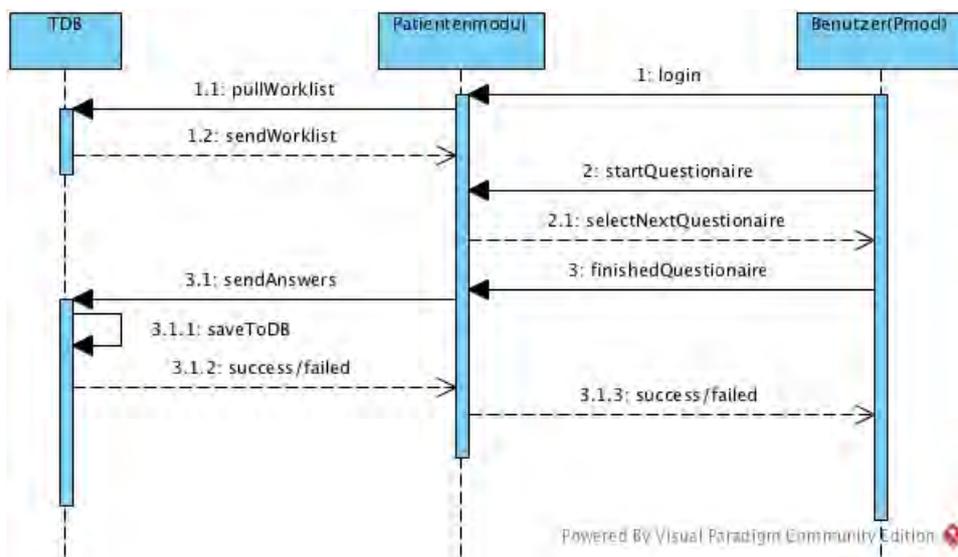


Abbildung 5.9: Übersicht über Ablauf im Patientenmodul.

## 5.2.2 Architektur

Um das Patientenmodul eventuell auch mit anderen Datenbanken verwenden zu können, wurde der Aufbau relativ simpel gehalten. Die Tinnitus Database wurde lediglich um einen `PatientModuleController` und zwei Relationen in der Datenbank erweitert. Dies reicht bereits, um sämtliche benötigten Funktionen implementieren zu können.

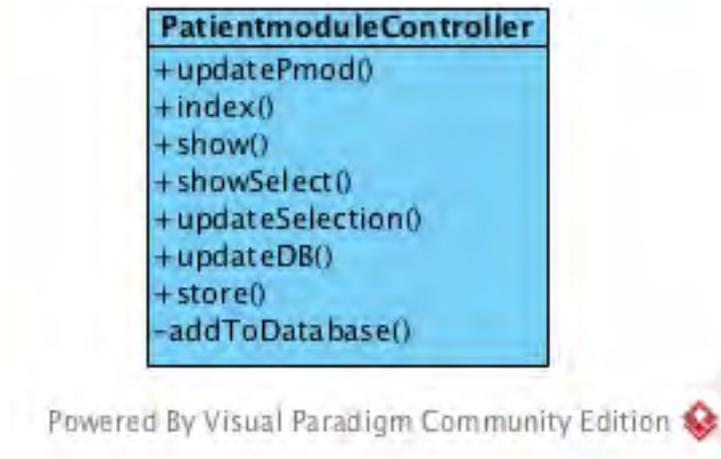


Abbildung 5.10: Klassendiagramm TDB-Modul des Patientenmoduls

Die in Abbildung 5.10 zu sehenden Klassen `Pmod_survey` und `Pmod_worklist` sind Teil der Struktur des verwendeten Frameworks Laravel [17] und werden für den in Laravel verwendeten Database Query Builder benötigt.

Für den Teil des Patientenmoduls, welcher von den Patienten bedient wird, wurde ebenfalls auf das Laravel Framework zurückgegriffen. Dies hat einen sehr ähnlichen strukturellen Aufbau wie in der Tinnitus Datenbank zur Folge.

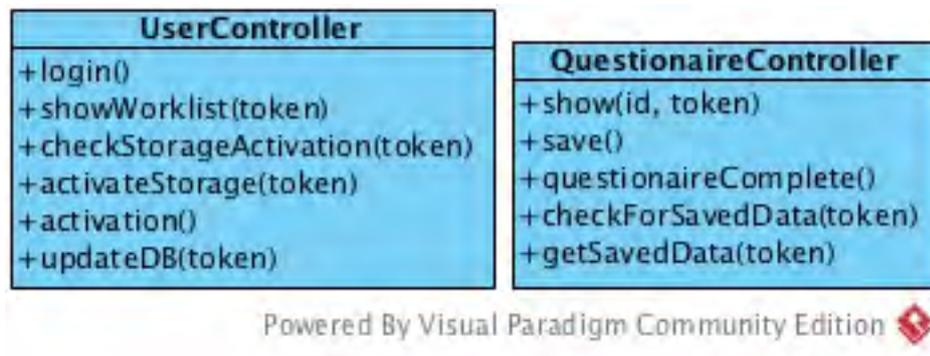


Abbildung 5.11: Klassendiagramm Patientenmodul

Wie in Abbildung 5.11 zu sehen besitzt das Patientenmodul zwei Controller, dabei verwaltet der `UserController` das Aktualisieren der Fragebögen sowie die Weiterleitung der An-

torwrten an die Tinnitus Datenbank. Der `QuestionnaireController` dient lediglich der Weiterleitung der in JSON formatierten Fragebögen an JavaScript, um dann mit Hilfe des SurveyJS Frameworks [18] die Fragebögen zu generieren.

Um mittels SurveyJS die Fragebögen generieren zu können, müssen diese als JSON-formatierte Stringvariable in JavaScript vorliegen. Um keine Datenbankaufrufe in JavaScript ausführen zu können, wurde eine Erweiterung des Laravel Framework verwendet, welche es ermöglicht, serverseitige Daten in Javascript verfügbar zu machen.

Die genaue Generierung der Fragebögen mittels SurveyJS wird in Kapitel 5.2.1 erläutert.



# 6

## Ausgewählte Aspekte der Implementierung

Dieses Kapitel betrachtet zunächst die verwendeten Technologien bei der Erstellung des Patientenmoduls sowie die Beweggründe, welche zur Auswahl der entsprechenden Technologien geführt haben. Anschließend werden zwei der Kernelemente des Systems im Detail betrachtet.

### 6.1 Technologien

#### **PHP [19] / Laravel [17] :**

Als Programmiersprache für diese Bachelorarbeit wurde PHP gewählt. Der ausschlaggebende Punkt für diese Entscheidung war, dass Teile des Patientenmoduls in die Tinnitus Datenbank integriert werden sollten, welche mit dem Laravel Framework realisiert wurde, welches wiederum ein PHP Framework ist.

#### **Nginx [20] :**

Nginx wurde als Server zum Hosten der Webseiten verwendet. Die Entscheidung für Nginx gegenüber Apache wurde getroffen, da diese Bachelorarbeit hauptsächlich auf einem MacOS Rechner erstellt wurde. Für Nutzer von MacOS bietet Laravel den sogenannten Laravel Valet Dienst an, welcher sehr einfaches Hosten von Webseiten ermöglicht, was schlussendlich sehr viel Konfigurationsarbeit erleichtert hat.

#### **SQL [21] :**

Als Datenbankmodell wurde, wie in der Tinnitus Database, auf eine relationale Datenbank zurückgegriffen. Zu Beginn des Projekts wurde zunächst die dokumentbasierte MongoDB

## 6 Ausgewählte Aspekte der Implementierung

Datenbank verwendet. Aufgrund der Tatsache, dass alle Fragebögen als JSON-formatierter String gespeichert werden müssen, war der Gedanke hinter der Verwendung von MongoDB eine leichtere Modifikation der JSON Dateien ohne aufwändige Datenbankoperationen ausführen zu müssen. Im Laufe des Projekts wurde jedoch klar, dass dies gar nicht nötig ist, da die Fragebögen alle standardisiert sind und sich nur äußerst selten ändern. Da Laravel außerdem keine native Unterstützung für MongoDB bietet und diese nur über die Verwendung von Community Packages realisiert werden kann, wurde schlussendlich die Entscheidung getroffen, eine relationale Datenbank zu verwenden.

### **SurveyJS [18] :**

Für die Modellierung der Fragebögen wurde SurveyJS verwendet. SurveyJS bietet die Möglichkeit, die Fragebögen mit Hilfe des Builder Tools [22] zu erstellen. Anschließend werden nur ein paar Zeilen JavaScript Code benötigt, um den Fragebogen im Patientenmodul darzustellen. Die leichte Handhabung und generelle Unkompliziertheit von SurveyJS hat dazu geführt, dass SurveyJS den Vorzug vor händischer Erstellung der Fragebögen in HTML erhält. Sollte ein Fragebogen doch einmal abgeändert werden müssen, geht dies ebenfalls sehr einfach. Man kann ganz einfach den JSON String in den Builder kopieren und mit ein paar Klicks die Reihenfolge von Fragen tauschen, oder Antwortoptionen abändern und erhält sofort den aktualisierten JSON String.

## **6.2 Auszüge aus der Implementierung**

Im Folgenden werden einige ausgewählte Teilbereiche des System genauer betrachtet. Bei den ausgewählten Bereichen handelt es sich um die wichtigsten Kernfunktionen des Patientenmoduls.

### **6.2.1 SurveyJS**

Um mittels SurveyJS einen Fragebogen darzustellen, muss der entsprechende JSON String vorhanden sein. Dafür sind alle Fragebögen in der Datenbank des Patientenmoduls als JSON String hinterlegt. Dieser muss nun lediglich in der JavaScript Datei verfügbar gemacht

werden. Um keine Datenbankaufrufe in JavaScript ausführen zu müssen, wurde ein Laravel Package [23] verwendet, welches es ermöglicht, Variablen global in JavaScript verfügbar zu machen.

```
1 JavaScript::put([
2   'json' => $questionaire->questionaire,
3   'token' => $token,
4   'qId' => $id
5 ]);
```

Listing 6.1: Weiterleitung des JSON Strings für SurveyJS

Somit kann anschließend mittels `Patientmodule.json` auf den JSON String zugegriffen werden. Um anschließend den Fragebogen zu generieren, werden nur wenige Zeilen mehr benötigt, wie im nächsten Codebeispiel zu sehen ist.

In Listing 6.2 in Zeile 1 wird dem Survey Model der JSON String zugewiesen und in Zeile 20 anschließend in den HTML Code eingefügt. Dies reicht bereits aus um den Fragebogen vollständig darzustellen.

Um die eingegebenen Antworten am Ende auch weiter verarbeiten zu können, müssen jedoch noch ein paar Funktionen definiert werden. Die in Listing 6.2 Zeilen 6-14 definieren eine Funktion, welche nach beenden des Fragebogens aufgerufen wird. Hierbei handelt es sich lediglich um das Absenden eines HTTP Post Requests, um die Antworten an die Tinnitus Datenbank zu senden. Die in Listing 6.2 in den Zeilen 15-19 definierte Funktion wird ebenfalls nach Beenden des Fragebogens aufgerufen und sorgt dafür, dass der Benutzer anschließend automatisch wieder auf die Übersichtsseite geleitet wird.

## 6 Ausgewählte Aspekte der Implementierung

```
1 //laden des JSON Strings ins SurveyJS Model
2 window.survey = new Survey.Model(Patientmodule.json);
3 //von SurveyJS bereitgestellte Funktion, wird aufgerufen bei beenden des
  Fragebogens
4 survey.onComplete.add(function (result){
5   var xhttp = new XMLHttpRequest();
6   var body = Patientmodule.token.concat('::', Patientmodule.qId);
7   xhttp.onreadystatechange = function() {
8     if (this.readyState == 4 && this.status == 200) {
9       //redirect zur Startseite
10      window.location.assign(xhttp.responseText);
11    }
12  };
13  xhttp.open('POST', 'https://patientmodule.devlocal/questionnaireComplete',
14            true);
14  xhttp.send(body);
15 });
16 $("#surveyElement").Survey({model: survey});
```

Listing 6.2: JavaScript Code für SurveyJS Funktionalität

### 6.2.2 Zwischenspeicherung im Patientenmodul

Eine der an das System gestellten Anforderungen beinhaltet die Möglichkeit, nicht vollständig ausgefüllte Fragebögen im Patientenmodul zwischenspeichern, sodass der Benutzer den Fragebogen zu einem späteren Zeitpunkt, unter Umständen auch von einem anderen Gerät aus, fertig ausfüllen kann. Hierfür wird dem Benutzer auf der Startseite des Systems die Möglichkeit geboten, diese Speicherung zu aktivieren (vgl. Abbildung 5.7). Über den dargestellten Button gelangt der Benutzer zur Seite der eigentlichen Aktivierung (vgl. Abbildung 6.1). Dort wird dem Benutzer ebenfalls erläutert wie die Speicherfunktion zu bedienen ist.

To save your answers in the system you need to choose a password which will be used to encrypt your answers prior to saving them. Your password will not be stored in the system. To help you remember your password you need to enter a phrase or question which you associate with your chosen password. This phrase or question then will be displayed once you need to enter your password again.

**Enter your phrase/question here:**

Abbildung 6.1: Aktivierung der Speicherfunktion.

Hat der Benutzer seinen Merksatz eingegeben, wird er zur Startseite zurückgeleitet. An dieser Stelle ergibt sich nun ein Problem, welches aus der streng gehaltenen Anonymität der Benutzer im Patientenmodul hervorgeht. Da im Patientenmodul nur der Merksatz, nicht jedoch das Passwort, gespeichert wird, gibt es keine Möglichkeit bereits verschlüsselte Daten dem Benutzer wieder verfügbar zu machen, sollte dieser sein Passwort vergessen haben. Der gewählte Merksatz soll an dieser Stelle eine Hilfestellung bieten, jedoch kann ein schlecht gewählter Merksatz, welcher zu einem späteren Zeitpunkt keinen eindeutigen Hinweis mehr darstellt, zum Verlust des Passwortes und damit der verschlüsselten Daten führen. Der Ablauf der Speicherfunktion aus Sicht des Benutzer ist in Abbildung 6.2 als Flussdiagramm dargestellt.



Abbildung 6.2: Ablauf der Speicherfunktion aus sicht des Benutzer.

## 6 Ausgewählte Aspekte der Implementierung

Sollte der Benutzer der Speicherung zugestimmt haben, indem er seinen Merksatz eingegeben hat, werden bei jedem Seitenwechsel innerhalb eines Fragebogens alle bis dahin eingegebenen Antworten verschlüsselt und anschließend in der Datenbank des Patientenmoduls abgespeichert. Zum Verschlüsseln der Antworten wird das in Kapitel 2.3 beschriebene AES Verfahren verwendet. Zur Implementierung im Patientenmodul wurde auf die Library CryptoJS für JavaScript zurückgegriffen (vgl. Listing 6.3).

```
1 survey
2   .onCurrentPageChanged
3   .add(function () {
4     if (Patientmodule.storage == true) {
5       //AES encryption mit CryptoJS
6       var data = CryptoJS.AES.encrypt(JSON.stringify(survey.data),
7         sessionStorage.getItem('password'));
8       //speichern der aktuellen Seite
9       var page = survey.currentPageNo;
10      //erstellen HTTPRequests
11      var req = new XMLHttpRequest();
12      var saveData = data.concat('::', page);
13      var body = Patientmodule.token.concat('::', saveData);
14      req.open('POST', 'https://patientmodule.devlocal/save', true);
15      req.send(body);
16    }
17  });
```

Listing 6.3: Speicherung des teilausgefüllten Fragebögen.

Um anschließend einen bereits begonnenen Fragebogen fortsetzen zu können, wird vor der Generierung des Fragebogens mittels SurveyJS zunächst geprüft, ob Daten von einem begonnenen Fragebogen vorliegen. Ist dies der Fall wird das incomplete-Flag auf `true` gesetzt und entsprechend nach dem Laden des Fragebogens dieser mit den vorhandenen Daten ausgefüllt, nachdem diese entschlüsselt wurden (vgl Listing 6.4). Hierbei kann ein weiteres Problem entstehen. Sollte der Benutzer sich beim Eingeben seines Passwortes vertippt, oder generell ein falsches Passwort eingegeben haben, kann dies erst nach der Entschlüsselung der Daten festgestellt werden.

## 6.2 Auszüge aus der Implementierung

```
1 if(Patientmodule.incomplete == true){
2   survey.data = CryptoJS.AES.decrypt(Patientmodule.savedData, sessionStorage.
   getItem('password'));
3   survey.currentPageNo = Patientmodule.page;
4 }
```

Listing 6.4: Laden des Fragebogens mit vorhandenen Daten.



# 7

## Anforderungsabgleich

An dieser Stelle wird die tatsächlich erreichte Funktionalität des Systems mit den in 4.3 und 4.4 aufgelisteten Funktionalitäten verglichen. Dabei wird zwischen folgenden Erfüllungsgraden unterschieden.

- (5) Die Funktionalität wird vollständig erfüllt.
- (4) Die Funktionalität wird zufriedenstellend erfüllt.
- (3) Die Funktionalität wird ausreichend erfüllt.
- (2) Die Funktionalität wird teilweise erfüllt.
- (1) Die Funktionalität wurde vorbereitet.
- (0) Die Funktionalität wird nicht erfüllt.

### 7.1 Funktionale Anforderungen

Es werden zunächst wieder die funktionalen Anforderungen, wie in Kapitel 4.3 beschrieben, betrachtet.

## 7 Anforderungsabgleich

Nummer	Titel	Priorität	Erfüllungsgrad
FA01	Fortschrittsanzeige	verpflichtend	(5)
FA02	Ablauf des Fragebogens	verpflichtend	(5)
FA03	Abgabe eines Fragebogens	verpflichtend	(5)
FA04	Worklist	verpflichtend	(5)
FA05	Reihenfolge der Fragebögen	verpflichtend	(5)
FA06	Datenschutz	verpflichtend	(5)
FA07	Nutzerfeedback	optional	(1)
FA08	Zeitmessung	optional	(0)
FA09	Sprache	verpflichtend	(5)
FA10	Rückmeldung	verpflichtend	(4)

Tabelle 7.1: Funktionale Anforderungen

## 7.2 Nicht-funktionale Anforderungen

Die nicht-funktionalen Anforderungen definieren die Qualität des Systems in Bezug auf Benutzerfreundlichkeit sowie Stabilität und Sicherheit des Systems. Der tatsächlich erreichte Erfüllungsgrad der in Kapitel 4.4 festgelegten Anforderungen wird hier abgeglichen.

Nummer	Titel	Erfüllungsgrad
NA01	Usability	(5)
NA02	Fehlerprävention	(5)
NA03	Wartbarkeit	(5)
NA04	Robustheit	(5)
NA05	Effizienz	(5)
NA06	Datenschutz	(5)

Tabelle 7.2: Nicht-funktionale Anforderungen

# 8

## Zusammenfassung und Ausblick

In diesem abschließenden Kapitel werden die Inhalte dieser Arbeit nochmals zusammengefasst. Anschließend schließt diese Arbeit mit einem Ausblick auf Möglichkeiten der Erweiterung des Patientenmoduls ab.

### 8.1 Zusammenfassung

Ziel dieser Arbeit war es, ein System zu entwickeln, welches den Benutzern der Tinnitus Database erlaubt, die bisher in Papierform verwendeten Fragebögen nun digital ausfüllen zu können. Dazu wurden anhand einer Analyse des Tinnitus, sowie einem Vergleich von bereits existierenden Projekten, die Anforderungen an ein solches System definiert. Die Tinnitus Database wurde anschließend, basierend auf den Anforderungen (vgl. Kapitel 4), um Funktionalitäten erweitert sowie das Patientenmodul entwickelt. Das System soll dabei möglichst intuitiv in der Bedienung sein und den Benutzer durch den Prozess geleiten. Die leicht erweiterbare Mehrsprachigkeit ermöglicht, dass das System von vielen Benutzern in ihrer Landessprache verwendet werden kann, was die Benutzerfreundlichkeit weiter fördert.

### 8.2 Ausblick

Das aktuelle System beschränkt sich zunächst darauf die essenziellen Funktionalitäten bereitzustellen, welche notwendig sind, um in der Praxis Verwendung finden zu können. Dabei bieten sich jedoch Möglichkeiten der Erweiterung oder Verbesserung des System für die Zukunft.

## 8 Zusammenfassung und Ausblick

Zunächst gilt zu beachten, dass das Patientenmodul in seiner aktuellen Version sehr stark auf die Verwendung mit der Tinnitus Database ausgelegt ist. Um in der Zukunft die Verwendung mit anderen medizinischen Datenbanken zu erleichtern, wären einige Anpassungen an der Web Schnittstelle notwendig. So könnte das Patientenmodul beispielsweise um ein REST Interface erweitert werden, was Entwicklern anderer Systeme die Kommunikation mit dem Patientenmodul erleichtern würde .

Weiter lässt sich das System um einige Funktionalitäten erweitern. So bestand die Überlegung, die Zeit, welche ein Benutzer benötigt, um einen Fragebogen auszufüllen, sowie die Anzahl der Unterbrechungen/Zwischenspeicherungen, zu messen, da diese Anhaltspunkte dafür geben, wie aussagekräftig die entsprechenden Daten sind.

Hinzu kommt, dass das System in der aktuellen Version nur für Patienten zugänglich ist, wenn diese über eines der verwendeten Token verfügen. Patienten müssen somit bereits in der Tinnitus Database angelegt sein, um das Patientenmodul verwenden zu können. Um dieses Problem zu umgehen, könnte das System um einen Registrierungsprozess erweitert werden. Hierbei gilt jedoch zu beachten, dass eine solche Funktionalität in der aktuellen Version nicht implementiert wurde, um die anderweitig im System vollkommene Anonymität der Benutzer nicht zu kompromittieren.

In Kapitel 6.2.2 wurden einige Probleme im Zusammenhang mit der Zwischenspeicherung der Daten aus den Fragebögen beschrieben. Ein möglicher Ansatz um diese zu beheben, wäre dem Benutzer Merksätze vorzuschlagen, welche zu jeder Zeit für den Benutzer auf das gleiche Passwort hindeuten. Dabei sollte beachtet werden das einige in anderen Systemen häufig verwendete Merksätze, wie beispielsweise "Mädchenname der Mutter", sich unter Umständen nicht für die Verwendung im Patientenmodul eignen, da durch sie die Anonymität des Benutzers gefährdet werden könnte.

Um dem zweiten beschriebenen Problem, welches entsteht wenn ein Benutzer sich beim Eingeben des Passworts vertippt, entgegenzuwirken, kann vor dem Einfügen der Daten geprüft werden, ob diese noch in valider JSON Formatierung vorliegen. Sollte dabei festgestellt

werden, dass dies nicht der Fall ist, kann der Benutzer zur Startseite zurückgeleitet werden um das Passwort erneut einzugeben.



# Literaturverzeichnis

- [1] AL-SWIAHB, Jamil ; PARK, Shi N.: *Characterization of Tinnitus in Different Age Groups: A Retrospective Review*. Noise Health, 2016
- [2] MØLLER, A.R. ; LANGGUTH, B. ; DERIDDER, D. ; KLEINJUNG, T.: *Textbook of Tinnitus*. Springer New York, 2010 <https://books.google.de/books?id=YStcWFsxxQZEC>. – ISBN 9781607611455
- [3] HALL, Deborah A. ; LÁINEZ, Miguel J. ; NEWMAN, Craig W. ; SANCHEZ, Tanit G. ; EGLER, Martin ; TENNIGKEIT, Frank ; KOCH, Marco ; LANGGUTH, Berthold: Treatment options for subjective tinnitus: Self reports from a sample of general practitioners and ENT physicians within Europe and the USA", journal="BMC Health Services Research. 11 (2011), Nov, Nr. 1, 302. <http://dx.doi.org/10.1186/1472-6963-11-302>. – DOI 10.1186/1472-6963-11-302
- [4] *Tinnitus Database*. <https://www.tinnitus-research-db.com/welcome>, Abruf: 28.8.2018
- [5] LEVINE, R. A.: Somatic modulation appears to be a fundamental attribute of tinnitus. In: *Proceedings of the Sixth International Tinnitus Seminar, 1999*
- [6] *Tinnitus Research Initiative*. <https://www.tinnitusresearch.com>, Abruf: 29.8.2018
- [7] *Case Report Form*. [https://de.wikipedia.org/wiki/Case\\_Report\\_Form](https://de.wikipedia.org/wiki/Case_Report_Form), Abruf: 29.8.2018
- [8] DAEMEN, Joan ; RIJMEN, Vincent: *The design of Rijndael: AES — the Advanced Encryption Standard*", publisher = SSpringer-Verlag. 2002. – 238 S. – ISBN 3-540-42580-
- [9] *AES: Beginner's Guide*. <https://thebestvpn.com/advanced-encryption-standard-aes/>, Abruf: 12.9.2018
- [10] *Advanced Encryption Standard*. [https://www.tutorialspoint.com/cryptography/advanced\\_encryption\\_standard.htm](https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm), Abruf: 12.9.2018

## Literaturverzeichnis

- [11] PRYSS, Rüdiger ; SCHLEE, Winfried ; LANGGUTH, Berthold ; REICHERT, Manfred: Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback. In: *6th IEEE International Conference on AI & Mobile Services (IEEE AIMS 2017)*, IEEE Computer Society Press, June 2017
- [12] MEIKLE, Mary B.: Electronic access to tinnitus data: The Oregon Tinnitus Data Archive. In: *Otolaryngology - Head and Neck Surgery* 117 (1997), Nr. 6, 698 - 700. [http://dx.doi.org/https://doi.org/10.1016/S0194-5998\(97\)70055-X](http://dx.doi.org/https://doi.org/10.1016/S0194-5998(97)70055-X). – DOI [https://doi.org/10.1016/S0194-5998\(97\)70055-X](https://doi.org/10.1016/S0194-5998(97)70055-X). – ISSN 0194-5998
- [13] LANDGREBE, Michael ; ZEMAN, Florian ; KOLLER, Michael ; EBERL, Yvonne ; MOHR, Markus ; REITER, Jean ; STAUDINGER, Susanne ; HAJAK, Goeran ; LANGGUTH, Berthold: The Tinnitus Research Initiative (TRI) database: A new approach for delineation of tinnitus subtypes and generation of predictors for treatment outcome. In: *BMC Medical Informatics and Decision Making* 10 (2010), Aug, Nr. 1, 42. <http://dx.doi.org/10.1186/1472-6947-10-42>. – DOI 10.1186/1472-6947-10-42. – ISSN 1472-6947
- [14] SCHOBEL, Johannes ; PRYSS, Rüdiger ; SCHICKLER, Marc ; RUF-LEUSCHNER, Martina ; ELBERT, Thomas ; REICHERT, Manfred: End-User Programming of Mobile Services: Empowering Domain Experts to Implement Mobile Data Collection Applications. In: *5th IEEE International Conference on Mobile Services (MS 2016)*, IEEE Computer Society Press, May 2016, 1-8
- [15] *Chrodis+*. <https://www.chrodis.eu>, Abruf: 10.9.2018
- [16] SCHOBEL, Johannes ; PRYSS, Rüdiger ; SCHLEE, Winfried ; PROBST, Thomas ; GEBHARDT, Dominic ; SCHICKLER, Marc ; REICHERT, Manfred: Development of Mobile Data Collection Applications by Domain Experts: Experimental Results from a Usability Study. In: *29th International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, Springer, June 2017 (LNCS 10253), 60-75
- [17] *Laravel*. <https://www.laravel.com>, Abruf: 29.8.2018
- [18] *SurveyJs*. <https://www.surveyjs.io>, Abruf: 29.8.2018
- [19] *PHP*. <http://php.net/>, Abruf: 12.9.2018

- [20] *Nginx*. <https://www.nginx.com/>, Abruf: 12.9.2018
- [21] *SQL*. <https://www.mysql.com/de/>, Abruf: 12.9.2018
- [22] *SurveyJs Builder Tool*. <https://www.surveyjs.io/Survey/Builder/>, Abruf: 29.8.2018
- [23] *Laravel Package: laracasts/utilities*. <https://github.com/laracasts/PHP-Vars-To-Js-Transformer/>, Abruf: 12.9.2018
- [24] PRYSS, Rüdiger ; SCHOBEL, Johannes ; REICHERT, Manfred: Requirements for a Flexible and Generic API Enabling Mobile Crowdsensing mHealth Applications. In: *4th Int'l Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS), RE'18 Workshops, 2018*



Name: Dennis Rodriguez

Matrikelnummer: 845535

**Erklärung**

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Dennis Rodriguez