



Konzeption und Realisierung eines generischen Statistik-Moduls zur Auswertung klinischer Longitudinaldaten am Beispiel der Internationalen Tinnitus-Datenbank

Masterarbeit an der Universität Ulm

Vorgelegt von:

Norman Thiel
norman.thiel@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert
Dr. Rüdiger Pryss

Betreuer:

Dr. Winfried Schlee

2016

Kurzfassung

Tinnitus ist eine weit verbreitete Krankheit, die jeder vierte Deutsche zumindest in ihrer leichten Form schon einmal wahrgenommen hat. Die Auswirkungen der Krankheit unterscheiden sich und können die Betroffenen so stark einschränken, dass sie sich nicht mehr konzentrieren oder schlafen können. Die Therapiemöglichkeiten der Patienten werden mit Hilfe von Studien verbessert. Diese Studien sind jedoch sehr aufwendig, da die Datenaufbereitung umständlich ist und spezielle Kenntnisse verlangt. Gerade in der heutigen hochtechnologisierten Zeit müssen neue, effizientere Lösungen für die Datenaufbereitung gefunden werden, um die Forschung zu unterstützen.

Im Rahmen dieser Arbeit wird ein Statistikmodul für die Internationale Tinnitus Datenbank unter der Tinnitus Research Initiative konzipiert und entwickelt, das den Aufwand für die Datenaufbereitung reduziert und mehr Zeit für die Analyse schafft. Das Modul soll die Auswertung durch Grafiken unterstützen, um kausale Zusammenhänge schneller sichtbar zu machen. Weiterhin wird die Präsentation des Moduls für den Nutzer leicht und verständlich dargestellt.

Inhaltsverzeichnis

Kurzfassung	iii
1 Einleitung	1
1.1 Problemstellung	1
1.2 Ziel der Arbeit	2
1.3 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 Tinnitus	3
2.2 Tinnitus Research Initiative	4
2.3 Internationale Tinnitus Datenbank	4
2.4 Zentrale Begriffe	5
2.4.1 Zeitpunkte	5
2.4.2 Fragebogen	5
2.4.3 Statistische Gruppe	6
2.4.4 Kriterium	6
2.4.5 Filter	7
2.4.6 Longitudinale Statistik	8
2.4.7 Crosssectionale Statistik	8
3 Anforderungen	9
3.1 Vorhandenes Statistikprogramm	9
3.1.1 IBM SPSS Statistics	9
3.2 Funktionale Anforderungen	10
3.2.1 Erstellung von Statistiken	10
3.2.2 Wahl der Statistik	10
3.2.3 Auswahl der Zeitpunkte	10
3.2.4 Erstellung von statistischen Gruppen	10
3.2.5 Auswahl der Kriterien	11
3.2.6 Darstellung der Statistiken	11
3.2.7 Export der Statistiken	11
3.3 Nicht-funktionale Anforderungen	11
3.3.1 Bedienbarkeit	11
3.3.2 Erreichbarkeit und Design	12
3.3.3 Effizienz und Wartezeit	12

4 Entwurf	13
4.1 Architektur der Tinnitus Datenbank	13
4.2 Datenmodell	14
4.3 Erstellung einer Statistik	15
4.3.1 Statistikart auswählen	16
4.3.2 Zeitpunkte auswählen	16
4.3.3 Statistische Gruppen erstellen	17
4.3.4 Kriterien auswählen	18
4.4 Gestaltung des Wizards	18
4.4.1 Wahl der Statistikart	19
4.4.2 Wahl der Zeitpunkte	19
4.4.3 Erstellung der Gruppen	20
4.4.4 Wahl der Kriterien	21
4.5 Gestaltung der Auswertung	21
4.5.1 Auswertung der statistischen Gruppen	21
4.5.2 Individualdaten eines Kriteriums	22
5 Prototypische Realisierung	23
5.1 Verwendete Technologien	23
5.1.1 PHP	23
5.1.2 JavaScript	24
5.1.3 jQuery	24
5.1.4 MySQL	24
5.2 Verwendete Frameworks	24
5.2.1 Laravel	24
5.2.2 Bootstrap	25
5.2.3 AdminLTE	25
5.2.4 Chart.js	25
5.3 Ausgewählte Implementierungsaspekte	25
5.3.1 Nutzereingaben erfassen	26
5.3.2 Finden geeigneter Daten	28
5.3.3 Berechnung der Statistik	30
5.3.4 Anzeigen der Statistik	33
5.3.5 Grafische Umsetzung des Statistikmoduls	34
6 Zusammenfassung und Ausblick	37
6.1 Zusammenfassung	37
6.2 Ausblick	38
A Anhang	39
A.1 Datenmodell	39

1 Einleitung

Tinnitus ist eine weit verbreitete Erkrankung und nur schwer therapierbar. Obwohl es viele verschiedene Therapiemöglichkeiten gibt, können nicht alle Patienten davon profitieren, da nicht allen die gleichen Therapien helfen [29]. Es wird angenommen, dass es unterschiedliche Formen von Tinnitus gibt, die auch verschiedene Therapien benötigen. Eine große Herausforderung ist diese unterschiedlichen Formen anhand von Prädiktoren oder Merkmalen zu bestimmen, um so die vielversprechendsten Therapien für die Patienten zu identifizieren. Bisher sind solche Prädiktoren noch weitgehend unbekannt. Die Suche nach aussagekräftigen Prädiktoren ist jedoch erschwert. Bei klinischen Studien nehmen oft nur eine geringe Anzahl an Probanden teil [29] und die Datenaufbereitung ist zeitaufwendig und umständlich.

Mit Hilfe von unterstützender Software wird das Identifizieren der Prädiktoren und somit das Finden von geeigneten Therapien vorangetrieben. Die Daten können auf Servern gesammelt und ausgewertet werden. Der Zugriff auf die Daten kann über das Internet und mit einem Smartphone sogar unterwegs erfolgen. Bei Besprechungen und Konferenzen können die Ergebnisse so recht simpel und schnell gezeigt, repliziert, geteilt und diskutiert werden, um die weltweite Tinnitusforschung voranzutreiben.

1.1 Problemstellung

Die Internationale Tinnitus Datenbank unter der Tinnitus Research Initiative beschäftigt sich unter anderem mit der Identifizierung von Tinnitusformen und ihren Prädiktoren. Dafür werden Fragebögen während unterschiedlichen Zeitpunkten der Therapie von Tinnitus-Patienten ausgefüllt. Mithilfe der Fragebögen können Merkmale über den Patienten, sowie die Art und den Schweregrad seines Tinnitus erfasst werden. Die erfassten Daten werden anschließend von Mitarbeitern händisch in eine Datenbank übertragen. Diese Daten werden dann zur Erstellung von Statistiken und zu Forschungszwecken herangezogen.

Die Erstellung einer solchen Statistik ist sehr zeitaufwendig. Den Großteil der Arbeit macht jedoch nicht die Analyse, sondern die Datenaufbereitung aus. Zuerst müssen die Daten aus der Datenbank in ein *Statistical Package for the Social Sciences* (SPSS) Format exportiert werden. Ein SPSS Skript berechnet daraus dann Summenscores der Fragebögen und hängt diese der Exportdatei an. Anschließend muss die Datei in *R* importiert und nach relevanten Versuchspersonen, Zeitpunkten und Variablen gefiltert werden. Mit den aufbereiteten Daten kann nun die Analyse durchgeführt werden. Eine grafische Darstellung der Ergebnisse gibt es jedoch nicht.

Dieser Vorgang kann mehrere Tage dauern, da für die Prozesse unter Umständen nur bestimmte Mitarbeiter mit speziellen Kenntnissen in SPSS zur Verfügung stehen.

1.2 Ziel der Arbeit

In dieser Arbeit wird die Konzeption und Realisierung eines Statistikmoduls für die bereits bestehende Tinnitus Datenbank beschrieben. Ziel ist es, den in Kapitel 1.1 beschriebenen Aufwand für die Datenaufbereitung wesentlich zu reduzieren und den Fokus auf die Datenanalyse zu legen. Die Erstellung einer Statistik mit dem neuen Statistikmodul soll keine Kenntnisse über SPSS oder eine Programmiersprache voraussetzen. Die Filterung relevanter Daten und das Erstellen von statistischen Gruppen soll über ein benutzerfreundliches Bedienkonzept erfolgen. Für eine schnelle Analyse stehen dem Nutzer Grafiken, deskriptive Statistiken und Exportmöglichkeiten zur Verfügung.

1.3 Aufbau der Arbeit

Kapitel 1 schafft einen Überblick über die vorliegende Arbeit und dessen Thematik. In **Kapitel 2** werden Symptome des Tinnitus erläutert und die forschende Einrichtung dieser Arbeit vorgestellt. Desweiteren werden die zentrale Begriffe der Arbeit eingeführt. In **Kapitel 3** wird zunächst bereits vorhandene Software betrachtet und anschließend funktionale und nicht-funktionale Anforderungen des Statistikmoduls erhoben. **Kapitel 4** erläutert zunächst die Systemarchitektur der Tinnitus Datenbank und das zugrunde liegende Datenmodell. Anschließend werden anhand der Anforderungen der Erstellprozess der Statistik mittels eines Wizards entwickelt und das Design des Wizards sowie der Auswertung entworfen. In **Kapitel 5** wird auf Basis des Entwurfs die Implementierung erläutert. Dazu werden die verwendeten Technologien beschrieben und ausgewählte Aspekte der Implementierung vorgestellt. **Kapitel 6** fasst die Inhalte der Arbeit zusammen und gibt einen Ausblick auf zukünftige Entwicklungen des Statistikmoduls.

2 Grundlagen

In diesem Kapitel werden die Grundlagen dieser Arbeit erläutert um ein grundlegendes Verständnis zu schaffen. Es wird geklärt, was Tinnitus ist und welche Rolle die Tinnitus Research Initiative und die Internationale Tinnitus Datenbank in der Tinnitusforschung einnehmen. Desweiteren werden zentrale Begriffe dieser Arbeit erläutert.

2.1 Tinnitus

Mit *Tinnitus*, vom Lateinischen Wort *Tinnitus aurium* abstammend, was „das Klingeln der Ohren“ bedeutet, werden Ohrgeräusche oder das Ohrensausen bezeichnet. Diese werden oft als Pfeifen, Rauschen, Zischen oder Summen erlebt. Jeder vierte Deutsche hat dieses Phänomen zumindest vorübergehend schon einmal wahrgenommen [5].

Es wird unterschieden in subjektiven und objektiven Tinnitus. Beim viel häufiger auftretenden subjektiven Tinnitus gibt es keine für andere Personen wahrnehmbare äußere Quelle des Geräusches. Das Geräusch hört nur der Betroffene selbst. In seiner leichten Form kann der Tinnitus von Umgebungsgeräuschen unterdrückt werden oder der Patient gewöhnt sich an ihn. In der schweren Form werden Betroffene so stark im Alltag behindert, dass sie oft nicht schlafen, sich konzentrieren oder psychischen Belastungen standhalten können [5]. Bekannte Ursachen für das Auftreten von Tinnitus können Lärmschädigungen, Schwerhörigkeit, ein Hörsturz, ein Schädel-Hirn-Trauma, Otosklerose, Akustikus-Neurinom, eine Vergiftung des Innenohres, ein muskulär bedingter oder somatoformer Tinnitus oder Morbus Menière sein. Es kann jedoch bei etwa 45 Prozent der Betroffenen sowohl bei der Erstuntersuchung als auch im weiteren Verlauf der Behandlung keine Ursache für Tinnitus gefunden werden [6]. 1,1 Prozent der Deutschen, das entspricht 1,5 Millionen Menschen, leiden an der mittleren bis sehr schweren Form von Tinnitus [5].

Beim objektiven Tinnitus werden körpereigene Geräusche wahrgenommen. Ursache hierfür sind oft Schallquellen in Ohr-Nähe, wie etwa Tumore, Gefäßmissbildungen oder auch Zuckungen der Muskeln, die vom Arzt erfasst und wenn nötig operativ entfernt werden können. Nur bei einem Prozent aller Tinnitusfälle handelt es sich um den objektiven Tinnitus [5].

Die Zahl der Tinnitus-Patienten ist in den letzten Jahrzehnten vor allem in den westlichen Industrienationen stetig angestiegen. Deshalb wird der Tinnitus auch als Volkskrankheit bezeichnet. Ob jedoch die Zahl der Betroffenen tatsächlich angestiegen ist oder sich nur mehr Betroffene in medizinische Behandlung begeben haben, ist unklar [44].

2.2 Tinnitus Research Initiative

Die *Tinnitus Research Initiative* (TRI) ist eine gemeinnützige Forschungseinrichtung, die sich für eine Verbesserung der Lebensqualität der unter Tinnitus leidenden Patienten engagiert. Im Mittelpunkt steht die Unterstützung der biomedizinischen Forschung, die dann zu neuen effektiveren Therapiemöglichkeiten für Patienten führt. Die TRI handelt nach dem Prinzip, dass die Zusammenarbeit verschiedener Wissenschaftsbereiche für ein besseres Verständnis des Tinnitus sorgen und daraus effektivere Therapien entwickelt werden können [28, 9]. Die TRI konnte bereits einige neue Erkenntnisse von einer Vielzahl an Forschungsprojekten und wissenschaftlichen Ausarbeitungen erzielen [13, 14, 2, 27, 10, 17, 15, 16, 22, 23, 24, 20, 21, 13, 18, 19, 25, 12, 11, 26].

2.3 Internationale Tinnitus Datenbank

Die *Internationale Tinnitus Datenbank* (TD) ist die erste internationale Zusammenarbeit von auf Tinnitus spezialisierten Kliniken. Momentan nehmen 19 Zentren aus 11 Ländern an dem Projekt teil (Abbildung 2.1). Seit dem Start des Projekts im Sommer 2008 wurden bereits fast 3000 Patienten weltweit aufgenommen, ihr Krankheitsverlauf dokumentiert und in der Datenbank gespeichert [29]. Diese Vielzahl ermöglicht es, schnell auf für Studien relevante Daten, in größeren Stichproben zurückzugreifen und hilft somit Prädiktoren für verschiedene Tinnitusformen (Kapitel 1 und 2.1) zu bestimmten und geeignete Therapien zu entwickeln. Die Ziele der TD sind in Tabelle 2.1 aufgelistet [29].



Abbildung 2.1: Weltkarte der teilnehmenden Zentren [30].

Ziele

- Identifizierung der verschiedenen Formen des Tinnitus, sie in Zusammenhang mit ihren Symptomen bringen und geeignete Therapiemöglichkeiten finden
- Identifizierung der Prädiktoren auf Reaktionen von spezifischen Behandlungen
- Bewertung der Behandlungsergebnisse zu spezifischen Behandlungen mit modularer Herangehensweise
- Erklärung unstimmiger Ergebnisse in verschiedenen Studien
- Sammeln von epidomologischen Daten
- Kreuzvalidierung von verschiedenen Bewertungsinstrumenten in verschiedenen Sprachen
- Entwicklung eines individualisierten Behandlungsalgorithmus für jeden Patienten basierend auf seinem Diagnoseprofil

Tabelle 2.1: Übersicht der Ziele der TD

2.4 Zentrale Begriffe

In diesem Abschnitt sind die für das Statistikmodul und die statistische Auswertung relevanten Begriffe aufgeführt und kurz erläutert.

2.4.1 Zeitpunkte

Zeitpunkte repräsentieren verschiedene Phasen der Behandlung. Es wird unterschieden in *Screening* (Patientenaufnahme), *Baseline* (Ersterfassung des Krankheitsbildes), *Visit* (weitere Untersuchungen), *Final Visit* (finale Untersuchung) und *Follow Up* (Nachfolgeuntersuchung). Nicht jeder Fragebogen darf zu jedem Zeitpunkt ausgefüllt werden. Beispielsweise darf der *Tinnitus Sample Case History Questionnaire* nur während des Screenings ausgefüllt werden. Fragebögen für Screening, Visite und Follow Up können mehrmals ausgefüllt werden. Das bedeutet, dass beispielsweise ein Patient den selben Fragebogen in verschiedenen Visiten ausgefüllt hat.

2.4.2 Fragebogen

Die Fragebögen sind die wesentlichen Elemente der TD. Alle Fragebögen sind standardisiert und erfüllen ein bestimmtes Ziel [45]. Beispielsweise verfolgt der Fragebogen *Tinnitus Severity* den Zweck, den Tinnitus Schweregrad des jeweiligen Patienten zu ermitteln. Den Antwortmöglichkeiten der Fragen im Fragebogen werden Punktwerte zugeordnet, sodass sich ein Summenscore für den Fragebogen mit Hilfe eines Algorithmus berechnen lässt. Anhand des Summenscores können verschiedene Aussagen über den Patient oder den Tinnitus getroffen werden. Jeder Fragebogen kann jedoch nur zu bestimmten Behandlungszeitpunkten erfasst werden. Eine Übersicht aller Fragebögen wird in Tabelle 2.2 angezeigt.

Kürzel	Fragebogen
TSCHQ	Tinnitus Sample Case History Questionnaire
AE	Audiological Examination
THI	Tinnitus Handicap Inventory
TBF-12	Tinnitus Impairment Questionnaire
TS	Tinnitus Severity
BDI	Beck Depression Inventory
MDI	Major Depression Inventory
WHOQOL-BREF	Questions about the general health status
TQ	Tinnitus Questionnaire
Mini-TQ	Mini Tinnitus Questionnaire (Auswahl an Fragen aus TQ)
TFI	Tinnitus Functional Index
CGI	Clinical Global Impression

Tabelle 2.2: Übersicht aller Fragebögen

2.4.3 Statistische Gruppe

Eine *statistische Gruppe* (SG) ist eine Auswahl von Patientendaten unter relevanten Filtern und Zeitpunkten aus der TD. Diese werden unter gewählten Kriterien ausgewertet und können mit anderen SGs verglichen werden. Eine SG könnte beispielsweise alle Frauen im Alter von 20 bis 40 Jahren erfassen, die einen THI Summenscore kleiner 20 haben.

2.4.4 Kriterium

Die Kriterien geben an, unter welchen Gesichtspunkten die Statistik für eine oder mehrere SGs erstellt werden soll. Ist beispielsweise das Kriterium THI selektiert, so wird für die Patientendaten der ausgewählten Gruppen eine Statistik zum THI erstellt. Im Folgenden ist eine Übersicht aller Kriterien sichtbar (Tabelle 2.3).

Kriterien	
THI	BDI
TBF-12	CGI
TQ	MDI
Mini-TQ	WHOQOL-BREF Dimensionen 1 - 4

Tabelle 2.3: Übersicht aller Kriterien

2.4.5 Filter

Als Filter werden alle Merkmale oder Summenscores bezeichnet nach denen SGs sortiert werden können. Alle numerischen Filter (Tabelle 2.4), wie etwa Alter oder THI, können größere Zahlenbereiche filtern, beispielsweise *Alter > 10*. Nicht-numerische Filter (Tabelle 2.5), wie etwa der Name eines Zentrums, sortieren nur nach dem spezifisch ausgewählten Wert aus, beispielsweise *Zentrum = Regensburg*. Alle Filter verfügen über eine Option, die den gesamten Datenbereich erfasst, also keine Beschränkung darstellt.

Numerische Filter	Beschreibung
Alter	Alter der Patienten
Dauer	Dauer von Wahrnehmung des Tinnitus bis zur Behandlung
THI	Summscore des THI
TBF-12	Summscore des TBF-12
CGI	Summscore des THI
TQ	Summscore des TQ
Mini-TQ	Summscore von ausgewählten Fragen aus dem TQ
MDI	Summscore des MDI
BDI	Summscore des BDI
WHOQOL-BREF D1 - D4	Summscore des WHOQOL-BREF bezogen auf Dimension 1 - 4

Tabelle 2.4: Übersicht aller numerischen Filter

Nicht-numerische Filter	Beschreibung
Dropouts	Patienten, deren Behandlung abgebrochen wurde
Geschlecht	Geschlecht der Patienten
Psychische Probleme	Patienten leiden an psychischen Problemen
Tinnitus Variabilität	Die Lautstärke des Tinnitus ist variabel
Somatischer Tinnitus	Der Tinnitus ist somatisch
Hörverlust	Kategorisierung des Hörverlustes
Behandlungscode	Behandlungsart der Patienten
Zentrum	Behandlungszentrum der Patienten
Anzahl der Behandlungen	Anzahl der Behandlungen der Patienten
Hyperacusis	Patienten leiden an einer Hyperakusis
Tinnituslage	Die Position an der der Tinnitus wahrgenommen wird

Tabelle 2.5: Übersicht aller nicht-numerischen Filter

2.4.6 Longitudinale Statistik

Die *longitudinale Statistik* (LS) wertet eine SG zu mehreren gewählten Zeitpunkten, unter den gewählten Kriterien (Abbildung 2.2) aus. Mehr als eine SG kann nicht erstellt werden.

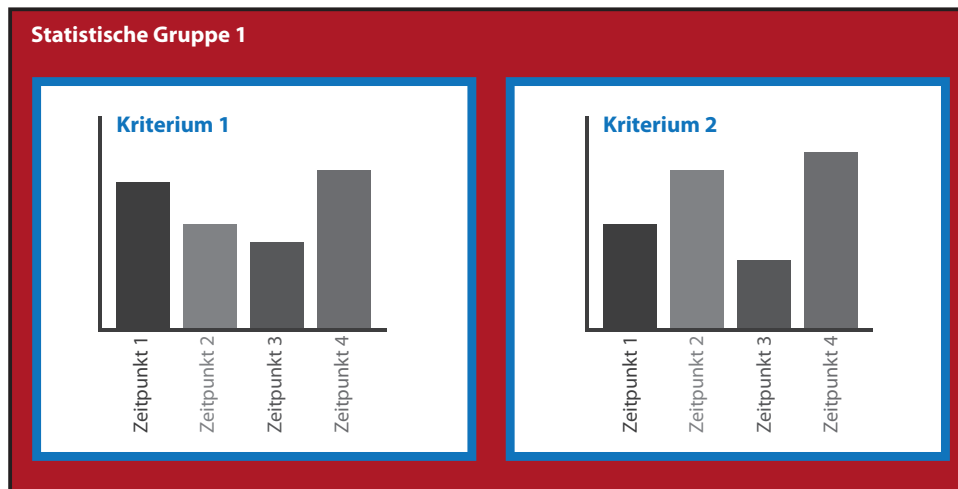


Abbildung 2.2: Longitudinale Statistik

2.4.7 Crossectionale Statistik

Die *crossectionale Statistik* (CS) wertet mehrere SGs zu einem gewählten Zeitpunkt, unter den gewählten Kriterien (Abbildung 2.3) aus. Mehr als ein Zeitpunkt kann nicht gewählt werden.

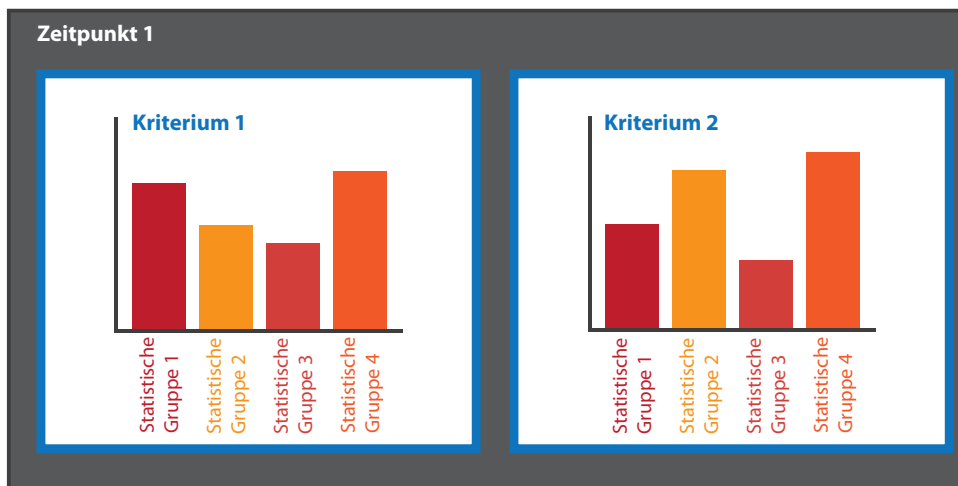


Abbildung 2.3: Crossectionale Statistik

3 Anforderungen

In diesem Kapitel wird ein bereits vorhandenes Statistikprogramm betrachtet und anschließend die funktionalen und nicht-funktionalen Anforderungen an das Statistikmodul erhoben.

3.1 Vorhandenes Statistikprogramm

Im Folgenden wird eine bereits vorhandene Statistiksoftware beschrieben, um einen Einblick über den Funktionsumfang und die Exportmöglichkeiten zu gewinnen.

3.1.1 IBM SPSS Statistics

IBM SPSS Statistics (SPSS steht für Statistical Package for the Social Sciences) ist eine kostenpflichtige Statistiksoftware. die Datenmanagement und statistische, sowie grafische Datenanalysen für die gängigsten statistischen Verfahren bietet [43]. Die Software wird in der Forschung für Marktanalysen, Data Mining und vieles mehr eingesetzt [43].

Funktionsumfang

IBM SPSS Statistics ist modular aufgebaut und beinhaltet bereits im Basismodul viele Funktionalitäten wie beispielsweise deskriptive Statistiken, Clusteranalysen, Verhältnisstatistiken, Kreuztabellen, T-Tests und vieles mehr [7]. Die Ergebnisse können in Tabellen, Diagrammen, 2D oder 3D Grafiken und vielen weiteren Formaten präsentiert werden [7]. Mit dem Hinzukauf von weiteren Modulen kann der Funktionsumfang für die eigenen Bedürfnisse erweitert werden. Beispielsweise können mit dem Modul *SPSS Statistics Programmability Extension* neue Prozeduren, die in anderen Programmiersprachen wie Python, R, Java und der .NET-Version von Microsoft Visual Basic geschrieben sind, hinzugefügt werden [8].

Import und Export

Mit IBM SPSS Statistics können Datensätze aus ASCII Textdateien oder verschiedenen relationalen Datenbanken mit ODBC und SQL eingelesen werden. Die Statistiken können unter anderem in Dateiformate wie MS Excel, MS Word, MS PowerPoint, PDF, als HTML und als Textdatei exportiert werden [7].

3.2 Funktionale Anforderungen

In diesem Abschnitt werden die funktionalen Anforderungen für das Statistikmodul erhoben. Funktionale Anforderungen beschreiben den für das Statistikmodul benötigten Funktionsumfang [36] und bilden damit das Fundament für den Entwurf.

3.2.1 Erstellung von Statistiken

Den Erstellprozess einer Statistik durchläuft der Nutzer in vier einzelnen in sich abgeschlossenen Schritten in einem Wizard. Zuerst wählt er die Art der Statistik, anschließend die relevanten Zeitpunkte der Fragebögen. Als nächstes werden SGs erstellt und zuletzt die Kriterien für die Statistik selektiert. Zwischen den verschiedenen Schritten des Wizard kann der Nutzer navigieren um seine Eingaben zu ändern.

3.2.2 Wahl der Statistik

Für das Erheben einer Statistik gibt es zwei Möglichkeiten: das Erheben einer LS oder einer CS (Kapitel 2.4.6 und 2.4.7). Wählt der Nutzer eine dieser Möglichkeiten, wird ihm in den nächsten Schritten des Wizards nur der dafür vorgesehene Erstellprozess gezeigt.

3.2.3 Auswahl der Zeitpunkte

Die Zeitpunkte (Kapitel 2.4.1) werden dem Nutzer aufgelistet angezeigt. Hat der Nutzer sich für eine LS entschieden, kann er beliebig viele Zeitpunkte selektieren. Hat der Nutzer eine CS gewählt, muss genau ein Zeitpunkt selektiert werden. Eine Mehrauswahl ist nicht möglich. Dieser Schritt des Wizards kann erst beendet werden, wenn mindestens ein Zeitpunkt selektiert ist.

3.2.4 Erstellung von statistischen Gruppen

SGs können erstellt oder geladen werden. Erstellte oder geladene SGs werden dem Nutzer mit allen Filtern in einer Übersicht angezeigt. Die Übersicht bietet Möglichkeiten an, um die SG zu speichern, zu editieren oder zu entfernen. Das Erstellen und das Editieren einer SG erfolgt in einem separaten Fenster, in welchem dem Nutzer alle Filter (Kapitel 2.4.5) angezeigt werden und entsprechende Parameter gesetzt werden können. Jede SG kann benannt werden.

Hat der Nutzer eine LS gewählt, muss genau eine SG erstellt oder geladen werden. Eine Mehrauswahl ist nicht möglich. Ist eine CS gewählt, können beliebig viele SGs erstellt oder geladen werden. Dieser Schritt des Wizards kann erst beendet werden, wenn mindestens eine SG erstellt oder geladen ist.

3.2.5 Auswahl der Kriterien

Die Kriterien (Kapitel 2.4.4) werden dem Nutzer aufgelistet angezeigt. Der Nutzer kann beliebige Kriterien selektieren, muss jedoch mindestens eines selektieren um den Schritt des Wizards abschließen zu können. Im Anschluss wird die Statistik erstellt.

3.2.6 Darstellung der Statistiken

Zu jedem Kriterium wird eine Statistik erstellt. Diese werden dem Nutzer in Übersichten untereinander angezeigt. In jeder Übersicht werden deskriptive Statistiken, wie Konfidenzintervall, Mittelwert und Standardabweichung zu jedem Zeitpunkt (LS) oder zu jeder SG (CS) tabellarisch angezeigt. Außerdem werden mit Hilfe eines Diagramms die Ergebnisse visualisiert. Die individuellen Daten, die zu der Auswertung geführt haben, können eingesehen werden. Dem Nutzer werden Exportfunktionen für die Tabellen angeboten.

3.2.7 Export der Statistiken

Wie in 3.2.6 beschrieben bietet die deskriptive Statistik eine Exportfunktion an. Hierbei kann der Nutzer die Tabelle als CSV-Datei, Excel-Datei oder PDF-Datei exportieren oder die deskriptive Statistik drucken.

3.3 Nicht-funktionale Anforderungen

Die nicht-funktionalen Anforderungen für das Statistikmodul werden im Folgenden Abschnitt definiert und sind in die Bereiche Bedienbarkeit, Erreichbarkeit und Leistung und Effizienz gegliedert. Nicht-funktionale Anforderungen beschreiben die Qualitätseigenschaften eines Systems bei der Nutzung [36].

3.3.1 Bedienbarkeit

Das Statistikmodul soll den zeitlichen Aufwand für die Datenaufbereitung reduzieren und den Komfort für die Datenanalyse erhöhen. Dafür wird der Nutzer beim Erstellen einer Statistik leicht und klar durch den Erstellprozess geleitet. Die Unterteilung des Wizards in vereinzelter Schritte soll die Komplexität der Erstellung senken und die Verständlichkeit für den Nutzer erhöhen. Weiterhin soll das Speichern und Laden von SGs dem Nutzer ermöglichen schnell und einfach an die gewünschte Statistik zu gelangen. Die grafische Unterstützung der Auswertung durch Diagramme und die Exportfunktionen sollen dem Nutzer Hilfestellungen bei der Analyse geben.

3.3.2 Erreichbarkeit und Design

Das Statistikmodul soll auf allen Endgeräten abrufbar sein die über einen Browser verfügen. Daher soll das Statistikmodul über ein responsives Design verfügen.

3.3.3 Effizienz und Wartezeit

Die TD verfügt über viele tausend Patientendaten [29]. Daher soll die Berechnung der Statistik effizient erfolgen, um die Wartezeiten möglichst gering zu halten. Da das Statistikmodul auf allen Engeräten über den Browser verfügbar ist, soll die Berechnung auch mit mobilem Internet schnell und zuverlässig erfolgen.

4 Entwurf

In diesem Kapitel wird das zu realisierende Statistikmodul entworfen. Hierfür wird zunächst auf die Architektur der bereits bestehenden TD eingegangen und anschließend das Datenmodell der TD erläutert. Ausgehend von den erhobenen funktionalen und nicht-funktionalen Anforderungen in Kapitel 3.2 und 3.3 entsteht der Entwurf.

4.1 Architektur der Tinnitus Datenbank

Die Architektur der TD baut auf dem Laravel Framework (Kapitel 5.2.1) auf. Die Architektur ist in Abbildung 4.1 veranschaulicht und wird im Folgenden erklärt.

Das *Routing* ist dafür da, Anfragen des Clients an einen *Controller* weiterzuleiten. Hierfür wird in der *routes.php* eine URI mit einem *Controller* und dessen Funktion verbunden. Sobald eine Anfrage des Clients mit dieser URI eingeht, wird die verlinkte Funktion in dem betreffenden *Controller* aufgerufen. Die *Controller* repräsentieren die Anwendungslogik der TD. Sie können mit dem *Model* interagieren, um insbesondere Daten abzufragen oder zu persistieren. Des Weiteren sind sie für das Verschicken der resultierenden *View* als Antwort verantwortlich. Das *Model* repräsentiert insbesondere Daten aus den Tabellen der Datenbank als PHP Klassen. Dafür wird aus dem *Laravel Framework* das Eloquent ORM genutzt. Das *Model* ist mit einer *MySQL Datenbank* verbunden (Kapitel 5.1.4). Diese ist für die Persistierung der Daten der TD verantwortlich. Die *Views* sind für die Darstellung der TD verantwortlich und bestehen aus HTML und Anwendungsdaten. Eine *View* wird im *Browser* angezeigt. Der Nutzer kann mit Hilfe des *Browser* mit der angezeigten *View* interagieren. Als Webserver dient ein *Apache Webserver*.

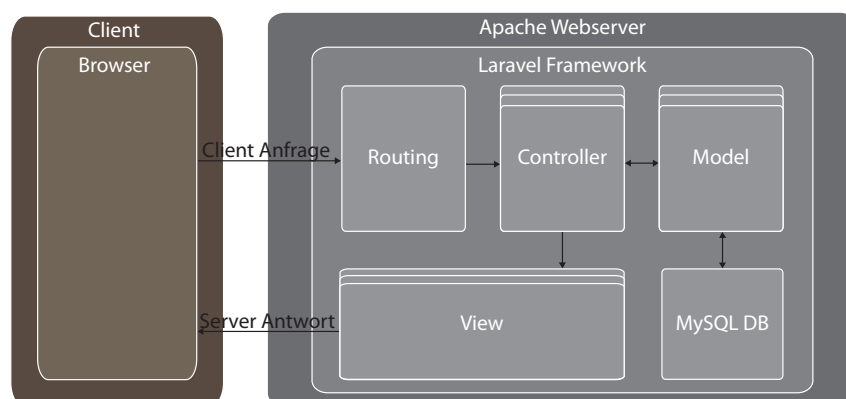


Abbildung 4.1: Architektur der TD

4.2 Datenmodell

Aus Gründen der Übersicht wird in diesem Abschnitt nur auf die für die Statistik besonders relevanten Entitäten der TD eingegangen (Abbildung 4.2). Eine vollständige Übersicht befindet sich im Anhang (Abbildung A.1, A.2 und A.3).

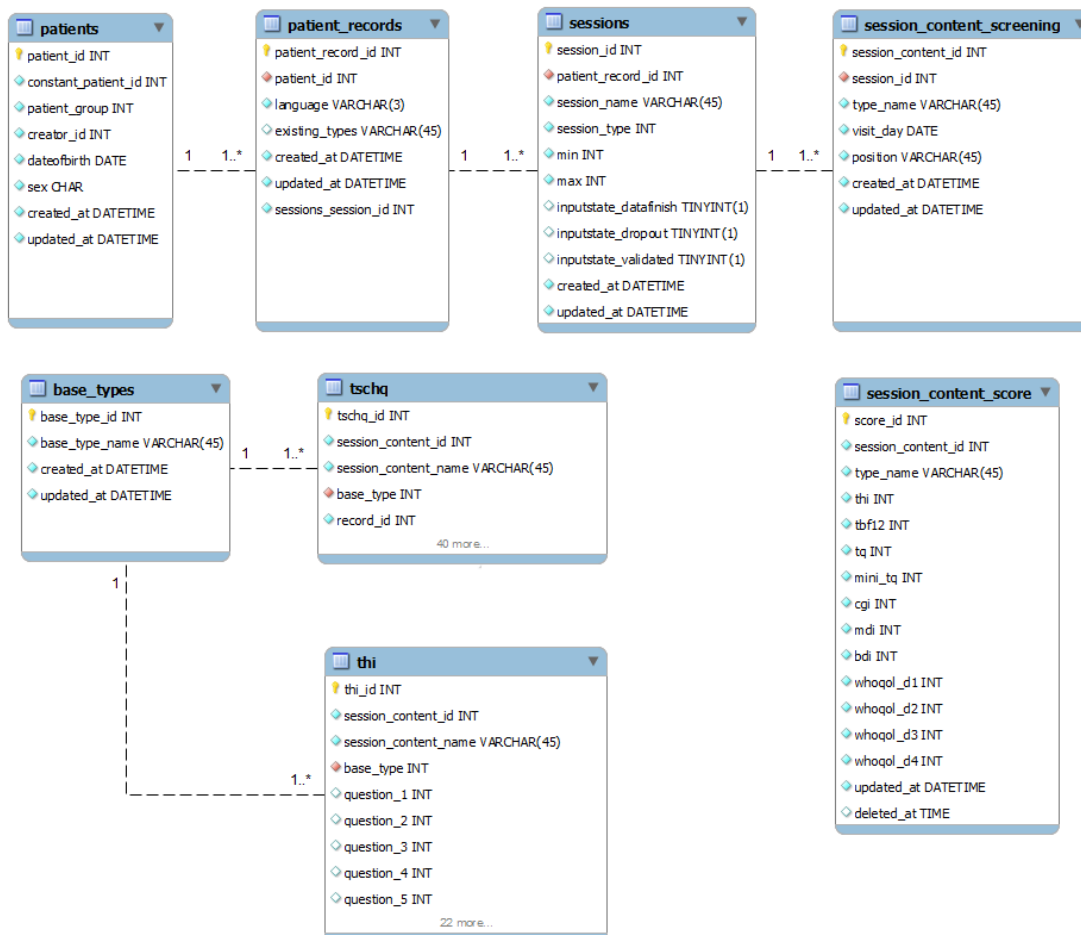


Abbildung 4.2: Datenmodell der TD

Die Entität *patients* repräsentiert die Patienten der TD und enthält wesentliche Merkmale, wie etwa das Geburtsdatum *dateofbirth* und das Geschlecht *sex*. Jeder Patient kann n-viele *patient_records* (PR) besitzen. Ein PR entspricht etwa einem Behandlungszyklus. Ist eine Behandlung abgeschlossen und der Patient wird Jahre später noch einmal behandelt, so wird ein neuer PR angelegt. PRs können n-viele *sessions* besitzen. Eine Session repräsentiert eine Behandlung des Patienten. Behandlungen können beispielsweise abgeschlossen (*inputstate_datafinish*) oder vorzeitig abgebrochen (*inputstate_dropout*) sein. In diesem Fall würde eine neue Session erstellt werden, falls der Patient nochmals behandelt werden sollte. Sessions können n-viele Behandlungszeitpunkte besitzen, zu denen Patienten zur Behandlung kommen und beispielsweise Fragebögen ausfüllen. Die Behandlungszeitpunkte werden in der Abbildung 4.2 durch die Entität *session_content_screening* repräsentiert. Es gibt zusätzlich die Zeitpunkte *sessi-*

on_content_baseline, *session_content_wardround*, *session_content_final_wardround* und *session_content_follow_up*, die aus Übersichtsgründen nicht aufgelistet sind. Jeder Behandlungszeitpunkt verfügt über eine *session_content_id*.

Jeder Fragebogen ist einem *base_type* zugeordnet. Fragebögen, in der Abbildung 4.2 durch den *tschq* und *thi* repräsentiert, verfügen über die Antworten zu den Fragen, beispielsweise *question_1* im *thi*. Weiterhin verfügen sie über den Fremdschlüssel *session_content_id*, womit sie einem Behandlungszeitpunkt zugeordnet werden können. Über die weiteren Fremdschlüssel *session_id* (Tabelle *session_content_screening*) und *patient_record_id* (Tabelle *sessions*) kann jeder Fragebogen genau einem Patienten zugewiesen werden. Weitere Fragebögen sind *tbf12*, *tinnitus_severity*, *tinnitus_questionnaire_gundh*, *cgi*, *mdi*, *whoqol-bref*, *tfi*, *orthological_examination* und *audiological_examination*.

Die Tabelle *session_content_score* beinhaltet alle berechneten Summenscores der Fragebögen zu einem Behandlungszeitpunkt eines Patienten. Da es möglich ist, dass verschiedene Behandlungszeitpunkte die gleiche *session_content_id* besitzen können, wird ein Eintrag eindeutig über das Paar *session_content_id* und *type_name* identifiziert. *Type_name* repräsentiert den Namen des Behandlungszeitpunktes und ist ebenfalls in den Fragebögen enthalten.

4.3 Erstellung einer Statistik

Basierend auf den erhobenen funktionalen Anforderungen unter 3.2.1 wird der Erstellprozess der Statistik mit einem Wizard umgesetzt. Dieser besteht aus den vier in sich abgeschlossenen Schritten *Statistikart auswählen*, *Zeitpunkte auswählen*, *statistische Gruppen erstellen* und *Kriterien auswählen* (Abbildung 4.3). Jeder dieser Schritte kann wiederholt werden, um die Auswahl zu ändern.

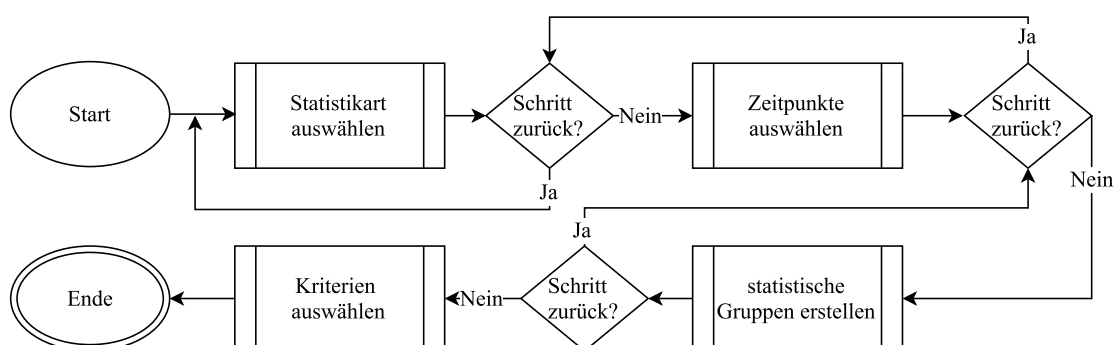


Abbildung 4.3: Flussdiagramm des Wizard

4.3.1 Statistikart auswählen

Die Auswahl der Statistik basiert auf den funktionalen Anforderungen unter Kapitel 3.2.2. Der Nutzer kann zwischen einer LS oder CS entscheiden. Es muss genau eine Art selektiert werden, um den Schritt beenden zu können (Abbildung 4.4).

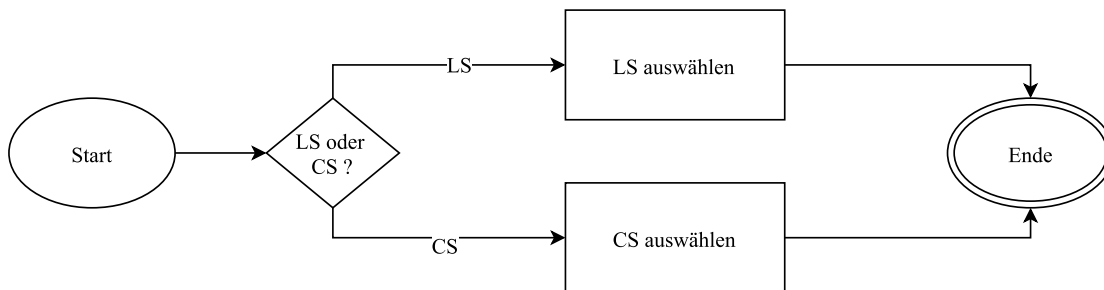


Abbildung 4.4: Flussdiagramm der Statistikauswahl

4.3.2 Zeitpunkte auswählen

Die Auswahl der Zeitpunkte für die Statistik beruht auf den funktionalen Anforderungen unter Kapitel 3.2.3. Hierbei wird unterschieden, ob es sich bei der Statistik um eine CS oder eine LS handelt. Bei einer CS, muss genau ein Zeitpunkt gewählt werden, um den Schritt beenden zu können (Abbildung 4.5).

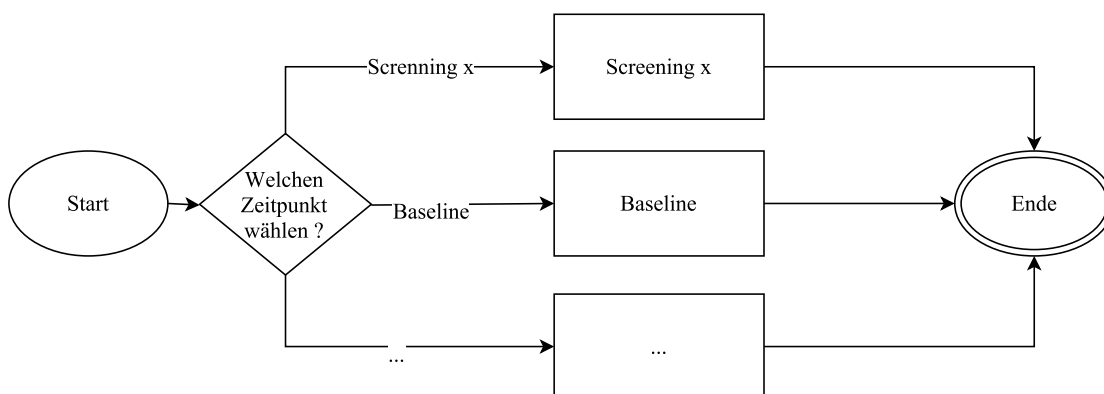


Abbildung 4.5: Flussdiagramm der Auswahl des Zeitpunkts einer CS

Handelt es sich um eine LS, so kann der Nutzer nach der Auswahl eines Zeitpunktes, beispielsweise Baseline, weitere Zeitpunkte wählen. Dies können beliebig viele sein. Gewählte Zeitpunkte können wieder abgewählt werden. Der Schritt kann beendet werden, wenn mindestens ein Zeitpunkt ausgewählt ist (Abbildung 4.6)

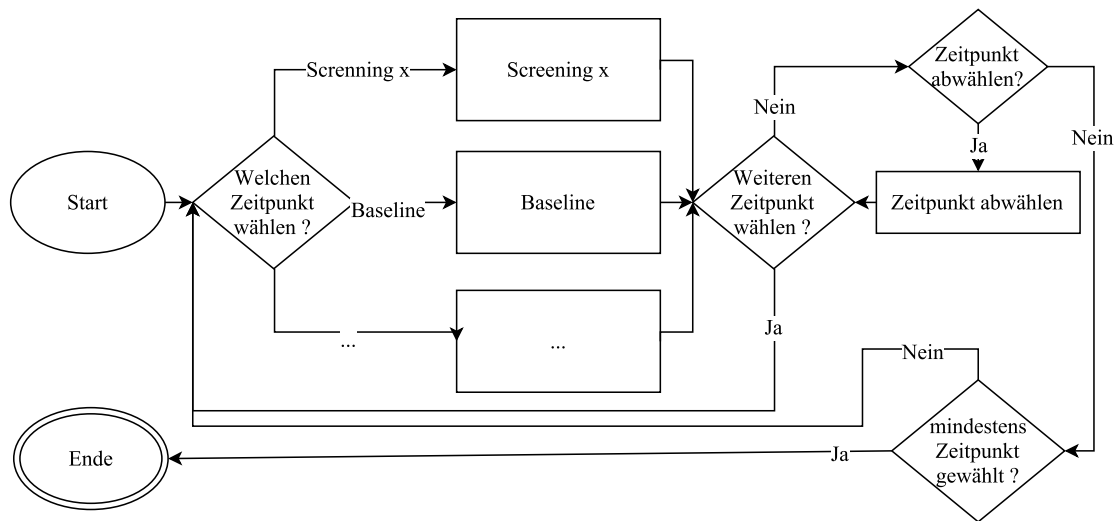


Abbildung 4.6: Flussdiagramm der Auswahl der Zeitpunkte einer LS

4.3.3 Statistische Gruppen erstellen

Das Erstellen einer SG basiert auf den funktionalen Anforderungen unter Kapitel 3.2.4 und ist in Abbildung 4.7 veranschaulicht.

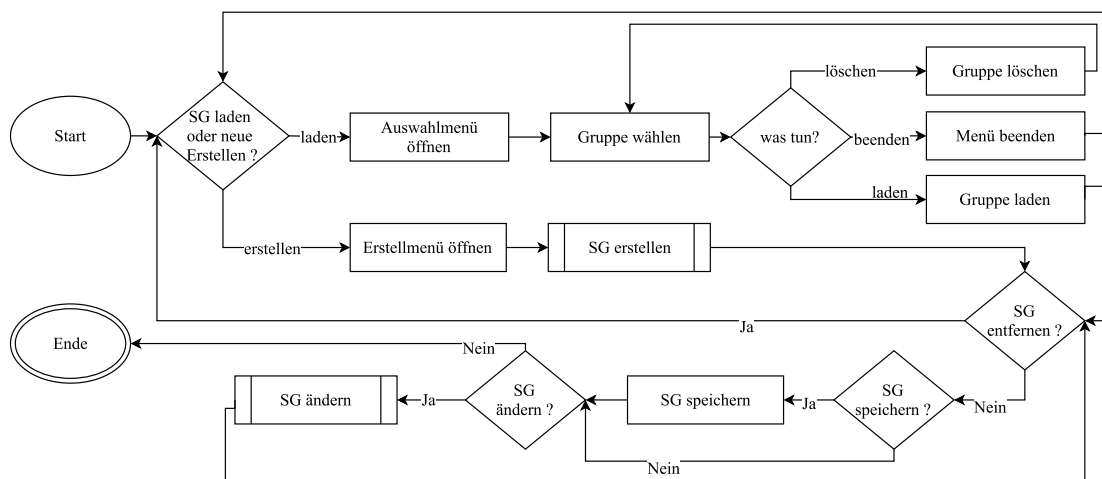


Abbildung 4.7: Flussdiagramm der Erstellung einer SG

Beim Erstellen unter der Wahl einer CS kann der Nutzer die Entscheidung treffen eine neue SG zu erstellen oder eine Gespeicherte zu laden. Soll eine SG geladen werden, so öffnet sich ein Auswahlmenü aller gespeicherten SGs aus welchen eine gewählt werden kann. Die gewählte SG kann geladen oder gelöscht werden. Ist sie gelöscht taucht sie im Auswahlmenü nicht mehr auf. Das Menü kann auch ohne Auswahl beendet werden. Das Erstellen einer neuen SG erfolgt in einem separaten Menü. Hier können die gewünschten Filter gesetzt werden und die SG anschließend erstellt werden. Die geladene oder erstellte SG wird in einer Übersicht mit allen Einstellungen der Filter angezeigt. Die Übersicht bietet Möglichkeiten die SG zu entfernen, zu

speichern oder sie zu ändern. Um den Schritt des Wizards zu beenden, muss genau eine SG erstellt worden sein.

Die Erstellung von einer SG unter der Wahl einer LS erfolgt ähnlich wie in Abbildung 4.7, mit der Änderung, dass der Prozess beliebig oft wiederholt werden kann. Um den Schritt des Wizards zu beenden, muss mindestens eine SG erstellt worden sein.

4.3.4 Kriterien auswählen

Die Auswahl der Kriterien im Statistikmodul basiert auf den funktionalen Anforderungen unter Kapitel 3.2.5. Der Nutzer kann unter den in Tabelle 2.3 gezeigten Kriterien wählen. Es können beliebig viele Kriterien selektiert werden. Gewählte Kriterien können wieder abgewählt werden. Zum Beenden des Schrittes muss mindestens ein Kriterium ausgewählt sein. Der Prozess ist in Abbildung 4.8 veranschaulicht.

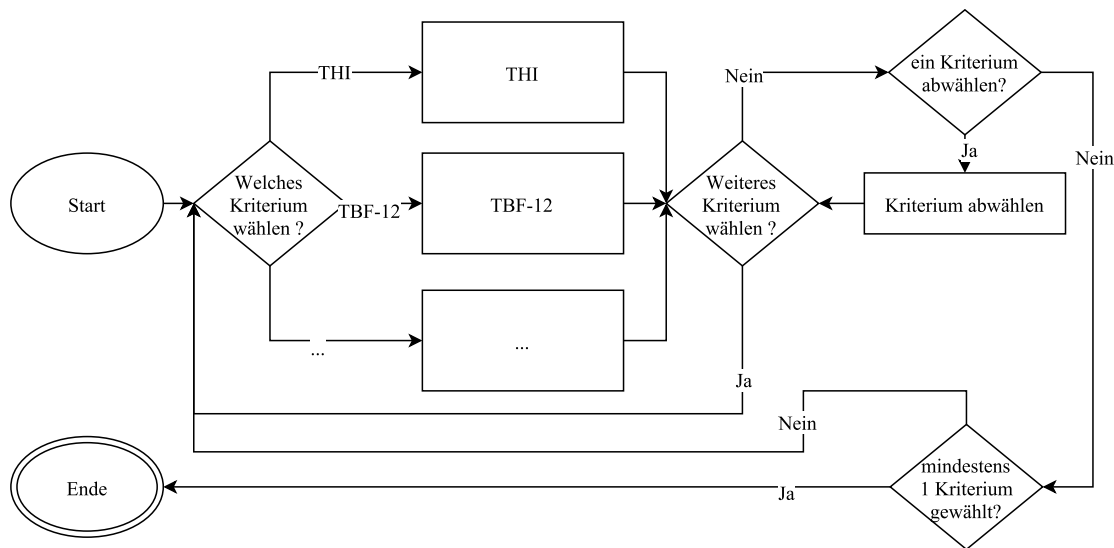


Abbildung 4.8: Flussdiagramm der Auswahl der Kriterien

4.4 Gestaltung des Wizards

Wie in Kapitel 3.3.1 erwähnt, soll der Nutzer klar und leicht durch den Erstellprozess einer Statistik geleitet werden. Deshalb wird der Wizard in vier Schritte unterteilt und unterstützt den in 4.3 geplanten Ablauf. Alle Schritte des Wizards sind nach dem gleichem Prinzip strukturiert. Der Kopfteil gibt Informationen über den aktuellen Schritt des Wizards und eine grobe Tätigkeitsbeschreibung. Im Auswahlteil wird angezeigt, zwischen welchen Elementen der Nutzer auswählen kann oder welche bereits ausgewählt sind. Darunter befindet sich der Navigationsteil mit dem der Nutzer zwischen einzelnen Schritten navigieren kann. Die Struktur wird in Abbildung 4.9 veranschaulicht.

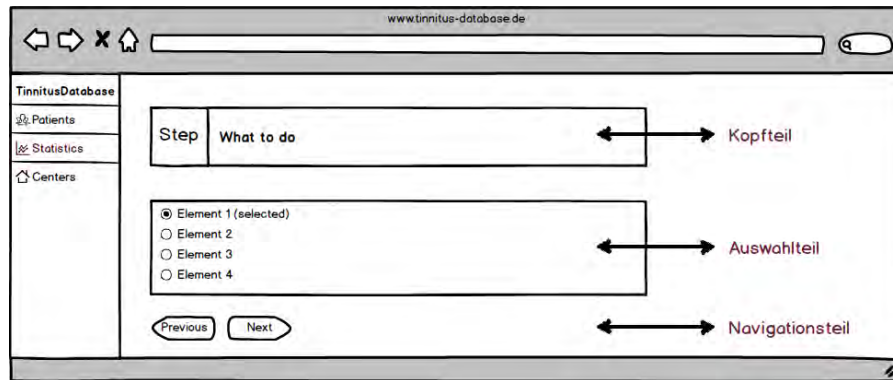


Abbildung 4.9: Struktur der Schritte des Wizards

4.4.1 Wahl der Statistikart

Bei der Wahl der Statistik kann der Nutzer zwischen einer CS oder einer LS wählen (Abbildung 4.10). Je nach Wahl verändern sich die nächsten Schritte im Wizard. Deshalb ist es wichtig, diese Wahl auch logisch durch einen Schritt im Wizard zu trennen. Die Gestaltung unterstützt den unter 4.3.1 geplanten Ablauf.

Abbildung 4.10: Wahl der Statistikart

4.4.2 Wahl der Zeitpunkte

In diesem Schritt des Wizards stehen dem Nutzer die in Kapitel 2.4.1 erwähnten Zeitpunkte zur Auswahl (Abbildung 4.11). Die Gestaltung unterstützt den in Kapitel 4.3.2 geplanten Ablauf. Bei einer LS können mehrere Zeitpunkte mit Hilfe von Checkboxes ausgewählt werden. Bei einer CS kann nur ein Radiobutton ausgewählt werden.

The screenshot shows a web browser window with the URL 'www.tinnitus-database.de'. On the left is a sidebar with 'TinnitusDatabase' and three menu items: 'Patients', 'Statistics', and 'Centers'. The main content area is titled '2 Select your timepoint'. It contains a list of radio buttons: 'Screening (selected)', 'Baseline', 'Visite', 'Final visite', and 'Follow Up'. At the bottom of the main area are 'Previous' and 'Next' buttons.

Abbildung 4.11: Wahl der Zeitpunkte

4.4.3 Erstellung der Gruppen

Der Erstellprozess von SGs unterstützt den in Kapitel 4.3.3 dargestellten Ablauf. Der Nutzer hat die Wahl eine neue SG zu erstellen oder eine gespeicherte zu laden (Abbildung 4.12). Diese wird ihm dann in einer Übersicht mit allen gewählten Filtern angezeigt und kann entfernt, gespeichert oder nochmal bearbeitet werden. Bei einer CS werden gegebenenfalls mehrere SGs mit Hilfe von mehreren Übersichten untereinander angezeigt.

The screenshot shows the '3 Select your groups' step. It includes 'Add group' and 'load group' buttons. Below them is a table titled 'All women over 30 years from Regensburg'. The table lists various attributes and their values, with some attributes having dropdown arrows. At the bottom are 'Previous' and 'Next' buttons.

All women over 30 years from Regensburg		
Sex	Female	TBF-12
Age	Over 30	CGI
Dropout	No	TQ
Psychiatric problems	-	Mini-TQ
Treatment	-	MDI
Center	Regensburg	BDI
Duration	-	Whoqol D1
No of treatments	-	Whoqol D2
Location	-	Whoqol D3
Variability	-	Whoqol D4
Somatic	-	Hyperacusis
THI	-	Hearingloss

Abbildung 4.12: Erstellung der Gruppen

4.4.4 Wahl der Kriterien

Dem Nutzer stehen die in Kapitel 2.4.4 erwähnten Kriterien zur Auswahl (Abbildung 4.13). Die Gestaltung unterstützt den in Kapitel 4.3.4 geplanten Ablauf.

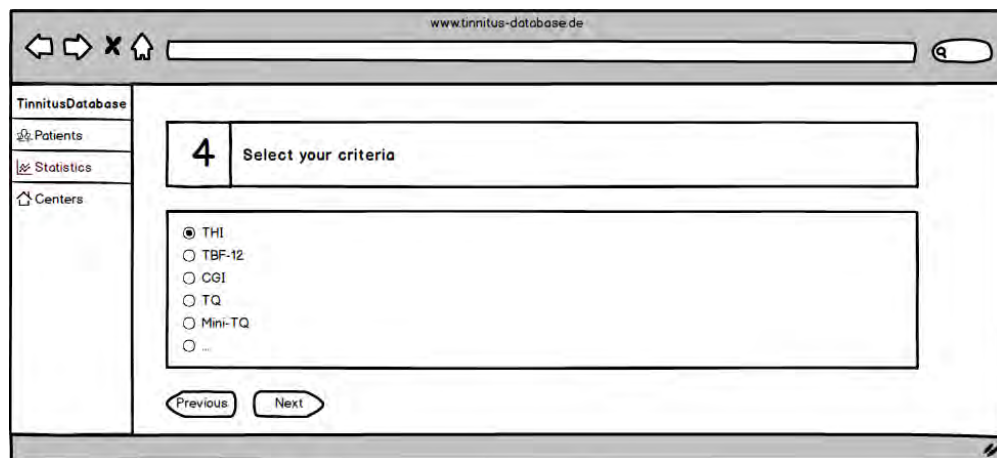


Abbildung 4.13: Wahl der Kriterien

4.5 Gestaltung der Auswertung

Die statistische Auswertung wird aus Übersichts- und Platzgründen getrennt. Es gibt Ansichten für die Auswertung von SGs und für die Anzeige von Individualdaten. Im Folgenden wird die Gestaltung beider Ansichten erläutert. Die Gestaltung der statistischen Auswertung erfolgt unter den in Kapitel 3.2.6 und 3.2.7 erhobenen funktionalen Anforderungen.

4.5.1 Auswertung der statistischen Gruppen

Die statistische Auswertung der SGs unter den gewählten Kriterien soll, wie bei der Erstellung, übersichtlich und verständlich sein. Deshalb wird für jedes Kriterium eine eigene Übersicht erstellt (Abbildung 4.14). Bei mehreren Kriterien werden diese untereinander angeordnet. In der Übersicht werden die verschiedenen SGs grafisch mit Hilfe eines Diagramms gegenüber gestellt und können so schnell verglichen werden. Unter dem Diagramm werden deskriptive Werte wie das Konfidenzintervall, der Mittelwert und die Standardabweichung zu den SGs angezeigt. Mit *Scores/Missing* wird dem Nutzer angezeigt, wie viele der Patienten der SG einen gültigen oder nicht gültigen Summenscore zu dem Fragebogen haben. Die Tabellendaten der deskriptiven Statistik können exportiert oder gedruckt werden. Jede Übersicht eines Kriteriums bietet die Möglichkeit die individuellen Daten, die zu der Auswertung geführt haben, per Knopfdruck einzusehen. Die Darstellung einer CS unterscheidet sich von einer LS, wie in 2.3 und 2.2 zu sehen ist.

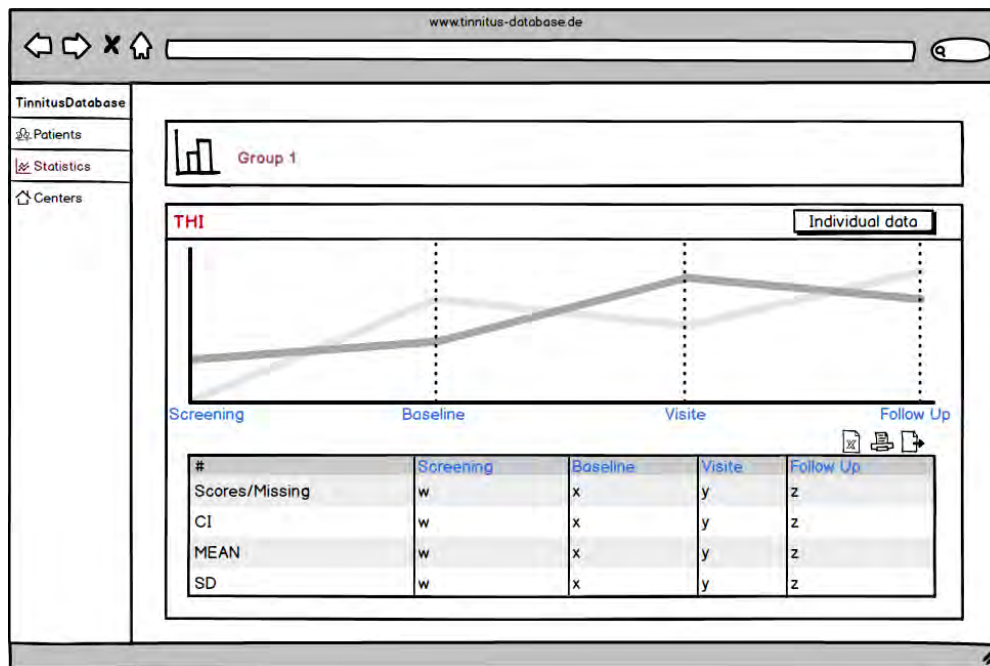


Abbildung 4.14: Statistische Auswertung am Beispiel einer LS

4.5.2 Individualdaten eines Kriteriums

Mit den Individualdaten eines Kriteriums werden dem Nutzer alle Daten, die bei der Auswertung berücksichtigt wurden, aufgelistet. Hierbei wird dem Nutzer zu jeder SG eine eigene Übersicht angezeigt, die die einzelnen Daten tabellarisch aufbereitet (Abbildung 4.15). Zusätzlich wird die ID des Patienten gezeigt der den Fragebogen ausgefüllt hat. Jede Tabelle kann einzeln exportiert oder gedruckt werden.

PatientID	THI	X	Y	Z
1234	-	-	-	-
1235	-	-	-	-

PatientID	THI	X	Y	Z
1234	-	-	-	-
1236	-	-	-	-
1237	-	-	-	-

Abbildung 4.15: Individuelle Daten am Beispiel einer LS

5 Prototypische Realisierung

Im Folgenden wird das unter Kapitel 4 konzipierte Statistikmodul realisiert. Dafür werden zunächst die für die Implementierung verwendeten Technologien und Frameworks erläutert und im Anschluss ausgewählte Aspekte der Implementierung vorgestellt.

5.1 Verwendete Technologien

Im diesem Abschnitt werden die bei der Implementierung verwendeten Technologien erläutert.

5.1.1 PHP

Das Erstellen und Auswerten von statistischen Gruppen wird auf dem Server mit *PHP* vorgenommen. PHP (ursprünglich Personal Home Page Tools) ist die, mit 81,8%, am häufigsten genutzte serverseitige Skriptsprache der Welt [34]. Sie wird hauptsächlich zur Erstellung von Webanwendungen und dynamischen Webseiten verwendet und zeichnet sich durch eine breite Datenbankunterstützung aus. Seit Version 5 wird zusätzlich die Objektorientierung unterstützt [42]. PHP-Code wird serverseitig verarbeitet. Das bedeutet, dass der Quelltext vom Client, als Anfrage, erst an den PHP-Interpreter auf einem Webserver übermittelt wird, bevor die Rückgabe des Interpreters als Antwort an den Browser übertragen wird (Abbildung 5.1).

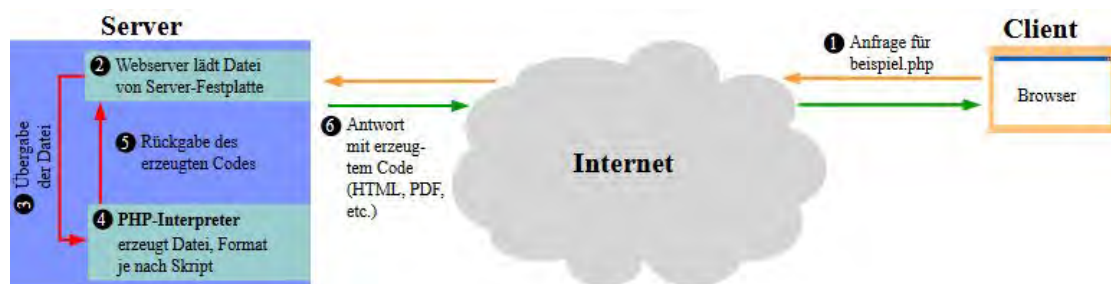


Abbildung 5.1: Funktionsweise von PHP [35]

5.1.2 JavaScript

Interaktionen mit dem Statistikmodul und die Erstellung der Diagramme werden mit *JavaScript* behandelt. JavaScript ist eine clientseitige Skriptsprache, die entwickelt wurde, um in Webbrowsern Benutzerinteraktionen auszuwerten, Inhalte zu erstellen, zu verändern oder nachzuladen und so die Möglichkeiten von HTML und CSS zu erweitern [38]. Sie wird auf 93,7% aller Webseiten genutzt [32]. Mittlerweile findet sie auch Anwendung auf Servern und Microcontrollern. In Javascript kann sowohl objektorientiert, als auch prozedural und funktional programmiert werden [38].

5.1.3 jQuery

jQuery ist die meistgenutzte JavaScript-Bibliothek und wurde 2006 vorgestellt. Sie stellt Funktionen bereit, um durch die DOM-Elemente zu navigieren und diese zu manipulieren. Die Bibliothek wird von der unabhängigen jQuery Foundation fortlaufend weiterentwickelt [39]. jQuery findet auf jeder zweiten Webseite Anwendung [31].

5.1.4 MySQL

Für die Speicherung der Daten der TD wird eine *MySQL* Datenbank verwendet. MySQL gehört zu den meist verbreitetsten relationalen Datenbankverwaltungssystem der Welt [4] und basiert auf Structured Query Language (SQL) [41].

5.2 Verwendete Frameworks

Im diesem Abschnitt werden die bei der Implementierung verwendeten Frameworks kurz vorgestellt.

5.2.1 Laravel

Die TD, die dem Statistikmodul zugrunde liegt, wurde mit *Laravel* entwickelt. Laravel ist ein Open-Source PHP Framework, das zum Entwickeln von Webanwendungen im Model-View-Controller Pattern dient und viele Vorteile und Erleichterungen bietet [40].

Für das Statistikmodul ist beispielsweise der *Query builder* interessant, da beim Erstellen einer Statistik viele Anfragen mit verschiedenen Parametern an die DB anfallen. Der Query builder ermöglicht es SQL Queries nicht direkt schreiben zu müssen, sondern Queries mit Hilfe von Klassen und Methoden programmatisch erstellen zu lassen [40].

Einen weiteren Vorteil bietet das *Eloquent ORM* (object-relational mapping), das Tabellen in der Datenbank als Klassen repräsentiert. Einzelne Instanzen dieser Klasse werden einzelnen Reihen in der Datenbank zugeordnet. Wird eine Instanz verändert oder gelöscht und die entsprechende Funktion der Klasse aufgerufen, so wird die ihr zugeordnete Reihe in der Datenbank ebenfalls verändert oder gelöscht. Beim Erstellen und anschließendem Speichern einer neuen Instanz wird eine neue Reihe in der Datenbank erzeugt [40].

Laravel bietet ebenfalls *Reverse Routing* an. Dabei wird eine URI aus der Route und deren Parametern generiert, so dass beim Verändern der Routen automatisch wieder funktionierende Links erzeugt werden. Laravel bietet viele weitere Features wie etwa *RESTful controllers*, *Class auto loading*, *Migrations* oder die Templating-Engine *Blade* [40].

5.2.2 Bootstrap

Zur Gestaltung der Bedienelemente wird *Bootstrap* verwendet. Bootstrap ist ein auf HTML, CSS und JavaScript basierendes Frontend Framework das Gestaltungsvorlagen für Typographie, Navigationselemente und Oberflächengestaltungselemente wie etwa Buttons, Tabellen, Grid-Systeme und viele weitere bietet. Es wurde von Twitter entwickelt und als Open-Source Projekt veröffentlicht [37]. Bootstrap wird mittlerweile in 13,4% aller Webseiten verwendet [33].

5.2.3 AdminLTE

AdminLTE ist eine OpenSource Sammlung von Gestaltungs- und Bedienelementen, basierend auf Bootstrap 3 [1] und wird ebenfalls zur Gestaltung des Statistikmoduls verwendet.

5.2.4 Chart.js

Für die grafische Unterstützung des Statistikmoduls wird *Chart.js* verwendet. Chart.js ist eine Open-Source Javascript Bibliothek mit dem responsive Diagramme erzeugt werden können. Es bietet 8 verschiedene Grunddiagrammtypen (Line-, Bar-, Radar-, Polar Area-, Pie-, Doughnot-, Bubble Chart and Scales), die jeder für sich einzeln als Modul geladen und angepasst werden können. Da Chart.js das HTML 5 canvas Element zum Rendern der Diagramme nutzt läuft es in allen gängigen Browsern (Firefox, Chrome, Internet Explorer, Safari) [3].

5.3 Ausgewählte Implementierungsaspekte

In diesem Kapitel werden ausgewählte Teile der Implementierung vorgestellt. Hierbei stehen die Erfassung der Nutzereingaben und die daraus resultierenden Verarbeitungsschritte dieser Daten bis zur Anzeige der Statistik im Vordergrund. Zunächst wird in Kapitel 5.3.1 darauf eingegangen, wie Zeitpunkte, verschiedene SGs und Kriterien gespeichert und übertragen werden.

Anschließend wird in Kapitel 5.3.2 gezeigt, wie anhand dieser Eingaben auf die DB zugegriffen und geeignete Daten gefunden werden. Im Kapitel 5.3.3 wird betrachtet, wie mit Hilfe der gefundenen Daten die Statistik erstellt wird. Im folgenden Kapitel 5.3.4 steht die Darstellung der berechneten Statistik im Vordergrund. Abschließend wird in Kapitel 5.3.5 die grafische Umsetzung des Wizards und der Auswertung erläutert.

5.3.1 Nutzereingaben erfassen

Der Nutzer gibt Kriterien, Zeitpunkte, SGs und die Art der Statistik über einen Wizard ein (Kapitel 4.3). Nach dem Beenden eines Schrittes werden die getätigten Eingaben gespeichert. Dafür werden JavaScript Objekte verwendet, die in der Literalschreibweise vordefiniert wurden. Für jeden Schritt im Wizard gibt es ein JavaScript Objekt. Diese JavaScript Objekte bieten den Vorteil, dass sie einfach über die Methode `JSON.stringify(object)` zu einem JSON-String konvertiert und so übertragen werden können. Im Folgenden wird der Vorgang am Beispiel der Erstellung einer SG verdeutlicht.

Um die Eingaben des Nutzers im Wizard für das Erstellen einer SG zu speichern ist ein `groupDefinition` Objekt in der Literalschreibweise definiert (Listing 5.1). Dieses Objekt enthält verschiedene Eigenschaften, wie `name`, `sex`, `thi`, `tbfi` und weitere. Diese Eigenschaften repräsentieren die Filter nach denen ein Nutzer eine SG erstellen kann (Kapitel 2.4.5). Eine besondere Rolle spielen hierbei die numerischen Filter, da sie größere Zahlenbereiche abdecken können. Hierfür wird den betreffenden Eigenschaften der `groupDefinition` das Unterobjekt `complexRangeDefinition` zugewiesen (Listing 5.1, Zeile 5-6). Mit diesem können Zahlenbereiche abgedeckt werden. Die Eigenschaft `type` repräsentiert dabei, ob es sich um Zahlen *größer als*, *kleiner als* oder *einen Bereich zwischen zwei Zahlen* handelt. Die Eigenschaften `x` und `y` halten die entsprechenden Grenzen.

```
1 var groupDefinition = {  
2   name: '',  
3   sex: '',  
4   ...  
5   thi: complexRangeDefinition,  
6   tbfi: complexRangeDefinition,  
7   ...  
8 };  
9 var complexRangeDefinition = {  
10  type: 0,  
11  x: 0,  
12  y: 0  
13 };
```

Listing 5.1: Definition des `groupDefinition` Objekts

Nach dem Erstellen der SG im Wizard wird ein neues Objekt von `groupDefinition` erzeugt (Listings 5.2, Zeile 4). Die Werte der Filter werden den entsprechenden Eigenschaften der `group` in der `writeValuesIntoGroup(...)` Methode zugewiesen (Listing 5.2, Zeile 6 und

Listing 5.3, Zeile 2). Anschließend wird das Objekt dem Array `statisticGroups` hinzugefügt, das alle erstellten SGs speichert (Listings 5.2, Zeile 8).

```

1  var statisticGroups = new Array();
2
3  function addNewGroup(){
4      var group = Object.create(groupDefinition);
5      ...
6      writeValuesIntoGroup(group, name);
7      ...
8      statisticGroups.push(group);
9  }

```

Listing 5.2: Erstellen einer neuen SG

Am Beispiel eines numerischen Filters wird veranschaulicht, wie die Filter den entsprechenden Eigenschaften zugewiesen werden (Listing 5.3). Der Methode `getSelectedComplexRange` wird die ID des HTML DOM Elements übergeben, das die Information des gesetzten Filters enthält. Mit dieser ID kann das Element selektiert und die Typinformation (hier `option`), sowie die Werte der Grenzen ausgelesen werden. Je nach Typ werden dann die Eigenschaften des Unterobjekts `complexRange` gesetzt (Listing 5.3, Zeile 11-21). Anschließend wird dieses Unterobjekt zurückgegeben und der Eigenschaft der SG, hier `group.thi`, zugeordnet (Listing 5.3, Zeile 2 und 22). Bei nicht numerischen Filtern wird der Wert des entsprechenden HTML DOM Elements einfach ausgelesen und der SG hinzugefügt.

```

1  function writeValuesIntoGroup(group, name){
2      group.thi = getSelectedComplexRange('THI-select');
3      ...
4  }
5
6  function getSelectedComplexRange(id){
7      if(document.getElementById(id)!=null){
8          var select = document.getElementById(id);
9          var option = select.options[select.selectedIndex].text;
10         var complexRange = Object.create(complexRangeDefinition);
11         if(option == "All"){
12             ...
13         }else if(option == "Less than x"){
14             complexRange.type = 1;
15             complexRange.x = document.getElementById(id+"-value-x").value;
16             complexRange.y = -1;
17         }else if(option == "More than x"){
18             ...
19         }else{ // between x and y
20             ...
21         }
22         return complexRange;
23     }
24     return null;
25 }

```

Listing 5.3: Zuweisung eines numerischen Filters an die entsprechende Eigenschaft

Für die Übertragung der SGs wird dem HTML DOM ein neues Formular und ein Element vom Typ `hidden` angefügt (Listing 5.4). Dem Element wird als Wert der JSON String der SGs hinzugefügt und das Formular anschließend mit `form.submit()` via HTTP POST Methode übertragen. Die Javascript Objekte der anderen Schritte des Wizards werden analog übertragen.

```
1 function send(){
2     var form = document.createElement("form");
3     form.setAttribute("method", "post");
4     form.setAttribute("action", window.location.origin+'../../');
5     var hiddenGroups = document.createElement("input");
6     hiddenGroups.setAttribute("type", "hidden");
7     hiddenGroups.setAttribute("value", JSON.stringify(StatisticGroups));
8     form.appendChild(hiddenGroups);
9     document.body.appendChild(form);
10    form.submit();
11 }
```

Listing 5.4: Übertragen der SG

5.3.2 Finden geeigneter Daten

Nach Absetzen der Anfrage mit Übertragung der Nutzereingaben (Kapitel 5.3.1) müssen geeignete Patienten für die gesetzten Filter der SGs gefunden werden. Dafür werden serverseitig Anfragen an eine MySQL Datenbank gestellt. Diese werden im Folgenden beschrieben.

Da die Anfrage serverseitig bearbeitet wird, finden Zugriffe auf der Datenbank in PHP statt. In der Klasse `StatisticController` werden zunächst die übertragenen POST Variablen mit `json_decode()` von JSON-Strings in PHP-Variablen konvertiert (Listing 5.5, Zeile 4-5). Die Variable `statistic_id` hält die Information, ob der Nutzer eine LS oder CS möchte. Beide Fälle müssen voneinander unterschieden und anders behandelt werden (Kapitel 2.4.6 und 2.4.7). Im weiteren Verlauf wird das Vorgehen bei einer CS betrachtet.

```
1 class StatisticController extends BaseController{
2     public function show(){
3         ...
4         $groupsJSON = $_POST['groupsAsJSONInput'];
5         $groups = json_decode($groupsJSON, true);
6         // analog fuer $criteria, $timepoints und $statistic_id
7         if($statistic_id=="longitudinal"){
8             return $this -> longitudinalStatistics($criteria, $timepoints, $groups);
9         }else{
10            return $this -> crossSectionalStatistics($criteria, $timepoints, $groups);
11        }
12    }
13 }
```

Listing 5.5: Unterscheidung von CS und LS

Da es sich bei der Statistik um eine CS handelt, wird über die Anzahl der `groups` iteriert (Listing 5.6, Zeile 2). Nun wird für jede SG zum gewünschten Zeitpunkt eine Methode aufgerufen, die dann den spezifischen Datenbankaufruf vornimmt. Diese Methoden unterscheiden sich jedoch, da beispielsweise nicht zu jedem Zeitpunkt alle Fragebögen ausgefüllt sind (Kapitel 2.4.1) und somit nicht jederzeit jeder Filter erfüllt werden kann. Gesetzte Filter, die nicht erfüllt werden können, hätten immer zur Folge, dass kein Patient gefunden wird. Um das zu vermeiden, wird vor dem Methodenaufruf unterschieden (Listing 5.6, Zeile 7-9). Da Patienten zu spezifischen Behandlungszeitpunkten zu finden sind, wird nach der `session_content_id` gesucht. Diese repräsentiert eine Behandlung zu einem bestimmten Zeitpunkt und kann über Fremdschlüsselbeziehungen eindeutig einem Patienten zugeordnet werden (Kapitel 4.2). Die Liste der `session_content_id`'s, die alle gesetzten Filter der aktuellen SG erfüllen, wird von der aufgerufenen Methode zurückgegeben und in `relevant_keys` gespeichert. Dieses Vorgehen wird für alle SGs wiederholt.

```

1 private function crossSectionalStatistics($criteria, $timepoints, $groups){
2     for($current = 0; $current < count($groups); $current++){
3         ...
4         // $searching_session_content beinhaltet die Information des Zeitpunktes
5         // gemapped auf die zuständige Tabelle in der DB
6         $relevant_keys;
7         if($searching_session_content == 'session_content_screening'){
8             $relevant_keys = $this -> searchScreening($groups, $searching_session_content,
9                 $current, $pos);
10        }else if{...}
11        ...
12    }
13 }

```

Listing 5.6: Methodenaufruf für das Erstellen einer CS

Stellvertretend für die verschiedenen Methoden wird die Methode `searchScreening` vorgestellt (Listing 5.7), die passende `session_content_id`'s zu den Filtern der SG zum Zeitpunkt Screening findet und zurückgibt. Der SQL Query für die Anfrage an die DB ist nicht statisch festgelegt, sondern wird programmatisch mit Hilfe des Laravel Query builder beim Aufruf der Methode erstellt. Ein statischer Query hätte zur Folge, dass für Änderungen eventuell neue Queries festgelegt werden müssten, was schnell unübersichtlich wird und nur schwer wartbar ist. Um die gesetzten Filter überprüfen zu können, müssen verschiedene DB Tabellen mit dem `SQL JOIN` Anweisung zusammengefügt werden (Listing 5.7, Zeile 3-10). Es wird darauf geachtet nur die für die Filter spezifischen Spalten der Tabellen in die `SQL JOIN` Anweisung miteinzubeziehen (Listing 5.7, Zeile 3-10, rot markierte Spaltennamen), um sehr große Tabellen zu vermeiden und somit Zeit bei der Berechnung einzusparen. Um `WHERE` Anweisungen nur bei gesetzten Filtern hinzuzufügen, gibt es die Möglichkeit *Advanced Wheres* zu erstellen. *Advanced Wheres* können Bedingungen für das Erstellen hinzugefügt werden. In Listing 5.7, Zeile 12-17, wird beispielsweise dem Query nur dann die `WHERE` Anweisung hinzugefügt, wenn ein Filter für das Zentrum gesetzt ist, also nicht dem neutralen 'All' entspricht. Die `WHERE` Anweisungen für alle restlichen Filter sind analog erstellt, sodass beim Erstellen des Query vom Query builder nur die notwendigen `WHERE` Anweisungen angehängt werden. Nach dem Beenden des Auf-

rufs werden die gefundenen *session_content_id*'s in *session_content_ids* gespeichert und zurückgegeben.

```

1 private function searchScreening($groups, $searching_session_content, $current, $pos){
2     $session_content_ids = DB::table($searching_session_content)
3     ->join('sessions', 'sessions.session_id', '=', $searching_session_content.'.
        session_id')
4     ->join('patient_records', 'patient_records.patient_record_id', '=', 'sessions.
        patient_record_id')
5     ->join('patients', 'patients.patient_id', '=', 'patient_records.patient_id')
6     ->join('id_translation', 'patients.patient_id', '=', 'id_translation.patient_id')
7     ->join('centers', 'centers.center_id', '=', 'id_translation.center_id')
8     ->join('session_content_description', 'session_content_description.session_id', '='
        , 'sessions.session_id')
9     ->join(DB::raw('(Select tschq.session_content_id, tschq.q35_psychological_treatment
        , tschq.q11_daily_volume, tschq.q21_head_change, tschq.q9_perception, tschq.
        q5_begin_tinnitus, tschq.q28_noise_sensitive FROM tschq) tempTschq'), '
        tempTschq.session_content_id', '=', 'session_content_screening.
        session_content_id')
10    ->join(DB::raw('(select audiological_examination.session_content_id,
        audiological_examination.left_hearing_loss, audiological_examination.
        right_hearing_loss from audiological_examination) audilogic'),'audiologic.
        session_content_id', '=', $searching_session_content.'.session_content_id' )
11    //Zentrum
12    ->where(function($query) use ($groups, $current){
13        $scen = $groups[$current]['center'];
14        if($scen != 'All'){
15            $query -> where('centers.center_name', '=', $scen);
16        }
17    })
18    ...
19    ->select($searching_session_content.'.session_content_id')
20    ->get();
21    return $session_content_ids;
22 }

```

Listing 5.7: DB Anfrage für den Zeitpunkt Screening

5.3.3 Berechnung der Statistik

Mit den in Kapitel 5.3.2 gefundenen *session_content_id*'s kann nun zu jeder *session_content_id* die Summenscores der Fragebögen angefragt werden. Die Summenscores werden bereits nach dem Speichern der Fragebögen vom System automatisch berechnet und in die Tabelle *session_content_score* überführt. Um die Summenscores für eine SG zu erhalten, wird die Methode `getGroupScores(...)` aufgerufen (Listing 5.8). Diese sucht in der Tabelle *session_content_score* nach Einträgen zu jeder *session_content_id*, die sich in dem übergebenen Array *session_content_ids* befindet (Listing 5.8, Zeile 4) und gibt diese zurück.

```

1 private function getGroupScores($session_content_ids, $searching_session_content){
2   $scores = DB::table("session_content_score")
3     ->where('type_name','=', $searching_session_content)
4     ->whereIn('session_content_id', $session_content_ids)
5     ->select('thi','tbfl2','tq','mini_tq','cgi','bdi','mdi','whoqol_d1','whoqol_d2','
        whoqol_d3','whoqol_d4')
6     ->get();
7   return $scores;
8 }

```

Listing 5.8: Anfordern aller Summenscores einer SG

Nach Erhalt der Summenscores einer SG müssen diese nun statistisch ausgewertet werden. Für die Auswertung wurde die Klasse `DescStatModel_Type_Numeric` implementiert (Listing 5.9), die beim Hinzufügen eines Scores mit der `addValue(...)` Methode (Listing 5.9, Zeile 7) automatisch statistische Werte wie Varianz, Standardabweichung, Minimum und Maximum berechnet. Weitere Werte wie das Konfidenzintervall werden beim Aufruf des jeweiligen *Getters* berechnet (Listing 5.9, Zeile 25-33).

```

1 class DescStatModel_Type_Numeric {
2   ...
3   private $standard_deviation = 0;
4   private $variance = 0;
5   private $confidence = 0;
6
7   public function addValue($value) {
8     // -1 ist score = missing
9     if($value != -1){
10      ...
11      $this -> variance = $this -> calcVarianz();
12      $this -> standard_deviation = $this -> calcStandardDeviation();
13    }
14  }
15  private function calcVarianz(){
16    $v = 0;
17    for($i = 0; $i < $this -> valid_input; $i++){
18      $v = $v + pow($this -> input[$i] - $this -> average, 2);
19    }
20    return $v / $this -> valid_input;
21  }
22  private function calcStandardDeviation(){
23    return sqrt($this -> variance);
24  }
25  public function getConfidenceInterval(){
26    if($this -> valid_input > 0){
27      $n = sqrt($this -> valid_input);
28      $sd = $this -> standard_deviation;
29      $ci = 1.96 * $sd / $n;
30      return number_format($ci,2);
31    }
32    return number_format(0,2);
33  }
34  ...
35 }

```

Listing 5.9: Die Klasse für die statistische Auswertung

Für jedes Kriterium ist eine Klasse implementiert, die `DescStatModel_Type_Numeric` erweitert (Listing 5.10). Somit kann jedes Kriterium für sich ausgewertet werden. Da jede SG die gleichen Kriterien hat, müssen diese voneinander getrennt werden, um sie voneinander unter-

scheiden zu können. Dafür wurde die Klasse `DescStatGroup` implementiert (Listing 5.10). Sie repräsentiert die Auswertung einer SG und hält die für sich spezifischen Kriterien. Da die Kriterien mit `public` deklariert sind, kann die Methode `addValue(...)` von außerhalb gerufen und Werte hinzugefügt werden.

```

1 class DescStatGroup {
2   public $name;
3   public $tbtf;
4   public $thi;
5   ...
6   public function init($name){
7     $this -> name = $name;           // Name der SG
8     $this -> tbtf = new Tbf_desc;    // Erweitert DescStatModel_Type_Numeric
9     $this -> thi = new Thi_desc;    // Erweitert DescStatModel_Type_Numeric
10    ...
11  }
12 }

```

Listing 5.10: `DescStatGroup` enthält alle statistischen Auswertungen einer SG

Wird Listing 5.6 um die statistische Auswertung erweitert, sieht die Funktion wie folgt aus (Listing 5.11). Nach dem Finden der `session_content_id`'s (Kapitel 5.3.2) wird eine neue `DescStatGroup` `group` angelegt und initialisiert (Listing 5.11, Zeile 13-14). Nun werden mit der Methode `getGroupScores(...)` die Summenscores der Fragebögen für alle `session_content_id`'s aus der DB geholt (Listing 5.11, Zeile 15). Anschließend werden diese den Kriterien der `group` hinzugefügt und automatisch statistisch verwertet (Listing 5.11, Zeile 16-19). Sind alle Summenscores verarbeitet wird die `group` dem Array `statisticalGroups` hinzugefügt (Listing 5.11, Zeile 20). Dieses enthält letztlich die Auswertung der CS.

```

1 private function crossSectionalStatistics($criteria, $timepoints, $groups){
2   $statisticalGroups = array();
3   ...
4   for($current = 0; $current < count($groups); $current++){
5     ...
6     // $searching_session_content beinhaltet die Information des Zeitpunktes
7     // gemapped auf die zuständige Tabelle in der DB
8     $relevant_keys;
9     if($searching_session_content == 'session_content_screening'){
10      $relevant_keys = $this -> searchScreening($groups, $searching_session_content,
11        $current, $pos);
12    }else if{...}
13    ...
14    $group = new DescStatGroup;
15    $group -> init($groups[$current]['name']); // Name der SG
16    $scores = $this -> getGroupScores($relevant_keys, $searching_session_content);
17    foreach($scores as $score){
18      $group -> thi -> addValue($score -> thi);
19    }
20    array_push($statisticalGroups, $group);
21  }
22  ...
23 }

```

Listing 5.11: Statistische Auswertung einer CS

5.3.4 Anzeigen der Statistik

Die Auswertung wird nun vom Server als HTTP Response an den Client gesendet. Im Client wird diese mit Hilfe von HTML und JavaScript grafisch aufbereitet und dargestellt. Zur Darstellung der Charts wird die JavaScript Bibliothek Chart.js verwendet (Kapitel 5.2.4). Da die Daten der HTTP Response als PHP Objekte übertragen werden, müssen diese erst aufbereitet und in JavaScript konvertiert werden, um sie mit Chart.js nutzen zu können. Dazu werden beim Erzeugen der HTML Seite JavaScript Variablen erzeugt. Das PHP Array `groups` ist das übertragene Array *statisticalGroups* aus Listing 5.11. Es enthält Objekte der Klasse *DescStatGroup*, die weitere Objekte der Klasse *DescStatModel_Type_Numeric* enthalten. Daher werden alle für die Statistik wichtigen Werte ausgelesen und in ein neues konvertierfähiges Array `groupData` geschrieben (Listing 5.12, Zeile 1-14). Dieses und weitere PHP Objekte werden nun mit `json_encode(...)` in JSON konvertiert und den neuen Variablen zugewiesen (Listing 5.12, Zeile 16-19).

```

1  <?php $groupData = array(); ?>
2  @foreach($groups as $group)
3  @foreach($group as $property)
4  @if($property instanceof DescStatModel_Type_Numeric)
5  @if($property -> wasUsed())
6  <?php $groupData[$group -> name][$property -> getPropertyName().'_mean'] = $property ->
   getAverage(); ?>
7  <?php $groupData[$group -> name][$property -> getPropertyName().'_ci'] = $property ->
   getConfidenceInterval(); ?>
8  @else
9  <?php $groupData[$group -> name][$property -> getPropertyName().'_mean'] = 0; ?>
10 <?php $groupData[$group -> name][$property -> getPropertyName().'_ci'] = 0; ?>
11 @endif
12 @endif
13 @endforeach
14 @endforeach
15
16 <script>
17     var groups = <?php echo json_encode($groupData); ?>;
18     ...
19 </script>

```

Listing 5.12: Umwandlung der PHP Daten in JSON

Beim Erzeugen des HTML Dokuments wird für jedes Kriterium ein eigenes Element `box` von Admin LTE für die Auswertung erstellt (Listing 5.13, Zeile 1-2). Dieses enthält ein `canvas` Element auf dem das jeweilige Chart gezeichnet wird und ein `table` Element in dem die Daten tabellarisch aufbereitet werden (Listing 5.13, Zeile 5-6).

```

1  @for($index = 0; count($criteria) > $index; $index++)
2  <div class="box box-primary">
3      <div class="box-header with-border"> /* Titel etc. */ </div>
4      <div class="box-body">
5          <canvas id = "{${$criteria[$index]}}-cross-sectional-canvas"></canvas>
6          <table class="table table-condensed"> /* Tabellendaten */ </table>
7      </div>
8  </div>
9  @endfor

```

Listing 5.13: Erzeugen des HTML Dokuments für die statistische Auswertung

Um Charts auf dem *canvas* zu zeichnen, wird beim Laden des JavaScript Dokuments die `initCharts()` Methode aufgerufen (Listing 5.14). Um für jedes Kriterium ein Chart zu erstellen, wird über `criteria.length` iteriert (Listing 5.14, Zeile 3). Der für das jeweilige Kriterium erstellte *canvas* wird mit `document.getElementById(...)` selektiert (Listing 5.14, Zeile 4). Anschließend wird das Chart mit Methoden der Chart.js Bibliothek erzeugt und dem Kontext des *canvas* zugewiesen (Listing 5.14, Zeile 6-7). Weitere Einstellungen am Chart, wie etwa das Deaktivieren der Animation, können beim Erstellen mit angegeben werden (Listing 5.14, Zeile 8-12).

```
1 function initCharts(){
2     if(mode == 'crossSectional'){
3         for(var index = 0; index < criteria.length; index++){
4             var c = document.getElementById(criteria[index]+'-cross-sectional-canvas');
5             ...
6             var ctx = c.getContext('2d');
7             barChart = new Chart(ctx).Bar(getChart(criteria[index], groups),{
8                 scaleShowVerticalLines: true,
9                 scaleGridLineColor : "rgba(0,0,0,.05)",
10                 responsive:true,
11                 scaleBeginAtZero: true,
12                 animation: false
13             });
14         }
15     }else{ //longitudinal
16         ...
17     }
18 }
19 initCharts();
```

Listing 5.14: Erstellen eines Charts mit Chart.js

5.3.5 Grafische Umsetzung des Statistikmoduls

Die grafische Umsetzung des Statistikmoduls orientiert sich an dem in Kapitel 4.3 definierten Erstellprozess und den in Kapitel 4.4 und 4.5 erstellten MockUps.

Beim Erstellen einer Statistik steht vor allem das Selektieren von SGs im Blickpunkt. Beim Selektieren (Abbildung 5.2, Punkt 1) oder Editieren einer SG (Abbildung 5.2, Punkt 4) öffnet sich dafür ein modaler Dialog mit allen zur Auswahl stehenden Filtern (Abbildung 5.2, Punkt 2). Diese Filter kann der Nutzer verändern und der SG einen Namen geben. Mit einer Bestätigung wird der Dialog beendet und die SG mit ihrem Namen und allen Filtern in einer Übersicht angezeigt (Abbildung 5.2, Punkt 3). Die SG kann mit einem Klick auf die nebenstehenden Icons editiert, gespeichert oder wieder entfernt werden (Abbildung 5.2, Punkt 4). Vorher gespeicherte SGs können mit einem Klick auf *Load group* (Abbildung 5.2, Punkt 5) sofort geladen werden. Ist der Nutzer mit der Auswahl zufrieden, kann er zum nächsten Schritt des Wizards übergehen (Abbildung 5.2, Punkt 6). Ebenfalls kann er zu einem der vorigen Schritte zurückkehren und Änderungen vornehmen. Die in diesem Schritt ausgewählten SGs, werden nur zurückgesetzt wenn der Nutzer die Art der Statistik von longitudinal zu crosssectional oder andersrum ändert.

The screenshot shows the 'Statistic' software interface. The left sidebar contains a 'STEP 3 Select your groups' section with buttons for 'Add group', 'Load group', 'Previous', and 'Next'. The main content area is titled 'Select a group' and contains a form with various filters: Name (Frauen, 20-50, Regensburg), Include dropouts (No), Sex (Both, Female, Male), Psychiatric problems (No declaration, Yes, No), Tinnitus variability (No declaration, Yes, No), Somatic tinnitus (No declaration, Yes, No), Hearing loss (All, Normal, IHL (>60 dB HL), OHL (21-60 dB HL)), Treatment (All), Center (Regensburg), Number of treatments (None), and Tinnitus location (No declaratio). A red number '2' is placed next to the 'Psychiatric problems' filter. Below this, the 'Generation' section shows a list of groups, with a red number '3' next to the 'Frauen, 20-50, Regensburg' group. A red number '4' is placed next to the 'Add group' button. A red number '5' is placed next to the 'Load group' button. A red number '6' is placed next to the 'Previous' button. A red number '1' is placed next to the 'Add group' button. A red number '2' is placed next to the 'Psychiatric problems' filter. A red number '3' is placed next to the 'Frauen, 20-50, Regensburg' group. A red number '4' is placed next to the 'Add group' button. A red number '5' is placed next to the 'Load group' button. A red number '6' is placed next to the 'Previous' button.

Abbildung 5.2: Erstellung einer SG im Wizards

In der Statistik werden dem Nutzer zu jedem Kriterium Auswertungen angezeigt (Abbildung 5.3, Punkt 1). Die Übersicht enthält deskriptive Statistiken wie Mittelwert, Standardabweichung und Konfidenzintervall zu jeder SG (Abbildung 5.3, Punkt 2). Weiterhin zeigt *SCORES/MISSING* dem Nutzer an, wie viele der gefundenen Fragebögen gültig und ungültig waren. Die Werte werden zusätzlich durch ein Chart veranschaulicht (Abbildung 5.3, Punkt 3) und können gedruckt, kopiert und als CSV, PDF oder Excel Datei exportiert werden (Abbildung 5.3, Punkt 4). Mit einem Klick auf *individual data* kann der Nutzer individuelle Daten einsehen (Abbildung 5.3, Punkt 5).

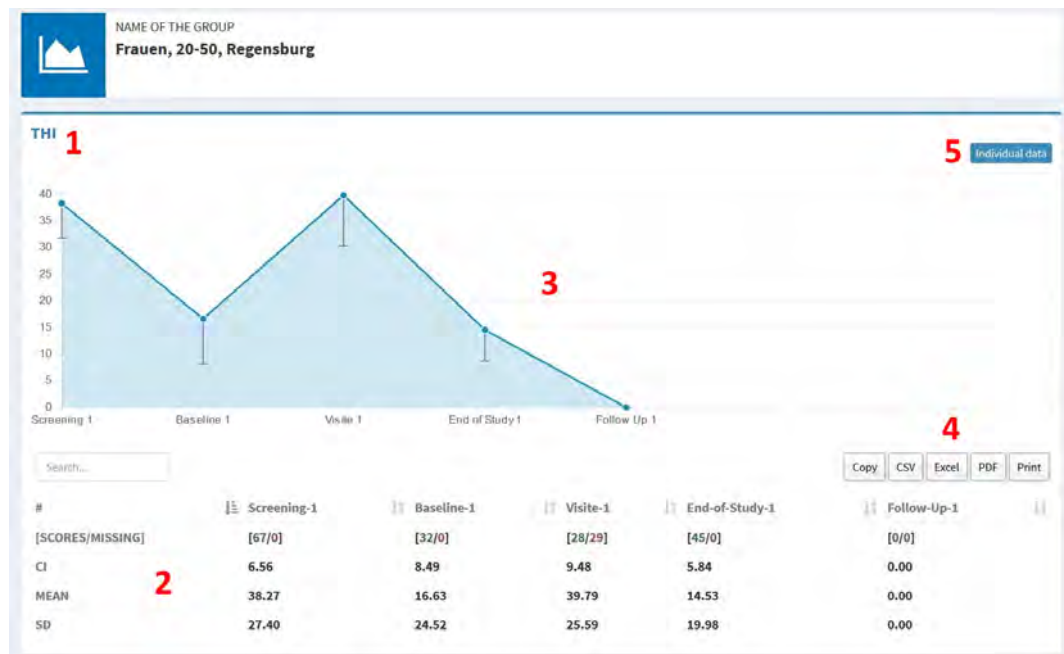


Abbildung 5.3: Ansicht der Statistik

Die individuellen Daten eines Kriteriums listen alle Summenscores auf, die bei der Auswertung berücksichtigt wurden (Abbildung 5.4, Punkt 1). Zusätzlich wird die zugehörige PatientenID und alle weiteren Summenscores des Patienten zum selben Zeitpunkt angezeigt. Die Daten können gedruckt, kopiert und als CSV, PDF oder Excel Datei exportiert werden (Abbildung 5.4, Punkt 2).

Statistic Data table

FRAUEN, 20-50, REGENSBURG
THI

Screening 1

Search...

2 Copy CSV Excel PDF Print

Patient	THI	TBF-12	CGI	TQ	Mini-TQ	BDI	MDI	WHOQOL_D1	WHOQOL_D2	WHOQOL_D3	WHOQOL_D4
38	54	12	-1	-1	0	8	0	-1	-1	-1	-1
257	20	0	16	6	0	1	0	0	0	0	0
262	72	0	46	15	0	10	0	0	0	0	0
269	84	0	61	20	0	33	0	0	0	0	0
276	54	15	56	17	0	20	0	-1	-1	-1	-1
398	38	0	35	10	0	24	0	0	0	0	0
555	56	0	61	20	0	15	0	0	0	0	0
629	56	0	40	14	0	12	0	0	0	0	0
684	42	0	44	12	0	7	0	0	0	0	0
765	24	0	15	6	0	11	0	0	0	0	0

Abbildung 5.4: Ansicht der Individualdaten der Statistik

6 Zusammenfassung und Ausblick

Abschließend werden in diesem Kapitel die vorgestellten Inhalte zusammengefasst und Ansatzpunkte für die zukünftige Weiterentwicklung des Statistikmoduls in einem Ausblick gegeben.

6.1 Zusammenfassung

Diese Arbeit hat sich zum Ziel gesetzt, ein Statistikmodul für die Tinnitus Datenbank zu entwickeln, welches den Aufwand für die Datenaufbereitung reduziert und somit mehr Zeit für die Datenanalyse schafft. Dazu wurde in **Kapitel 1** zunächst die Problemstellung im bisherigen Ablauf bei der Erstellung einer Statistik erläutert.

Anschließend wurde in **Kapitel 2** ein Verständnis für den Tinnitus geschaffen und die forschende Tinnitus Research Initiative und das Projekt der Internationalen Tinnitus Datenbank vorgestellt. Weiterhin wurden zentrale Begriffe beschrieben, die sowohl für diese Arbeit als auch für die Nutzung des Statistikmoduls relevant sind.

In **Kapitel 3** wurde ein kurzer Einblick über den Funktionsumfang aktueller Statistiksoftware gegeben und anschließend sowohl funktionale als auch nicht-funktionale Anforderungen an das System erhoben.

Im anschließenden **Kapitel 4** wurde das Statistikmodul entworfen. Dafür wurde zunächst die Architektur der Tinnitus Datenbank beschrieben und auf das darunter liegende Datenmodell eingegangen. Als nächstes wurde auf Grund der erhobenen funktionalen Anforderungen der Erstellprozess einer Statistik erläutert. Um die Komplexität gering zu halten und die Bedienung zu erleichtern, wird der Nutzer mit einem Wizard durch den Prozess geleitet. Die einzelnen Schritte wurden hierbei beschrieben. Anschließend wurde die Gestaltung der einzelnen Schritte des Wizards und der statistischen Auswertung vorgenommen und durch MockUps illustriert.

Kapitel 5 widmet sich der Entwicklung des Statistikmoduls. Dafür wurden zunächst die verwendeten Technologien und Frameworks kurz erläutert. Im Anschluss wurden Aspekte der Implementierung genau beschrieben. Dabei stehen die Eingaben des Nutzers und die daraus resultierenden Verarbeitungsschritte bis zur Anzeige einer Statistik als Resultat im Vordergrund. Abschließend wurde die grafische Umsetzung des Statistikmoduls an Hand von Screenshots illustriert und erläutert.

6.2 Ausblick

Während der Entwicklung des Statistikmoduls wurde deutlich, dass vor allem die Zeit zwischen Request und Response ein nicht zu unterschätzendes Problem ist. Dauert die Verarbeitung der Daten zu lange und die Response wird als Resultat dessen nicht gesendet, läuft der Browser unter Umständen in einen *Connection Timeout*. Dies geschieht im Besonderen dann, wenn insgesamt betrachtet viele Patienten unter vielen Kriterien zu einer Statistik herangezogen werden. Da die Datenbank weiterhin Daten aufnimmt, wird das Problem mit der Zeit nur größer. Hier muss eine Lösung gefunden werden. Das Einführen der Tabelle *session_content_score* war ein erster großer Schritt in die richtige Richtung. Sie spart viele tausende Berechnungen der Summenscores pro erstellter Statistik, was bisher den zeitlichen Aufwand ungefähr halbiert hat. Es gilt jedoch weitere Möglichkeiten zu finden Berechnungen zu sparen, Datenbankaufrufe zu vermeiden und Redundanzen zu erkennen.

Bei einer Statistik geht es vor allem darum, Werte oder in diesem Fall SGs miteinander zu vergleichen. Daher wären weitere Vergleichsmöglichkeiten, wie etwa eine Option Grafiken übereinander zu legen, sinnvoll. Ebenfalls könnten Tabellenwerte kontextuell ergänzt werden. Ein Beispiel hierfür wäre, dass die deskriptiven Statistiken farblich markiert werden, also der Wert einer SG mit dem kleinsten Mittelwert rot und der mit dem größten Mittelwert grün markiert wird. Ein weiteres Beispiel ist die Anzeige der Individualdaten. Diese beinhaltet oft mehrere Hundert Patienten pro SG. Um zu überblicken, welche Patienten den Mittelwert der SG wie beeinflusst haben, könnte der Nutzer eine Schwelle angeben. Alle Patienten unterhalb der Schwelle werden rot, die oberhalb grün markiert. Das sind nur einige wenige Möglichkeiten die Vergleichbarkeit der Werte zu erhöhen. Es gibt sicher unzählige.

Ein letzter Punkt der Verbesserung ist ganz klar die Erweiterung des Statistikmoduls um einen größeren Mehrwert zu schaffen. Die statistische Auswertung ist noch in ihren Kinderschuhen. Sie bietet bisher Basisfunktionalitäten und muss weiter ausgebaut werden. Das beinhaltet die Erweiterung der Kriterien, der Filter und der Vergleichsmöglichkeiten.

A Anhang

A.1 Datenmodell

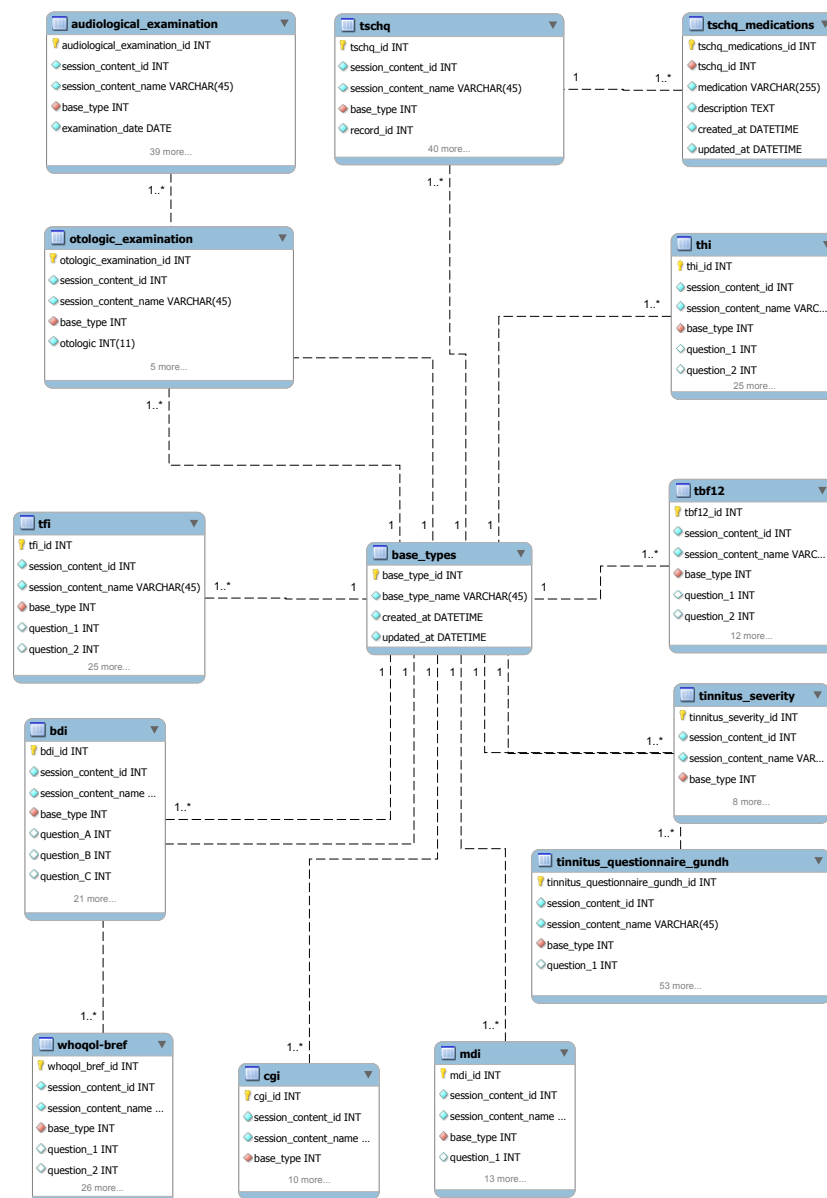


Abbildung A.1: Datenmodell Teil 1

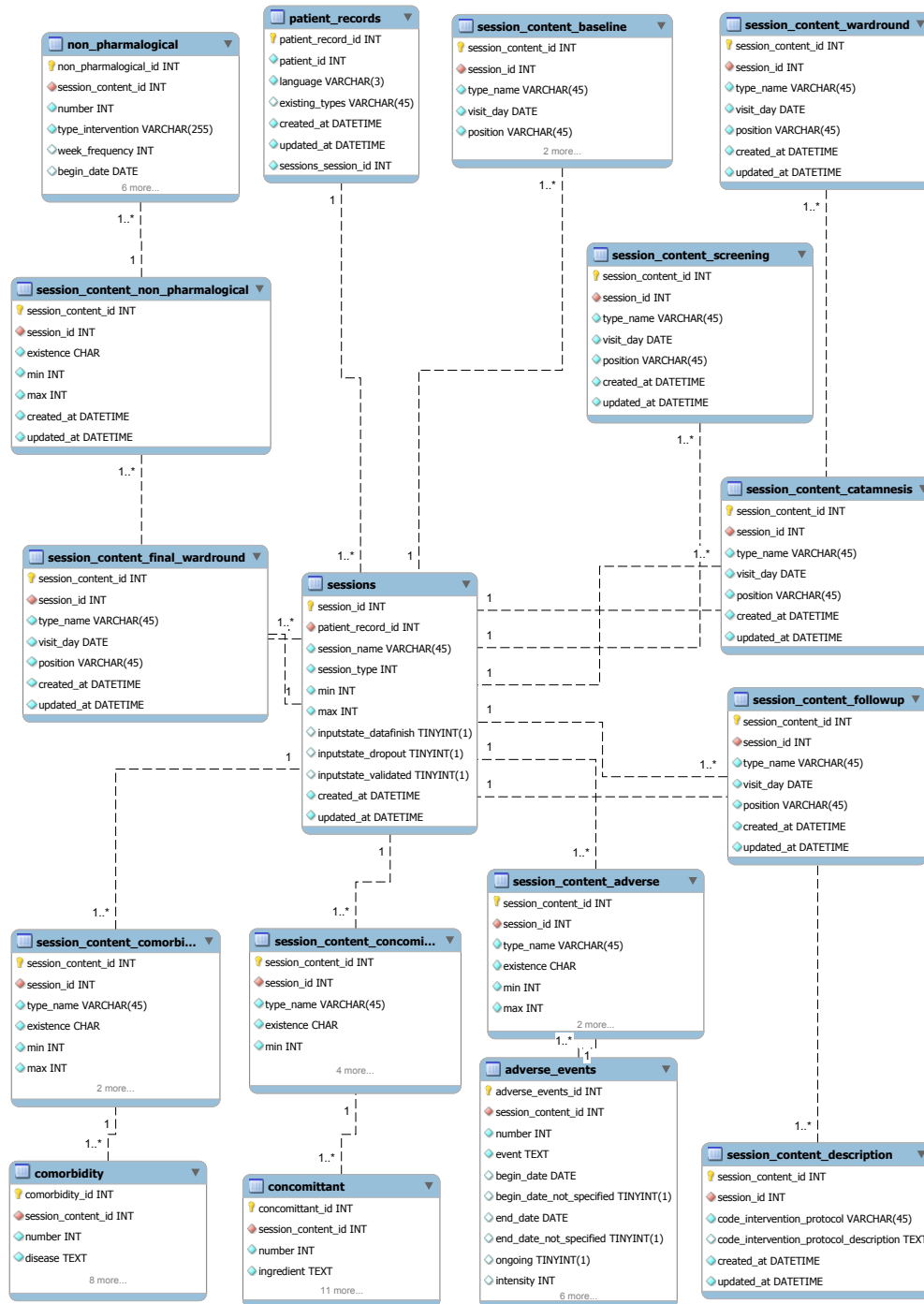


Abbildung A.2: Datenmodell Teil 2

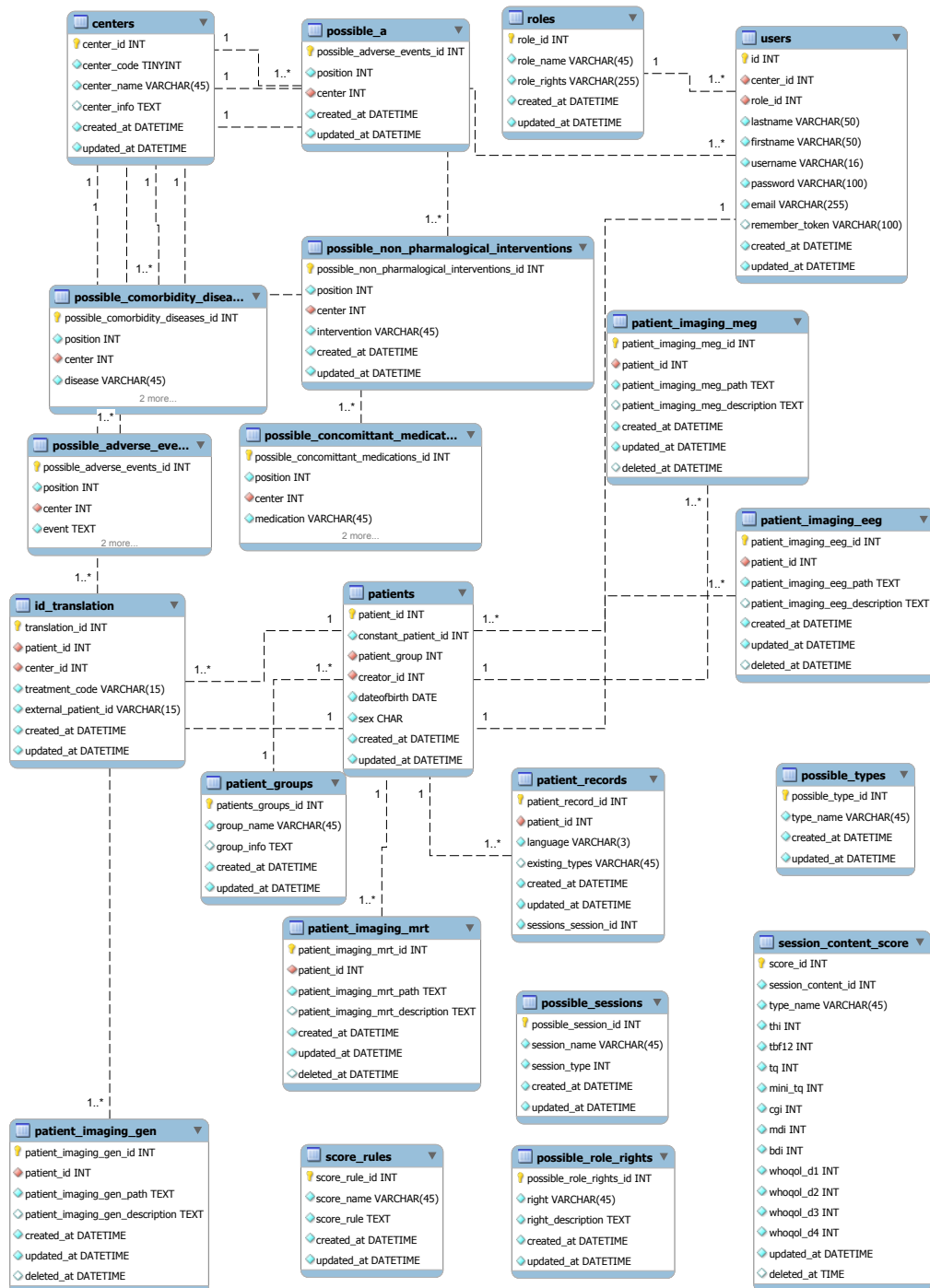


Abbildung A.3: Datenmodell Teil 3

Abbildungsverzeichnis

2.1	Weltkarte der teilnehmenden Zentren [30].	4
2.2	Longitudinale Statistik	8
2.3	Crossectionale Statistik	8
4.1	Architektur der TD	13
4.2	Datenmodell der TD	14
4.3	Flussdiagramm des Wizard	15
4.4	Flussdiagramm der Statistikauswahl	16
4.5	Flussdiagramm der Auswahl des Zeitpunkts einer CS	16
4.6	Flussdiagramm der Auswahl der Zeitpunkte einer LS	17
4.7	Flussdiagramm der Erstellung einer SG	17
4.8	Flussdiagramm der Auswahl der Kriterien	18
4.9	Struktur der Schritte des Wizards	19
4.10	Wahl der Statistikart	19
4.11	Wahl der Zeitpunkte	20
4.12	Erstellung der Gruppen	20
4.13	Wahl der Kriterien	21
4.14	Statistische Auswertung am Beispiel einer LS	22
4.15	Individuelle Daten am Beispiel einer LS	22
5.1	Funtionsweise von PHP [35]	23
5.2	Erstellung einer SG im Wizards	35
5.3	Ansicht der Statistik	36
5.4	Ansicht der Individualdaten der Statistik	36
A.1	Datenmodell Teil 1	39
A.2	Datenmodell Teil 2	40
A.3	Datenmodell Teil 3	41

Literaturverzeichnis

- [1] ALMSAEED STUDIO: *AdminLTE Documentation*. <https://almsaeedstudio.com/themes/AdminLTE/documentation/index.html>. Version: 2016. – [Online; accessed 17-October-2016]
- [2] BALTAKIRANOGLU, Burak: *Konzeption und Realisierung eines Patientenmoduls für eine multinationale und interdisziplinäre Datenbank*. October 2015
- [3] DB-ENGINES: *Chart.js|Open source HTML5 Charts for your website*. <http://www.chartjs.org/>. Version: 2016. – [Online; accessed 05-October-2016]
- [4] DB-ENGINES: *DB-Engines Ranking*. <http://db-engines.com/de/ranking>. Version: 2016. – [Online; accessed 29-September-2016]
- [5] DEUTSCHE TINNITUS-LIGA E.V.: *Tinnitus und sonstige Hörbeeinträchtigungen*. <http://www.tinnitus-liga.de/pages/tinnitus-sonstige-hoerbeeintraechtigungen/tinnitus.php>. Version: 2016. – [Online; accessed 23-September-2016]
- [6] DEUTSCHE TINNITUS-LIGA E.V.: *Ursachen: Vielfalt und Unklarheit*. <http://www.tinnitus-liga.de/pages/tinnitus-sonstige-hoerbeeintraechtigungen/tinnitus/ursachen.php>. Version: 2016. – [Online; accessed 23-September-2016]
- [7] EXPRESS, Software: *IBM SPSS Statistics – Funktionen und Leistungen der SPSS Module*. <http://www.software-express.de/business-intelligence/ibm-spss-statistics-funktionen-und-leistungen-der-spss-module/>, 2016. – [Online; accessed 20-October-2016]
- [8] IBM: *IBM - SPSS Statistics Programmability Extension*. <http://www-03.ibm.com/software/products/de/spss-stats-programmability>, 2016. – [Online; accessed 20-October-2016]
- [9] 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* 1 (2010), 8, Nr. 10, S. 1
- [10] MOHRING, Tim: *Design and Implementation of a NoSQL-concept for an international and multicentral clinical database*, Universität Ulm, Master Thesis, 2016

- [11] PROBST, Thomas ; PRYSS, Rüdiger ; LANGGUTH, Berthold ; SCHLEE, Winfried: Emotion dynamics and tinnitus: Daily life data from the “TrackYourTinnitus” application. In: *Scientific Reports* 6 (2016)
- [12] PROBST, Thomas ; PRYSS, Rüdiger ; LANGGUTH, Berthold ; SCHLEE, Winfried: Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the TrackYourTinnitus application. In: *Scientific Reports* 6 (2016), February
- [13] PRYSS, Rüdiger ; REICHERT, Manfred ; HERRMANN, Jochen ; LANGGUTH, Berthold ; SCHLEE, Winfried: Mobile Crowd Sensing in Clinical and Psychological Trials - A Case Study. In: *28th IEEE Int'l Symposium on Computer-Based Medical Systems*, IEEE Computer Society Press, June 2015, 23–24
- [14] PRYSS, Rüdiger ; REICHERT, Manfred ; LANGGUTH, Berthold ; SCHLEE, Winfried: Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research. In: *IEEE 4th International Conference on Mobile Services (MS 2015)*, IEEE Computer Society Press, June 2015, 352–359
- [15] SCHICKLER, Marc ; PRYSS, Rüdiger ; REICHERT, Manfred ; HEINZELMANN, Martin ; SCHOBEL, Johannes ; LANGGUTH, Berthold ; PROBST, Thomas ; SCHLEE, Winfried: Using Wearables in the Context of Chronic Disorders - Results of a Pre-Study. In: *29th IEEE Int'l Symposium on Computer-Based Medical Systems*, 2016, 68–69
- [16] SCHICKLER, Marc ; PRYSS, Rüdiger ; REICHERT, Manfred ; SCHOBEL, Johannes ; LANGGUTH, Berthold ; SCHLEE, Winfried: Using Mobile Serious Games in the Context of Chronic Disorders - A Mobile Game Concept for the Treatment of Tinnitus. In: *29th IEEE Int'l Symposium on Computer-Based Medical Systems (CBMS 2016)*, 2016, 343–348
- [17] SCHICKLER, Marc ; REICHERT, Manfred ; PRYSS, Rüdiger ; SCHOBEL, Johannes ; SCHLEE, Winfried ; LANGGUTH, Berthold: *Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health*. Springer Vieweg, 2015 (eXamen.press)
- [18] SCHICKLER, Marc ; SCHOBEL, Johannes ; PRYSS, Rüdiger ; REICHERT, Manfred: Mobile Crowd Sensing - A New way of collecting data from trauma samples? In: *XIV Conference of European Society for Traumatic Stress Studies (ESTSS) Conference*, 2015, 244
- [19] SCHOBEL, Johannes ; PRYSS, Rüdiger ; REICHERT, Manfred: Using Smart Mobile Devices for Collecting Structured Data in Clinical Trials: Results From a Large-Scale Case Study. In: *28th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2015)*, IEEE Computer Society Press, June 2015, 13–18
- [20] SCHOBEL, Johannes ; PRYSS, Rüdiger ; SCHICKLER, Marc ; REICHERT, Manfred: A Configurator Component for End-User Defined Mobile Data Collection Processes. In: *Demo Track of the 14th International Conference on Service Oriented Computing (ICSOC 2016)*, 2016

- [21] SCHOBEL, Johannes ; PRYSS, Rüdiger ; SCHICKLER, Marc ; REICHERT, Manfred: A Lightweight Process Engine for Enabling Advanced Mobile Applications. In: *24th International Conference on Cooperative Information Systems (CoopIS 2016)*, Springer, October 2016 (LNCS 10033), 552–569
- [22] SCHOBEL, Johannes ; PRYSS, Rüdiger ; SCHICKLER, Marc ; REICHERT, Manfred: Towards Flexible Mobile Data Collection in Healthcare. In: *29th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2016)*, 2016, 181–182
- [23] 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
- [24] SCHOBEL, Johannes ; PRYSS, Rüdiger ; WIPP, Wolfgang ; SCHICKLER, Marc ; REICHERT, Manfred: A Mobile Service Engine Enabling Complex Data Collection Applications. In: *14th International Conference on Service Oriented Computing (ICSOC 2016)*, 2016 (LNCS 9936), 626–633
- [25] SCHOBEL, Johannes ; SCHICKLER, Marc ; PRYSS, Rüdiger ; REICHERT, Manfred: Process-Driven Data Collection with Smart Mobile Devices. In: *10th International Conference on Web Information Systems and Technologies (Revised Selected Papers)*. Springer, 2015 (LNBIP 226), S. 347–362
- [26] SERQUERA, Jaime ; SCHLEE, Winfried ; PRYSS, Rüdiger ; NEFF, Patrick ; LANGGUTH, Berthold: Music Technology for Tinnitus Treatment Within Tinnnet. In: *Audio Engineering Society Conference: 58th International Conference: Music Induced Hearing Disorders* Audio Engineering Society, 2015
- [27] STENSKE, Irina: *Entwicklung eines Design-Konzepts für eine multinationale Forschungsdatenbank zur Speicherung von longitudinalen Patientendaten*. 2015
- [28] TINNITUS RESEARCH INITIATIVE FOUNDATION: *Foundation*. http://www.tinnitusresearch.org/en/foundation/foundation_en.php. Version: 2016. – [Online; accessed 27-September-2016]
- [29] TINNITUS RESEARCH INITIATIVE FOUNDATION: *Tinnitus Database*. <http://www.tinnitusresearch.org/index.php/for-clinicians/database>. Version: 2016. – [Online; accessed 24-September-2016]
- [30] TINNITUS RESEARCH INITIATIVE FOUNDATION: *Worldmap of Subtyping Centers*. http://database.tinnitusresearch.org/en/map/map_en.php. Version: 2016. – [Online; accessed 27-September-2016]
- [31] W3TECHS: *jQuery now runs on every second website*. https://w3techs.com/blog/entry/jquery_now_runs_on_every_second_website. Version: 2016. – [Online; accessed 29-September-2016]

- [32] W3TECHS: *Usage of client-side programming languages for websites*. https://w3techs.com/technologies/overview/client_side_language/all. Version:2016. – [Online; accessed 29-September-2016]
- [33] W3TECHS: *Usage statistics and market share of Bootstrap for websites*. <https://w3techs.com/technologies/details/js-bootstrap/all/all>. Version:2016. – [Online; accessed 05-October-2016]
- [34] W3TECHS: *Usage statistics and market share of PHP for websites*. <https://w3techs.com/technologies/details/pl-php/all/all>. Version:2016. – [Online; accessed 29-September-2016]
- [35] WIKIMEDIA FOUNDATION INC.: *Funtionsweise von PHP*. https://upload.wikimedia.org/wikipedia/commons/6/67/PHP_funktionsweise.svg. Version:2016. – [Online; accessed 29-September-2016]
- [36] WIKIPEDIA: *Anforderung (Informatik) — Wikipedia, The Free Encyclopedia*. [https://de.wikipedia.org/wiki/Anforderung_\(Informatik\)](https://de.wikipedia.org/wiki/Anforderung_(Informatik)), 2016. – [Online; accessed 06-October-2016]
- [37] WIKIPEDIA: *Bootstrap (Framework) — Wikipedia, The Free Encyclopedia*. [https://de.wikipedia.org/wiki/Bootstrap_\(Framework\)](https://de.wikipedia.org/wiki/Bootstrap_(Framework)), 2016. – [Online; accessed 05-October-2016]
- [38] WIKIPEDIA: *JavaScript — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=JavaScript&oldid=741694587>, 2016. – [Online; accessed 29-September-2016]
- [39] WIKIPEDIA: *JQuery — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=jQuery&oldid=741279449>, 2016. – [Online; accessed 29-September-2016]
- [40] WIKIPEDIA: *Laravel — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Laravel&oldid=742618523>, 2016. – [Online; accessed 05-October-2016]
- [41] WIKIPEDIA: *MySQL — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=MySQL&oldid=741415415>, 2016. – [Online; accessed 29-September-2016]
- [42] WIKIPEDIA: *PHP — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=PHP&oldid=740312082>, 2016. – [Online; accessed 29-September-2016]
- [43] WIKIPEDIA: *SPSS — Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=SPSS&oldid=742207437>, 2016. – [Online; accessed 20-October-2016]

- [44] WIKIPEDIA: *Tinnitus* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Tinnitus&oldid=736517210>, 2016. – [Online; accessed 23-September-2016]
- [45] ZEMAN, Florian ; KOLLER, Michael ; SCHECKLMANN, Martin ; LANGGUTH, Berthold ; LANDGREBE, Michael: Tinnitus assessment by means of standardized self-report questionnaires: psychometric properties of the Tinnitus Questionnaire (TQ), the Tinnitus Handicap Inventory (THI), and their short versions in an international and multi-lingual sample. In: *Health and quality of life outcomes* 1 (2012), 10, Nr. 10, S. 1

Name: Norman Thiel

Matrikelnummer: 702041

Erklärung

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

Ulm, den

Norman Thiel