



Hochschule für Angewandte Wissenschaften Hamburg  
Hamburg University of Applied Sciences



Universitätsklinikum  
Hamburg-Eppendorf

Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Life Sciences

## Bachelorarbeit

# Entwicklung einer webbasierten Datenbank- anwendung zur Erfassung von klinischen Daten in der Neuroradiologie

Vorgelegt von: Gunnar Blunk

Matrikelnummer: XXXXXXXXXX

Studiengang: Medizintechnik

1. Gutachter: Prof. Dr. Petra Margaritoff (HAW Hamburg)

2. Gutachter: Dipl.-Ing.(FH) Dirk Kilian (Universitätsklinikum Hamburg-Eppendorf)

Hamburg, den 10. Mai 2017

# Inhaltsverzeichnis

Abbildungsverzeichnis.....	3
Tabellenverzeichnis.....	4
Abkürzungsverzeichnis.....	5
1. Einleitung.....	6
1.1 Kurzvorstellung Universitätsklinikum Hamburg-Eppendorf und Neuroradiologie.....	6
1.2 Problemstellung .....	7
1.3 Ziel dieser Arbeit .....	9
2. Material und Methoden.....	10
2.1 Software / Entwicklungsumgebung .....	10
2.2 Probenmaterial .....	11
2.3 CSV.....	12
2.4 UML.....	13
2.5 Reguläre Ausdrücke.....	15
2.6 Entwicklungsablauf .....	17
3. Ergebnisse .....	19
3.1 Anforderungsliste.....	19
3.2 Datenbankstruktur .....	21
3.3 Darstellung der Weboberfläche .....	23
3.4 Gesamtarchitektur .....	32
3.5 Funktionskonzept.....	33
3.6 Testpläne und -ergebnisse .....	43
4. Diskussion.....	46
4.1 Vergleich der Ergebnisse mit Aufgabenstellung .....	46
4.2 Ausblick / mögliche Erweiterung.....	47
Literaturverzeichnis.....	48
Anhänge .....	51

# Abbildungsverzeichnis

<b>Abb. 1:</b> Übersicht iStrokeDB und anderes Systeme der NRAD .....	9
<b>Abb. 2:</b> Ausschnitt aus der iStroke Excel-Liste .....	11
<b>Abb. 3:</b> Ausschnitt aus der GSR-Importdatei .....	11
<b>Abb. 4:</b> Tabellarische Daten im CSV-Format .....	12
<b>Abb. 5:</b> Use Case Diagramm der iStrokeDB Weboberfläche.....	19
<b>Abb. 6:</b> E-R Diagramm der iStrokeDB.....	21
<b>Abb. 7:</b> Relationales Datenbankschema der iStrokeDB. ....	22
<b>Abb. 8:</b> Darstellung von Registrierung, Login und Passwort zurücksetzen.....	23
<b>Abb. 9:</b> Suchen und Auswählen eines Datensatzes über die Case ID .....	24
<b>Abb. 10:</b> Bearbeiten und Löschen eines Datensatzes.....	25
<b>Abb. 11:</b> Anlegen eines neuen Datensatzes.....	26
<b>Abb. 12:</b> Archivierung und Wiederherstellung von Datensätzen .....	27
<b>Abb. 13:</b> Darstellung des Importvorgangs einer GSR Datei .....	28
<b>Abb. 14:</b> Export von Daten aus der iStrokeDB .....	29
<b>Abb. 15:</b> Erweiterte Funktionen der Admin Oberfläche .....	30
<b>Abb. 16:</b> Hinzufügen, Ändern und Löschen von Frage-Items in der Admin Oberfläche.....	31
<b>Abb. 17:</b> Schichten-Modell eines Software Systems.....	32
<b>Abb. 18:</b> Checkbox-Auswahl in der GSR Datenbank .....	37
<b>Abb. 19:</b> Suchmaske.....	38
<b>Abb. 20:</b> Suchergebnis einer Suchanfrage. ....	38
<b>Abb. 21:</b> Auswahl eines Suchkriteriums anhand von Suchvorschlägen.....	39
<b>Abb. 22:</b> Beispiel Speicherung einer Suchanfrage .....	40
<b>Abb. 23:</b> Speicherung der Suchanfrage als Zeichenkette im JSON Format. ....	40
<b>Abb. 24:</b> Neues Item zum Fragenkatalog hinzufügen.....	41
<b>Abb. 25:</b> Item Parameter bearbeiten.....	41
<b>Abb. 26:</b> Vordefinierte Antwortmöglichkeiten für ein Item festlegen. ....	42
<b>Abb. 27:</b> Reihenfolge der Items im Fragenkatalog verändern.....	42

# Tabellenverzeichnis

<b>Tab. 1:</b> Verwendete Software und Bibliotheken.....	10
<b>Tab. 2:</b> Grundelemente UML Aktivitätsdiagramm.....	14
<b>Tab. 3:</b> RegExp Operatoren.....	16
<b>Tab. 4:</b> RegExp Metazeichen.....	16
<b>Tab. 5:</b> RegExp Modifikatoren.....	16
<b>Tab. 6:</b> Verwendete Anforderungsarten und ihre Abkürzungen.....	20
<b>Tab. 7:</b> Ausschnitt Anforderungsliste iStrokeDB.....	20
<b>Tab. 8:</b> Ausschnitt Checkbox-Ergebnis in GSR Export-Exceldatei.....	37
<b>Tab. 9:</b> Testfall: Grundfunktionalität.....	44
<b>Tab. 10:</b> Testfall: Provozieren von Falscheingaben.....	45
<b>Tab. 11:</b> Anforderungsliste iStrokeDB.....	54
<b>Tab. 12:</b> Testfall - Registrieren eines neuen Benutzeraccounts.....	55
<b>Tab. 13:</b> Testfall - Unzulässige Eingaben während des Registriervorgangs.....	56
<b>Tab. 14:</b> Testfall - Login mit neu erstelltem Benutzeraccount.....	57
<b>Tab. 15:</b> Testfall - Login mit falschen & leeren Benutzerdaten.....	57
<b>Tab. 16:</b> Testfall - Anlegen eines neuen Datensatzes mit 3 Eingabewerten.....	58
<b>Tab. 17:</b> Testfall - Bearbeiten des neu angelegten Datensatzes.....	58
<b>Tab. 18:</b> Testfall - Monitoring der veränderten Eingabewerte.....	59
<b>Tab. 19:</b> Testfall - Unzulässige Eingaben während der Datensatz Bearbeitung.....	60
<b>Tab. 20:</b> Testfall - Datensatz löschen & wiederherstellen.....	61
<b>Tab. 21:</b> Testfall - Exportieren der aktuellen Datenbank Instanz.....	62
<b>Tab. 22:</b> Testfall - Individueller Export von Daten.....	63
<b>Tab. 23:</b> Testfall - Import von GSR Testdaten.....	64
<b>Tab. 24:</b> Testfall - Import einer GSR Testdatei, welche nicht dem Import Format entspricht.....	65
<b>Tab. 25:</b> Testfall - Import einer GSR Testdatei ohne Importdaten.....	65
<b>Tab. 26:</b> Testfall - Suche mit festgelegten Suchparametern.....	66
<b>Tab. 27:</b> Testfall - Suche mit festgelegten Suchparametern #2.....	67
<b>Tab. 28:</b> Testfall - Exportieren aller Untersuchungsdaten zu den gefundenen Datensätzen.....	68
<b>Tab. 29:</b> Testfall - Suche mit fehlerhaften Suchparametern.....	69
<b>Tab. 30:</b> Testfall - Suche mit unvollständigen Suchparametern.....	69
<b>Tab. 31:</b> Testfall - Hinzufügen eines neuen Frage-Items.....	70
<b>Tab. 32:</b> Testfall - Bearbeiten des Frage-Items aus #T_0601.....	71
<b>Tab. 33:</b> Testfall- Verschieben der Position des Frage-Items aus #T_0601 im Fragenkatalog.....	72
<b>Tab. 34:</b> Testfall - Löschen des Frage-Items aus #T_0601.....	73
<b>Tab. 35:</b> Testfall - Neuvergabe eines Passworts bei Passwort Reset Anfrage.....	74

## Abkürzungsverzeichnis

<b>CSV</b>	Comma-separated values
<b>DB</b>	Datenbank
<b>DDL</b>	Data Definition Language
<b>DeGIR</b>	Deutsche Gesellschaft für Interventionelle Radiologie und minimal-invasive Therapie
<b>DICOM</b>	Digital Imaging and Communications in Medicine
<b>EMRAM</b>	Electronic Medical Record Adaption Model
<b>ERM</b>	Entity-Relationship-Model
<b>GSR</b>	German Stroke Registry
<b>HIMSS</b>	Healthcare Information and Management Systems Society
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>JSON</b>	JavaScript Object Notation
<b>KIS</b>	Krankenhausinformationssystem
<b>MR</b>	Magnetresonanz
<b>NRAD</b>	Klinik für Neuroradiologische Diagnostik und Intervention
<b>PHP</b>	<b>PHP: Hypertext Preprocessor</b>
<b>RegExp</b>	Regular Expression
<b>SQL</b>	Structured Query Language
<b>UKE</b>	Universitätsklinikum Hamburg-Eppendorf
<b>UML</b>	Unified Modeling Language
<b>XML</b>	Extensible Markup Language

# 1. Einleitung

## 1.1 Kurzvorstellung Universitätsklinikum Hamburg-Eppendorf und Neuroradiologie

Das Universitätsklinikum Hamburg-Eppendorf (UKE) ist mit über 10000 Mitarbeitern und über 386 000 behandelten Patienten im Jahr 2015 [1] eines der größten und modernsten Krankenhäuser in der Bundesrepublik Deutschland und Europa.

Mit seinen 14 Zentren und der Medizinischen Fakultät der Universität Hamburg wird am Standort ein starker Fokus auf Forschung und Lehre gelegt [2]. Als erstes Krankenhaus in Europa hat das UKE allumfassend eine elektronische Patientenakte eingeführt, die einen komplett papierlosen Arbeitsablauf ermöglicht. Für diese Leistung wurde das UKE im Jahr 2011 mit dem *Electronic Medical Record Adaption Model* (EMRAM) Stage 7 Award ausgezeichnet und somit als erstes vollständig digitales Krankenhaus Europas zertifiziert [3].

Das EMRAM wird von der *Healthcare Information and Management Systems Society* (HIMSS) verwendet, um Krankenhäuser nach dem erreichten Nutzungsgrad der elektronischen Patientenakte einzustufen. Die HIMSS ist eine weltweit tätige Organisation mit dem Ziel, die Gesundheitsversorgung durch den intelligenten Einsatz von IT zu verbessern [4].

Die Klinik für Neuroradiologische Diagnostik und Intervention (NRAD) als Teil des Zentrums für Radiologie und Endoskopie hat ihren Schwerpunkt auf die bildgebende Diagnostik und minimal-invasive Therapie von Erkrankungen des Gehirns und Rückenmarkes gelegt [5]. In der durchgehend besetzten Stroke-Unit steht eine interventionelle akute Schlaganfalltherapie bei Notfällen jederzeit zur Verfügung [6].

Zusätzlich arbeiten verschiedene Forschungsgruppen an der Verbesserung von Diagnostik und Therapie neurologischer Erkrankungen, wobei die Forschungsschwerpunkte in der interventionellen Angiographie, Bildverarbeitung sowie MR (Magnetresonanz) - Physik liegen [7].

Des Weiteren werden eigene klinische Studien durchgeführt und an internationalen neuroradiologischen Studien teilgenommen. Zu den Studienschwerpunkten zählt vor allem die interventionelle Schlaganfalltherapie [8].

## 1.2 Problemstellung

Jährlich erleiden in Deutschland etwa eine Viertel Million Menschen einen Schlaganfall [S. 1] [9]. Dieser zählt neben Krebs und Herzinfarkt zu den häufigsten Todesursachen in der Bundesrepublik. Zudem ist er die häufigste Ursache für dauerhafte Behinderungen und in Industrieländern, gemessen an den Kosten für Akutbehandlung, Rehabilitation und Therapie, die teuerste Krankheit überhaupt [S. 80] [10] [S. 198] [11].

Verursacht wird ein Schlaganfall in 80 Prozent der Fälle [S. 1] [9] durch ein Blutgerinnsel (*Thrombus*), welcher ein Blutgefäß verschließt und als Folge zu einer Minderdurchblutung des Gehirns führt (*ischämischer Schlaganfall bzw. Hirninfarkt*). Betroffene Gehirnareale können aufgrund der Mangelversorgung absterben [S. 201] [11] und bei der Person zu neurologischen Defiziten bis hin zu schweren Behinderungen oder Tod führen [S. 370] [12] .

Schlaganfälle zählen somit zu den medizinischen Notfällen. Bereits bei Verdacht sollte eine schnellstmögliche Verlegung auf eine für Schlaganfälle spezialisierte Station (*stroke unit*) erfolgen. Nach gesicherter Diagnose ist das Hauptziel die Wiedereröffnung der verschlossenen Blutgefäße (*Rekanalisation*) [S. 212 f.] [11].

Die Wiedereröffnung kann medikamentös oder operativ erfolgen. Während bei der intravenösen Thrombolyse ein Medikament zur Auflösung des Blutgerinnsels verabreicht wird [S. 370] [12], hat die mechanische Thrombektomie das Ziel, das Blutgerinnsel mithilfe eines Katheter-ähnlichen Gerätes aus dem Gefäß zu entfernen [S. 371] [12]. Diese Methode hat aufgrund positiver Studien in den letzten Jahren erheblich an Bedeutung gewonnen [S. 1] [13].

Im Regelfall wird die mechanische Thrombektomie in Kombination mit der intravenösen Thrombolyse durchgeführt [S. 8] [14], um ein bestmögliches Behandlungsergebnis zu erreichen. Außerdem kommt sie zum Einsatz, wenn der Thrombus lyseresistent ist [S. 371] [12] oder wenn eine Auflösung des Blutgerinnsels mit Medikamenten aufgrund von Kontraindikationen nicht möglich ist [S. 9] [14].

Im Rahmen eines Projekts der NRAD werden alle Patienten, bei denen eine mechanische Thrombektomie durchgeführt wurde, pseudonymisiert erfasst und die gewonnenen Untersuchungsergebnisse von den verantwortlichen Ärzten in einer Microsoft Excel-Liste festgehalten. Dem Projekt und der Liste wurde der Name *iStroke* gegeben.

Das Ziel dieses Projekts soll die Erschaffung einer möglichst umfassenden Datensammlung zu allen Aspekten dieser Behandlung sein. Von Interesse sind dabei neben Basisdaten wie Alter und Geschlecht des Patienten auch z.B. die Uhrzeit der Erstaufnahme oder ob bestimmte Behandlungsprozeduren durchgeführt und welche Medikamente verabreicht wurden. Diese Aspekte wurden in Form von sogenannten *Frage-Items* definiert und als Tabellenüberschriften in die Liste gesetzt.

Aus dieser Datensammlung sollen in Zukunft bei Bedarf ausgewählte Daten extrahiert werden können (sog. *Daten-Kollektive*), etwa im Rahmen einer Studie oder als Datengrundlage für Abschlussarbeiten am UKE.

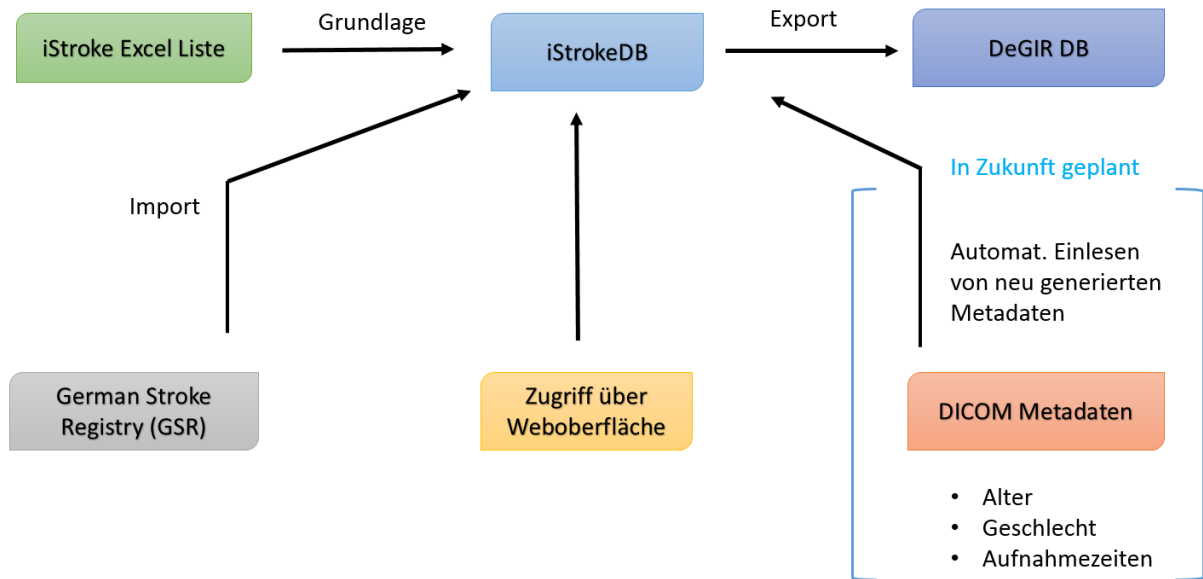
Jedoch hat die zur Speicherung der Untersuchungsdaten verwendete Excel-Liste mehrere Nachteile:

1. Aufgrund der Menge an enthaltenen Datensätzen und Spaltenüberschriften ist der Überblick sehr schlecht. Der Vergleich von Daten zwischen mehreren Datensätzen ist nur über Umwege möglich.
2. Die Einträge sind aufgrund der manuellen Eingabe über Microsoft Excel häufig in einer uneinheitlichen Formatierung abgespeichert – z.B. werden Uhrzeiten von Untersuchungen zusammen mit einem kurzen Kommentar in einer Tabellenzelle abgelegt. Eine spätere Auswertung der Uhrzeiten ist somit unmöglich.
3. Die *iStroke* Datensammlung greift auf Daten aus anderen Datenbanken zurück, welche unter hohem Zeit- und Arbeitsaufwand manuell mit den Einträgen aus der Liste abgeglichen und an der richtigen Stelle reinkopiert werden müssen.
4. Bei Änderungen oder Löschen von Einträgen besteht keine Möglichkeit der Rückverfolgbarkeit und Wiederherstellung von älteren Einträgen. Ebenso bleibt die Person, die geändert hat, anonym. Teilweise wurde versucht, dieses Problem zu lösen, indem die Liste mehrfach mit fortlaufender Datumsangabe im Dateinamen abgespeichert wurde.
5. Des Weiteren kann die Liste nicht von mehreren Personen zur selben Zeit bearbeitet werden.
6. Da die enthaltenen Daten in Zukunft auch in andere Datenbanken importiert werden sollen, müsste die Liste vorher manuell aufbereitet, fehlerbereinigt und in ein passendes Dateiformat für den Import konvertiert werden, was ebenfalls sehr zeitintensiv ist.
7. Gezielte Extraktion von Daten für andere Zwecke (Studien, Abschlussarbeiten) muss ebenfalls manuell erfolgen.



### 1.3 Ziel dieser Arbeit

Ausgehend von der bestehenden *iStroke* Excel-Liste soll eine relationale Datenbank (Name: *iStrokeDB*) mit zugehöriger Weboberfläche zur Benutzung der Datenbank programmiert und als Bindeglied zu anderen Systemen am Standort (Abb. 1) implementiert werden. Die Datenbank soll die Excel-Liste vollständig ersetzen.



**Abb. 1:** Übersicht *iStrokeDB* und anderer Systeme der NRAD. DeGIR steht für „Deutsche Gesellschaft für Interventionelle Radiologie und minimal-invasive Therapie“ [15]

Die unter 1.2 genannten Nachteile sollen mit dieser Datenbank gelöst werden. Die Datenbank soll im Intranet der NRAD von jedem KIS (*Krankenhausinformationssystem*) - PC aus über einen Webbrowser aufrufbar und benutzbar sein.

Des Weiteren sollen Schnittstellen für den Import und Export zu bereits bestehenden Datenbanken / Systemen bereitgestellt werden.

Die Datenbank soll für zukünftige Änderungen von Frage-Items im Fragenkatalog leicht anpassbar und erweiterbar sein.

Ein besonderer Fokus liegt auf der Benutzerfreundlichkeit, Vorbeugung von Falscheingaben, Schaffung von einheitlichen Datensätzen und der Beschleunigung der Arbeitsprozesse des anwendenden Personals. Eine detaillierte Beschreibung aller Anforderungen befindet sich im Anhang.

## 2. Material und Methoden

### 2.1 Software / Entwicklungsumgebung

Die nachfolgend aufgelistete Software (Tab. 1) wurde aufgrund der weiten Verbreitung und einfachen Wartbarkeit empfohlen. Einige Erweiterungen wurden erst im Verlauf des Projekts miteingebunden, dadurch konnten bestimmte Funktionen zeitsparend und ohne großen Programmieraufwand realisiert werden.

Name der Software	Version	Beschreibung
<b>Linux Ubuntu (64-bit) Server</b>	16.04.1 LTS	Linux-Distribution mit dem Fokus auf Serverdienste [16]
<b>Apache HTTP Server</b>	2.4.18	Bereitstellung eines HTTP Webservers [17]
<b>PHP</b>	7.0.8	weit verbreitete Open Source-Skriptsprache, welche speziell für die Webprogrammierung geeignet ist und in HTML eingebettet werden kann [18]
<b>PostgreSQL</b>	9.5.4	objektrelationales Datenbankmanagementsystem, konform mit dem ANSI-SQL:2008 Standard [19]
<b>phpPgAdmin</b>	5.1	Web-basiertes, dadurch system-unabhängiges Verwaltungsprogramm für PostgreSQL Datenbanken [20]
<b>PHP:PDO extension for PostgreSQL</b>	1.0.2	<b>PHP Data Objects</b> – Erweiterung / Schnittstelle, um mit PHP auf PostgreSQL Datenbanken zuzugreifen [21]. Maskiert automatisch Sonderzeichen und verhindert dadurch <i>SQL Injection</i> <sup>1</sup>
<b>PHPExcel</b>	1.8.1	PHP Bibliothek um Tabellen-Dateiformate (z.B. Excel Format) einzulesen, zu bearbeiten oder auszugeben [22]
<b>jQuery</b>	3.1.1	JavaScript Bibliothek, um auf Client-Seite schnell und einfach HTML Elemente zu manipulieren [23]
<b>jQuery UI</b>	1.12.1	Erweiterung von jQuery, bietet zahlreiche Vorlagen zur Erweiterung von Benutzeroberflächen [24].
<b>sweetalert2</b>	6.3.0	JavaScript Bibliothek, welche optisch anpassbare Pop-Up Fenster erzeugen kann [25]
<b>VirtualBox</b>	5.1.6	Virtualisierungssoftware, ermöglicht die Benutzung einer beliebigen Anzahl unterschiedlicher Betriebssysteme in einer virtuellen Maschine auf dem eigenen Gastsystem [26] und wurde für den Testbetrieb des Linux Ubuntu Servers verwendet
<b>Microsoft Internet Explorer</b>	10.0.9200	Webbrowser zum Testen der Weboberfläche

*Tab. 1: Verwendete Software und Bibliotheken*

---

<sup>1</sup> SQL Injection ist das bewusste Einschleusen von schädlichem SQL Code aufgrund von Sicherheitslücken bei der Verarbeitung von Benutzereingaben.

## 2.2 Probenmaterial

Die in 1.2 erwähnte *iStroke* Excel-Liste diente als Ausgangsmaterial für die Erstellung der Datenbank. In dieser wurden die Datensätze / Fälle zeilenweise aufgelistet (Abb. 2) und die einzelnen Frage-Items als Spaltenüberschriften angelegt.

Einzelne Spalten, welche Uhrzeit- oder Datumsangaben enthalten, sind zum Teil in Excel dementsprechend formatiert. Beispielsweise werden Einträge mit Excel Datumsformatierung in der Zelle als Anzahl der Tage seit dem 01.01.1900 gespeichert. Während des Imports der Einträge in die *iStrokeDB* musste somit eine passende Konvertierung stattfinden, wofür die in 2.1 genannten Erweiterung *PHPExcel* genutzt wurde.

ID	Soarian_NRA	Soarian_NL	Soarian_com	ID_alt	UKE_case	GSR_ID	Sex	Age
ukm_20100326_0107				60418282			m	59
ukm_20090319_0				60028063			f	84
ukm_20101218_1345				60036830			m	64
ukm_20081110_1050				60777857			m	60
ukm_20101001_1219				60128279			f	51
ukm_001				60036830			m	58
ukm_002				60079117			m	81
ukm_003				60094108			w	66
ukm_004				60245594			f	62
ukm_005				60331350			m	22

**Abb. 2:** Ausschnitt aus der *iStroke* Excel-Liste. Exemplarisch hier schon die Nachteile von Excel erkennbar: in der Spalte Geschlecht / Sex wurde ‚weiblich‘ sowohl mit ‚f‘ als auch mit ‚w‘ abgekürzt.

In der ersten Spalte stand eine pseudonymisierte Patienten-ID, welche allerdings keine eindeutige Identifikationsnummer eines Falls darstellte. Aus den Spaltenüberschriften und Daten ergaben sich die Parameter (Datentyp, vordefinierte Antworten) der Frage-Items für den Fragenkatalog der *iStrokeDB*.

Ausgewählte Daten aus der GSR (*german stroke registry*) - Datenbank wurden ebenfalls in einer Exceldatei (Abb. 3) bereitgestellt, welche zukünftig als Importdatei für die *iStrokeDB* dienen soll. Während des Importvorgangs einer GSR-Datei müssen die Einträge überprüft und, wenn nötig, korrigiert werden. Eine Lösung zur automatische Überprüfung und Korrektur von Einträgen wird in 3.5 vorgestellt.

				Basic data					
Center	Patient	ID	Age [y]	Symptom	Symptom	Last seen	Time of re	Admissio	
14	GSR140001		1	85	yes	19:00	19:00	19:00	2015
14	GSR140002		2	78	yes	10:30	10:20	10:30	2015
14	GSR140003		3	52	yes	12:30	12:30	12:30	2015
14	GSR140004		4	45	yes	20:50	20:50	20:50	2015
14	GSR140005		5	77	yes	14:45	14:45	14:45	2015
14	GSR140006		6	76	yes	15:00			2015
14	GSR140007		7	78	no		22:00		2015

**Abb. 3:** Ausschnitt aus der GSR-Importdatei

## 2.3 CSV

Für den Export von Daten aus der *iStrokeDB* wurde neben dem Excel-Format das CSV-Dateiformat festgelegt. CSV steht für *comma-separated values* und war ein weit verbreitetes Dateiformat zum Austausch von tabellarischen Daten zwischen verschiedenen Programmen und Betriebssystemen, bevor es von der *Extensible Markup Language* (XML) verdrängt wurde [S. 258] [27].

Jedoch wird CSV auch heute noch häufig für solche Zwecke eingesetzt, was im einfachen und leicht lesbaren Aufbau und der simplen Abspeicherung als Textdatei zugrunde liegt.

Zudem existieren für viele Programmiersprachen Skripte oder direkte Befehle zum Import von CSV - Dateien. Beispielsweise wird in PHP der Befehl `fgetcsv()` benutzt, um eine CSV-Datei einzulesen und liefert ein numerisches Array als Rückgabewert [28].

Es existiert keine einheitliche Spezifikation für das Format, daher sollen an dieser Stelle kurz die Grundregeln beschrieben werden, nach der jede CSV-Datei aufgebaut ist [29]:

- Jede Tabellenzeile bzw. Datensatz wird in eine Textzeile geschrieben und mit einem definierten Zeichen zur Trennung von Datensätzen abgeschlossen (in der Regel ein Zeilenumbruch)
- Einzelne Tabellenspalten bzw. Datenfelder werden mit einem definiertem Trennzeichen voneinander abgegrenzt (z.B. Komma, Semikolon oder Tabstop)
- Um Sonderzeichen wie Komma oder Semikolon auch innerhalb eines Datenfeldes benutzen zu können, wird ein Datenfeld von einem Feldbegrenzungszeichen umschlossen (z.B. Anführungszeichen)
- Die erste Zeile in einer CSV-Datei kann als Kopfzeile benutzt werden, um den einzelnen Datenfeldern Spaltennamen zuzuordnen
- Die Anzahl der Spaltennamen in der Kopfzeile und die Anzahl der Datenfelder in jeder Zeile müssen übereinstimmen

```
"Case ID","ID","ID alt","UKE case","GSR ID"  
"1","ukm_20100326_0107","60418282","",""  
"2","ukm_20090319_0","60028063","",""  
"3","ukm_20101218_1345","60036830","",""  
"4","ukm_20081110_1050","60777857","",""
```

**Abb. 4:** Tabellarische Daten im CSV-Format

Die Erzeugung von Exportdateien im CSV-Format wurde mithilfe der in 2.1 genannten Erweiterung *PHPExcel* realisiert.

## 2.4 UML

UML steht für *Unified Modeling Language* (dt. vereinheitlichte Modellierungssprache) und ist eine international standardisierte, grafische Modellierungssprache zur Beschreibung von Software-Systemen [S. 130 f.] [30].

Sie kann verwendet werden, um den gesamten Verlauf des Software-Entwicklungsprozesses von der Anforderungsanalyse über Entwurf und Implementierung grafisch zu dokumentieren [S. 3] [31].

Hierfür werden Diagramme benutzt, wobei die in UML verfügbaren Diagramme in 2 Kategorien eingeteilt werden können. Während Strukturdiagramme den statischen Zustand des Systems zu einem bestimmten Zeitpunkt darstellen, sind Verhaltensdiagramme dynamisch und veranschaulichen zeitliche Abläufe [32].



Der Vorteil in der Benutzung von UML-Diagrammen liegt in der programmiersprachen- und plattformunabhängigen Darstellung, welche leicht und schnell von anderen fachkundigen Personen verstanden werden kann, ohne den (eventuell umfangreichen) Quellcode oder andere Details kennen zu müssen [32].


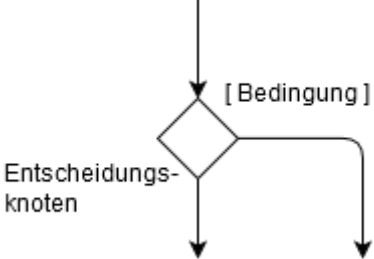
Für eine detaillierte Beschreibung aller Diagrammarten wird auf die vielfältige Literatur zum Thema UML verwiesen.

Folgend sollen die Grundelemente des UML *Aktivitätsdiagramms* (Tab. 2) erklärt werden, welches für die Darstellung der Weboberfläche in 3.3 benutzt wurde.

Das Aktivitätsdiagramm gehört zur Kategorie der Verhaltensdiagramme und stellt Abläufe, Vorgänge und Prozesse in einer gerichteten Reihenfolge mithilfe von Pfeilen und Elementen dar [S. 28] [31].

Zusätzlich kann beschrieben werden, welche Objekte am Prozess teilhaben bzw. diesen durchlaufen und welche verschiedenen Zustände das Objekt während des Prozesses annehmen kann [S. 12] [33]. Objekte können z.B. am Prozess beteiligte Personen oder auch Objekte einer Klasse des Software-Systems sein.

Element	Bedeutung
<p>Startknoten</p>  <p>↓</p>	<p>Markiert den Startpunkt des Diagramms. Ein Diagramm kann ein oder mehrere Startpunkte enthalten.</p>
<p>↓</p>  <p>Endknoten</p>	<p>Beschreibt den Endpunkt des Diagramms. Ein Diagramm kann keinen bis mehrere Endpunkte enthalten.</p>

	<p>Beschreibt eine Tätigkeit, die durchgeführt wird. Nach Beendigung erfolgt die Weiterleitung zum nächsten Element. Die Ausführung einer Aktion stellt den gegenwärtigen Systemzustand dar. Aktionen stellen somit die einzelnen Schritte des Prozesses dar, welche für dessen Durchlaufen abgearbeitet werden müssen.</p>
	<p>Entscheidungen im Ablauf werden mithilfe einer Raute dargestellt, von der 2 oder mehrere Abzweigungen ausgehen können. Bedingungen für eine Abzweigung werden in eckigen Klammern notiert.</p>

**Tab. 2:** Grundelemente UML Aktivitätsdiagramm

Für das Aktivitätsdiagramm existieren weitere Elemente, welche im Rahmen dieser Arbeit nicht weiter ausgeführt werden.

## 2.5 Reguläre Ausdrücke

Aufgrund des in 1.3 genannten Ziels der Vorbeugung von Falscheingaben und Schaffung von einheitlichen Daten kamen bei der Programmierung häufig *regular expressions* (dt. Reguläre Ausdrücke, Abkürzung *RegExp*) zum Einsatz.

Reguläre Ausdrücke sind ein Teilgebiet der theoretischen Informatik [S. 1] [34] und werden von vielen verschiedenen Programmiersprachen unterstützt [S. 118] [34].

Sie werden eingesetzt, um einen Text oder eine Zeichenkette nach einem definierten Muster zu durchsuchen. Dieses Muster muss keine exakten Worte als Suchbegriffe enthalten, sondern kann mithilfe von syntaktischen Regeln (Tab. 3, 4, 5) so erweitert werden, dass z.B. nach Wörtern mit einem bestimmten Anfangs- und Endbuchstaben gesucht werden kann. Diese Art der Suche wird auch *pattern matching* genannt.

Die Syntaxdefinition von *regular expressions* unterliegt keiner Standardisierung, sodass die einzelnen Implementierungen in verschiedenen Programmiersprachen teilweise stark voneinander abweichen [S. 118] [34].

An dieser Stelle soll daher nur ein kurzer Überblick über die Grundlagen der Syntaxformulierung gegeben werden, welche für PHP und JavaScript gelten.

Ein *RegExp* Ausdruck wird in den meisten Fällen von zwei *slash*-Zeichen `//` umschlossen, welche Start und Ende des Ausdrucks definieren.

Operator	Bedeutung	Beispiel
.	Wildcard - passt zu jedem Zeichen	<code>/h.llo/</code> - Passt zu allen Texten die h beliebiges Zeichen gefolgt von llo enthalten
<code>[]</code>	Überprüft ob einer der Zeichen enthalten ist	<code>/h[ae]llo/</code> - Passt zu hallo und hello <code>/[A-Za-z0-9]/</code> - Erlaubt einen Großbuchstaben, Kleinbuchstaben oder eine Zahl
<code>^</code>	Überprüft den Anfang des Textes. Kann ebenfalls für <i>nicht</i> stehen.	<code>/^test/</code> - Der Text muss mit test beginnen <code>/hall[<sup>^</sup>aeiou]/</code> - Die Buchstaben hall dürfen nicht mit a, e, i, o oder u enden
<code>\$</code>	Überprüft den Ende des Textes	<code>/test\$/</code> - Der Text muss mit test aufhören
<code> </code>	Ermöglicht Alternativen	<code>/(der das)/</code> - Passt zu der und das <code>/Kind(er ergarten le)/</code> - Passt zu Kinder, Kindergarten und Kindle.
<code>?</code>	Vorheriges Zeichen ist optional	<code>/iPhone[1-7]?/</code> - Passt zu iPhone, iPhone2 usw. bis iPhone7
<code>*</code>	Wiederholung des vorherigen Elements (0 oder häufiger mal)	<code>/Windows [0-9]*/</code> - Passt zu Windows, Windows 98 und Windows 7, aber nicht Windows7.
<code>+</code>	Wiederholung des vorherigen Elements (1 oder häufiger mal)	<code>/[0-9]+/</code> - Passt zu allen natürlichen Zahlen.

{n}	Exakt n-mal Wiederholung des vorherigen Elements	/[0-9]{3}/ - Passt zu allen 3 stelligen Zahlen.
{m,n}	Wiederholung des vorherigen Elements mindestens m-mal, maximal n-mal.	/[0-9]{1,4}/ - Passt zu allen 1 bis 4 stelligen Zahlen.

**Tab. 3:** RegExp Operatoren [35]

Ist einer der in Tab. 3 genannten Operator-Zeichen Teil des Suchmusters, so muss dieses Zeichen mit einem *backslash*-Zeichen \ maskiert werden, sodass seine Bedeutung als Operator aufgehoben wird.

Falls das *backslash*-Zeichen selbst Teil des Suchmusters ist, so wird dieses ebenfalls mit einem *backslash*-Zeichen \ maskiert.

Zusätzlich gibt es noch vordefinierte Metazeichen (Tab. 4), welche benutzt werden, um spezielle Zeichenklassen zu beschreiben. Diesen Zeichen wird immer ein *backslash*-Zeichen \ vorangestellt.

Metazeichen	Bedeutung
\d	eine Ziffer, also [0-9]
\w	ein Buchstabe, eine Ziffer oder der Unterstrich, also [a-zA-Z_0-9]
\s	Ein einzelnes Leerzeichen

**Tab. 4:** RegExp Metazeichen [S. 119] [34]

Hinter den Ausdruck-umschließenden *slash*-Zeichen kann zusätzlich ein Modifikator (Tab. 5) angegeben werden.

Modifikator	Bedeutung	Beispiel
i	<i>case-insensitive</i> Mustererkennung, d.h. Groß / Kleinschreibung wird ignoriert	/hallo/i - Passt zu <i>hallo</i> , <i>HALLO</i> , <i>hAlLo</i>
g	Globale Mustererkennung, d.h. nach dem ersten Suchtreffer wird weitergesucht	/hallo/g – findet alle <i>hallo</i> Zeichenketten im durchsuchten Text

**Tab. 5:** RegExp Modifikatoren [36]



## 2.6 Entwicklungsablauf

Der Entwicklungsablauf erfolgte nach einer Mischform aus Wasserfallmodell und agiler Methodik.

Beim Wasserfallmodell wird der Entwicklungsprozess in fest definierte Phasen eingeteilt, welche linear nacheinander durchlaufen werden. Erst wenn eine Phase vollendet ist (z.B. Implementierung der Software), wird zur nächsten Phase übergegangen (z.B. Test der Software) [S. 83 f.] [37].

Im Gegensatz dazu stehen bei agilen Vorgehensweisen die am Projekt beteiligten Personen (inklusive der Kunden) und deren Interaktionen im Fokus. Zudem wird Wert auf Flexibilität bei Änderungswünschen gelegt, anstatt starr einem vordefinierten Plan zu folgen [S. 116] [37].

Zu Beginn des Projekts wurde eine Anforderungsanalyse an die Datenbank und die Benutzeroberfläche mit den Stakeholdern (u.a. die Ärzte, die die Datenbank benutzen werden) durchgeführt. Dabei wurde geklärt, welche Lücke die Datenbank füllen soll und welche Schnittstellen zu anderen Systemen benötigt werden. Außerdem wurde grob festgelegt, welche Funktionalitäten die Benutzeroberfläche bereitstellen soll.

Der restliche Anforderungskatalog stand jedoch nicht von vornherein zu 100 Prozent fest, sondern wurde in vielen Gesprächen mit den beteiligten Personen nach und nach aufgebaut und erweitert bzw. verändert. Dies war unumgänglich, da den Stakeholdern selbst nicht von Anfang an die genauen Anforderungen bewusst waren. Eine komplette Liste der Anforderungen befindet sich im Anhang.

Somit wurden kleine, abgeschlossene Programmteile innerhalb kurzer Zeit fertiggestellt und nach einem gemeinsamen Review weitere Veränderungen oder Erweiterungen festgelegt. Die Dokumentation der Programmteile erfolgte pragmatisch innerhalb des Programm-Codes in Form von Kommentarzeilen.

Begonnen wurde mit dem Entwurf der Datenbank. Der Ablauf beim Entwurf einer Datenbank besteht im Wesentlichen aus 3 Schritten [S. 17] [38] [S. 211] [39][S. 25] [40]:

Zuerst wurde der Anforderungskatalog ausgewertet und die bereits bestehende *iStroke* Excel-Liste analysiert und daraus weitere Anforderungen abgeleitet.

Als zweiten Schritt konnte nun ein konzeptionelles Modell in Form eines Entity-Relationship Models (ERM) für die Datenbank erstellt werden. Bei der konzeptionellen Modellierung werden Objekte und Beziehungen verwendet, um den Ausschnitt aus der realen Welt zu beschreiben, der mit der späteren Datenbank abgebildet werden soll [S. 20 f.] [38]. Beim ERM werden die Objekte als Entitäten bezeichnet, welche eindeutig voneinander unterscheidbar sein müssen.

Bei der Erstellung des ERMs wurde bereits die Normalisierung der zukünftigen Datenbank miteinbezogen. Dadurch soll die Speicherung von redundanten Information und das Auftreten von daraus resultierenden Anomalien (beim Einfügen, Ändern und Löschen) vermieden werden [S. 38 f.] [38].

Abschließend wurde daraus im Zuge der logischen Modellierung ein relationales Modell erstellt, bei dem sowohl die Entitäten als auch die Beziehungen aus dem ERM in Form von Tabellen dargestellt werden [S. 29] [38]. Die Beziehung zwischen den Daten werden dabei über die in den Tabellen enthaltenen Werte hergestellt [S. 30] [39].

Zusätzlich werden notwendige Integritätsbedingungen über die eindeutige Zuordnung von Tupeln mithilfe von Identifikationsschlüsseln (sogenannte Primärschlüssel) und Einschränkungen des Wertebereichs (Domänen) von Attributen definiert. Des Weiteren ist das Festlegen von Fremdschlüsseln notwendig, um die Beziehungen zwischen verschiedenen Relationen herzustellen [S. 12] [39].

Aus dem relationalen Modell konnte dann schließlich ein vollständiges Datenbankschema erstellt werden. Die Erstellung erfolgte über die *phpPgAdmin* Benutzeroberfläche, welche den benötigten SQL (*structured query language*) Code für das Schema unter Verwendung der SQL *data definition language* (DDL) automatisch generiert.

Nachdem die leere Datenbank bereitstand, wurde zunächst ein Importskript in PHP erstellt, um die Datensätze aus der bestehenden Excel-Liste in die Datenbank zu kopieren. Der Einbau von verschiedenen Funktionen zur Fehlervorkorrektur (siehe 3.5) war in diesem Schritt sehr wichtig, um keine fehlerhaften Daten (wie in 1.2 beschrieben) in die Datenbank aufzunehmen.

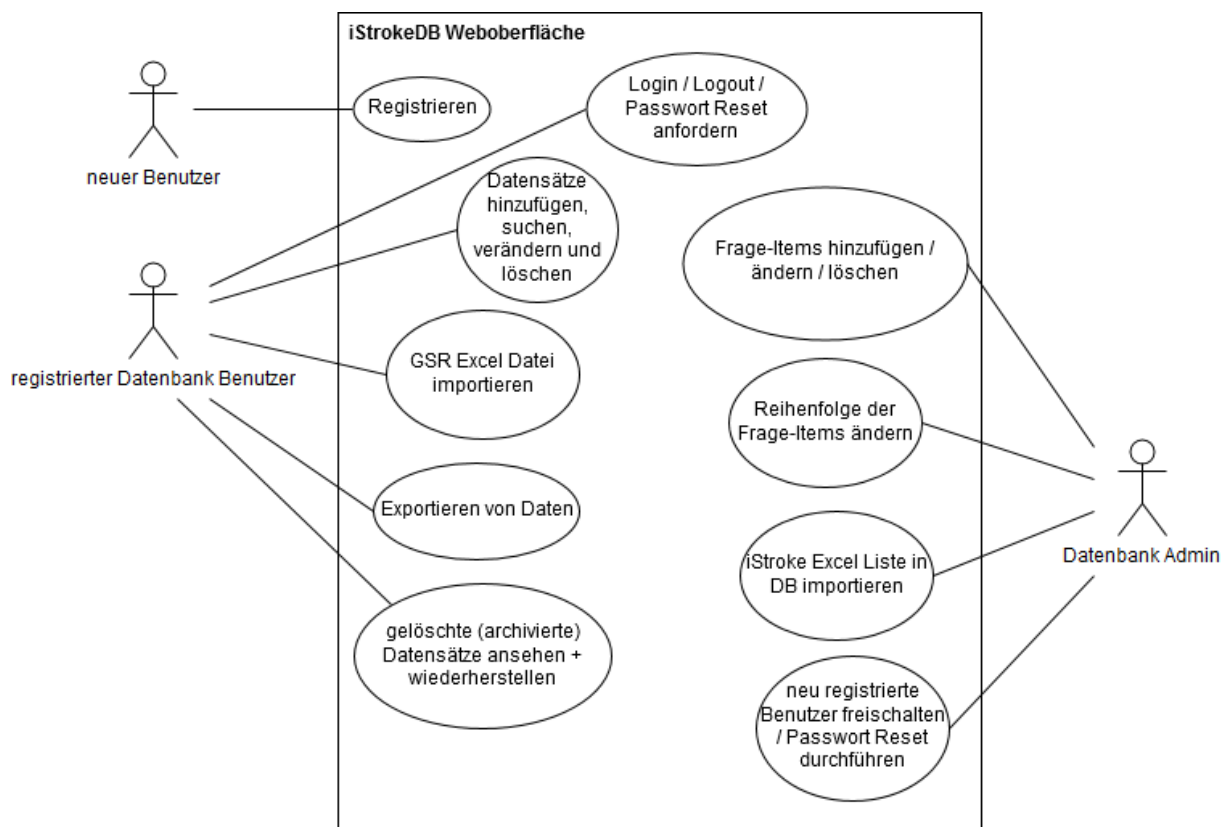
Nun wurde mit der o.g. abschnittswiseen Erstellung der einzelnen Funktionen der Benutzeroberfläche begonnen und die Anforderungsliste Schritt für Schritt umgesetzt.

Zum Ende des Projekts folgte eine intensive Testphase, welche in 3.6 näher beschrieben wird.

### 3. Ergebnisse

#### 3.1 Anforderungsliste

Bevor im Detail auf die Anforderungsliste eingegangen wird, soll mit dem folgenden *Use Case* (dt. Anwendungsfall) Diagramm (Abb. 5) ein erster Überblick über das entwickelte System gegeben werden. Es wurde unter Verwendung der UML bei der Anforderungsanalyse erstellt und ist in der Sprache der Anwender formuliert, um ein schnelles Verständnis über das System mit seinen Anwendungsfällen und verschiedenen Akteuren gewinnen zu können [S. 2 f.] [41][S. 25] [31].



**Abb. 5:** Use Case Diagramm der iStrokeDB Weboberfläche. Akteure sind als Strichmännchen und Use Cases in Ellipsen dargestellt [S. 2 f.] [41].

Folgend aufgelistet ein Ausschnitt aus der Anforderungsliste (Tab. 7) für die Datenbank und die zugehörige Weboberfläche. Jeder Anforderung wurde eine ID zugeordnet, welche die Abkürzung der jeweiligen Anforderungsart (Tab. 6) enthält. Die komplette Anforderungsliste ist im Anhang aufgeführt.

Art der Anforderung	Abkürzung
<b>funktionale Anforderung</b>	FA
<b>technische Rahmenbedingung</b>	TR
<b>Qualitätsanforderung</b>	QA
<b>Schnittstellenanforderung</b>	SA
<b>Leistungsanforderung</b>	LA

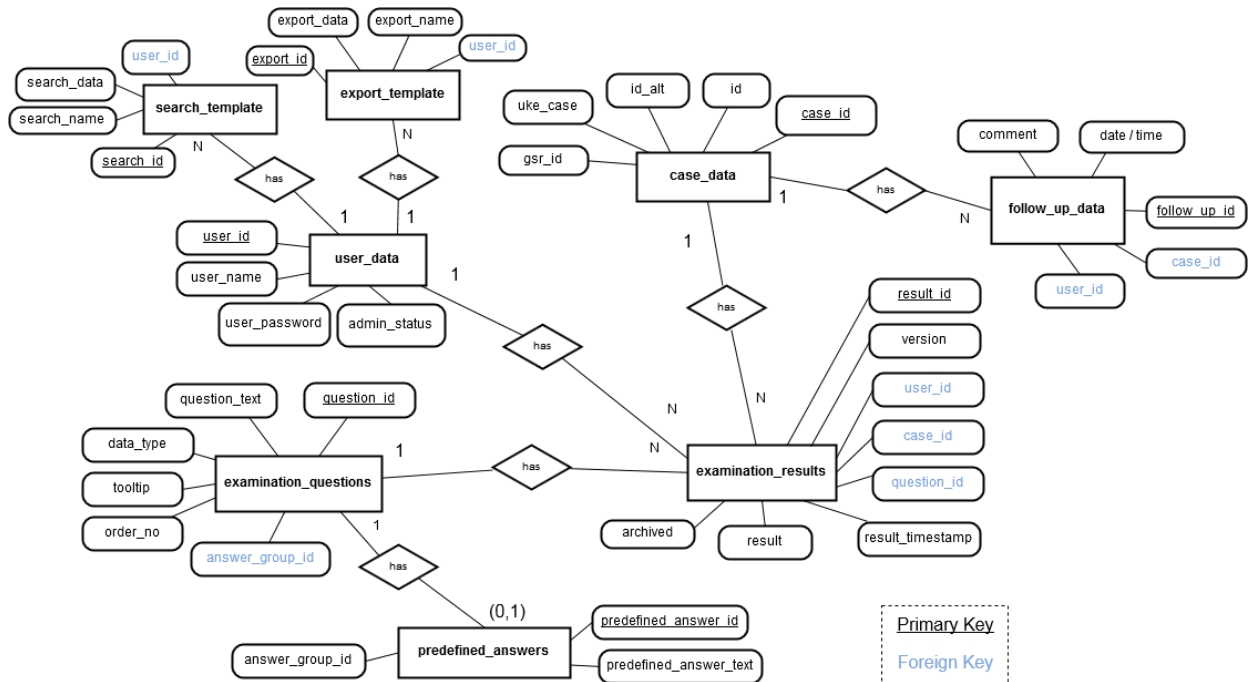
*Tab. 6: Verwendete Anforderungsarten und ihre Abkürzungen.*

ID	Beschreibung der Anforderung
#SA_0003	Eine GSR-Importdatei im Excelformat soll in die Datenbank importiert werden können.
#FA_0700	Eine Importdatei, welche keine GSR-Importdatei im Excel Format darstellt, soll eine entsprechende Fehlermeldung erzeugen.
#FA_0701	Benutzer sollen auswählen können welche Importeinträge sie importieren wollen.
#FA_0702	Bekannte Fehler von Importeinträgen sollen automatisch vorkorrigiert werden.
#FA_0703	Benutzer sollen fehlerhafte Importeinträge manuell korrigieren können.
#FA_0704	Mit einem seitlichen Navigationsmenü soll direkt zu einzelnen Importfehlern gesprungen werden können.
#LA_0001	Der Upload der GSR-Importdatei soll nicht länger als 2 Minuten dauern.

*Tab. 7: Ausschnitt Anforderungsliste iStrokeDB*

### 3.2 Datenbankstruktur

Zur Darstellung der Datenbankstruktur wurde ein Entity-Relationship Diagramm (Abb. 6) erstellt, welches die Beziehungen der einzelnen Entitätstypen miteinander grafisch darstellen soll.



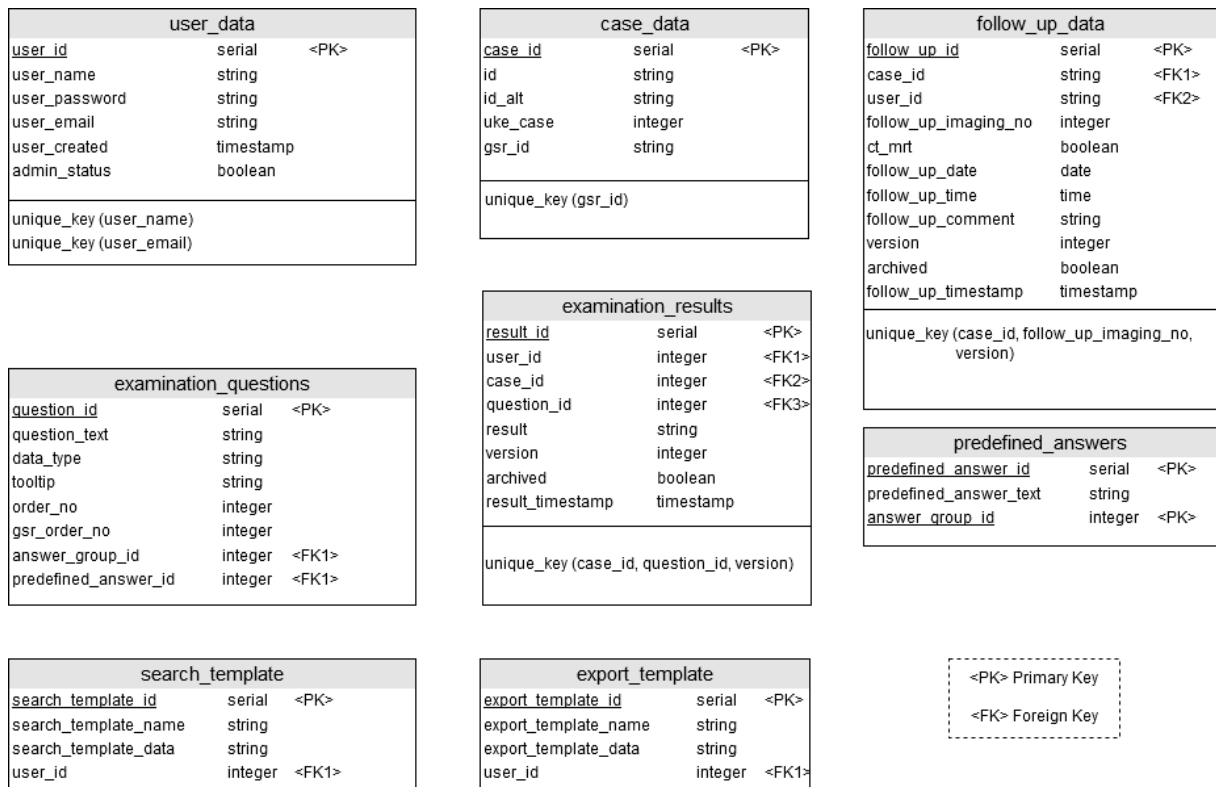
**Abb. 6:** E-R Diagramm der iStrokeDB. Aus Platz- und Übersichtlichkeitsgründen wurden nur die wichtigsten Attribute der Entitätstypen dargestellt.

Ein Entitätstyp wird durch eine Menge von Attributen beschrieben und stellt somit eine Abstrahierung einer Entität dar. Alle Entitäten eines Entitätstyps werden in einer Entitätsmenge zusammengefasst. Ein registrierter Datenbank-Benutzer mit individuellen, gespeicherten Benutzerdaten und einem eindeutigen Identifizierungsschlüssel (*user\_id*) ist somit eine Entität. Die Menge aller gespeicherten Datenbank Benutzern mitsamt ihren Daten stellt die Entitätsmenge dar, während der Begriff „Datenbank-Benutzer“ (im Diagramm *user data*) mit seinen zugehörigen Attributen den Entitätstyp charakterisiert [S. 214 f.] [39].

Aus Abb. 6 wird deutlich, dass jedem Fall eine eindeutige Case ID im Entitätstyp *case data* zugeordnet ist. Jeder Fall enthält eine bestimmte Anzahl an ausgefüllten Einträgen, welche in *examination results* gespeichert werden und eine beliebige Anzahl an Nachfolgeuntersuchungen in *follow up data*.

Jeder ausgefüllte Eintrag in *examination results* basiert auf einem Frage-Item aus *examination questions*, wo der komplette Fragenkatalog hinterlegt ist. Vordefinierte Antworten zu Frage-Items werden in *predefined answers* gespeichert.

Pro Datenbank-Benutzer werden außerdem benutzerdefinierte Suchanfragen (*search template*) und Exportvorlagen (*export template*) gespeichert.



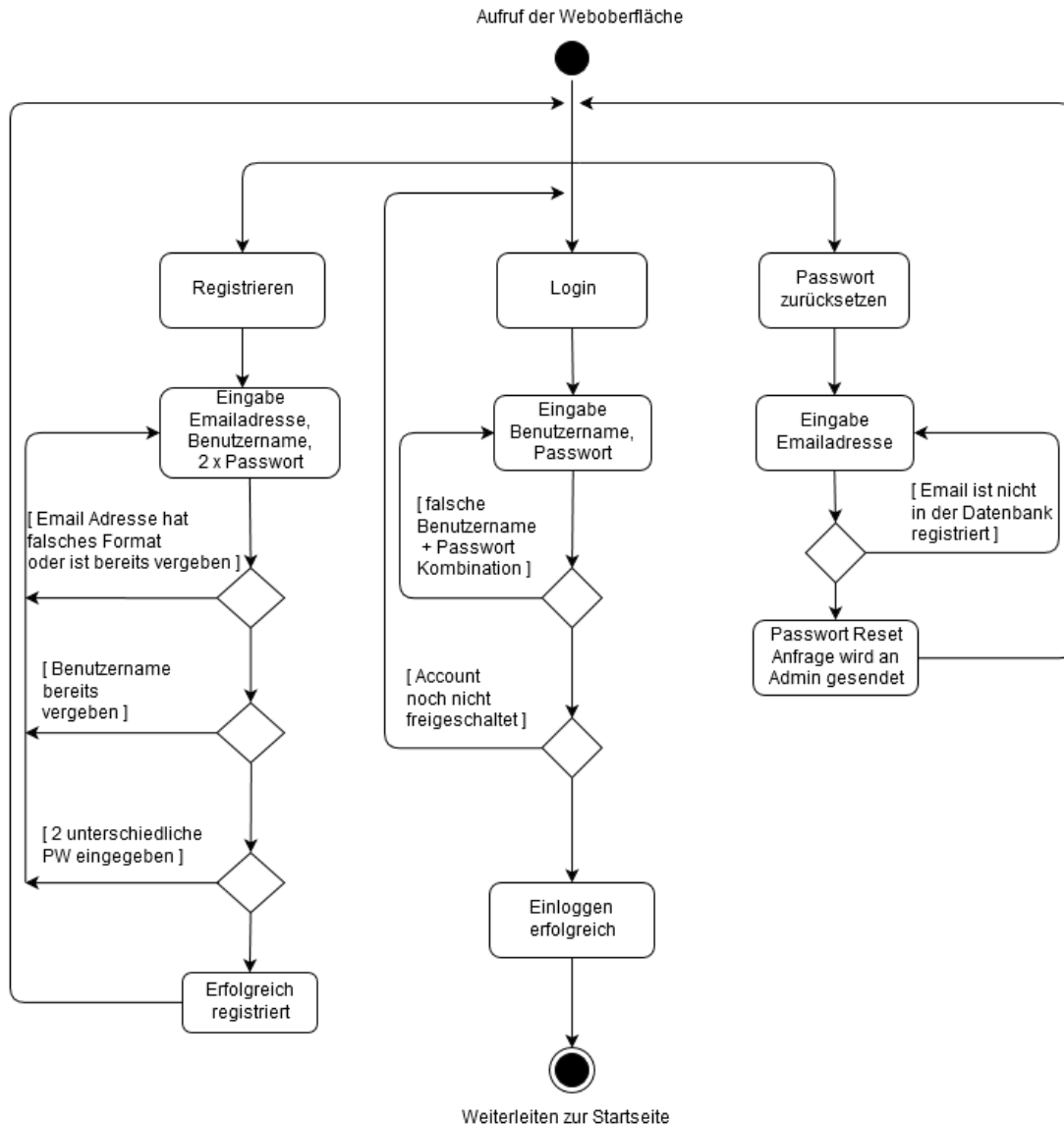
**Abb. 7:** Relationales Datenbankschema der iStrokeDB.

Das relationale Datenbankschema (Abb. 7), welches ausschließlich aus Tabellen besteht, ergibt sich wie in 2.6 beschrieben aus dem Entity-Relationship Model. Es enthält alle Spalten mitsamt festgelegten Datentypen und Integritätsbedingungen in Form von Primär-, Fremd- und eindeutigen Schlüssel Angaben.

Der im Schema verwendete Datentyp *serial* ist kein echter Datentyp, sondern eine interne Festlegung in PostgreSQL, dass eine Spalte als eine fortlaufende Sequenz von *Integer* (ganze Zahlen) - Werten definiert wird [42]. Daher wird dieser Datentyp in der Regel für Primärschlüssel-Spalten verwendet, da diese zwingend eindeutige Schlüssel-Werte zur Identifizierung eines Datensatzes enthalten müssen. In MySQL wird dasselbe Konzept mit der Angabe *AUTO\_INCREMENT* umgesetzt.

### 3.3 Darstellung der Weboberfläche

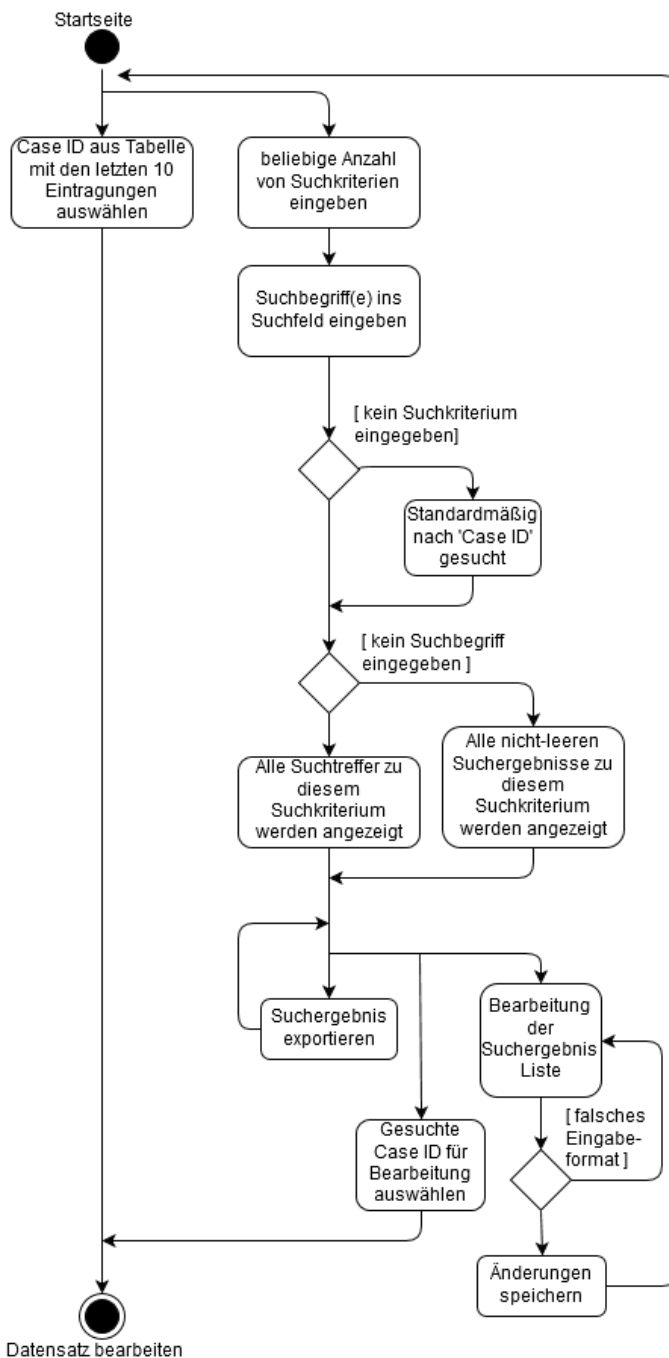
Für die Darstellung der Weboberfläche wurde das UML Aktivitäts-Diagramm gewählt. In den Diagrammen soll die Interaktion des Benutzers mit den einzelnen Funktionen der Oberfläche und das Durchlaufen der einzelnen Prozess-Schritte / Aktionen dargestellt werden.



**Abb. 8:** Darstellung von Registrierung, Login und Passwort zurücksetzen.

Der Benutzer hat beim Aufruf der Weboberfläche (Abb. 8) die Möglichkeiten, sich zu registrieren, sich mit seinen registrierten Benutzerdaten einzuloggen oder ein neues Passwort für seinen Benutzeraccount anzufordern. Bei der Eingabe von Daten werden diese auf Richtigkeit und Vollständigkeit hin überprüft.

Nach erfolgreicher Registrierung muss der Account vom Datenbankadministrator freigeschaltet werden. Ohne einen erfolgreichen Login kann die Startseite der Weboberfläche nicht aufgerufen werden.

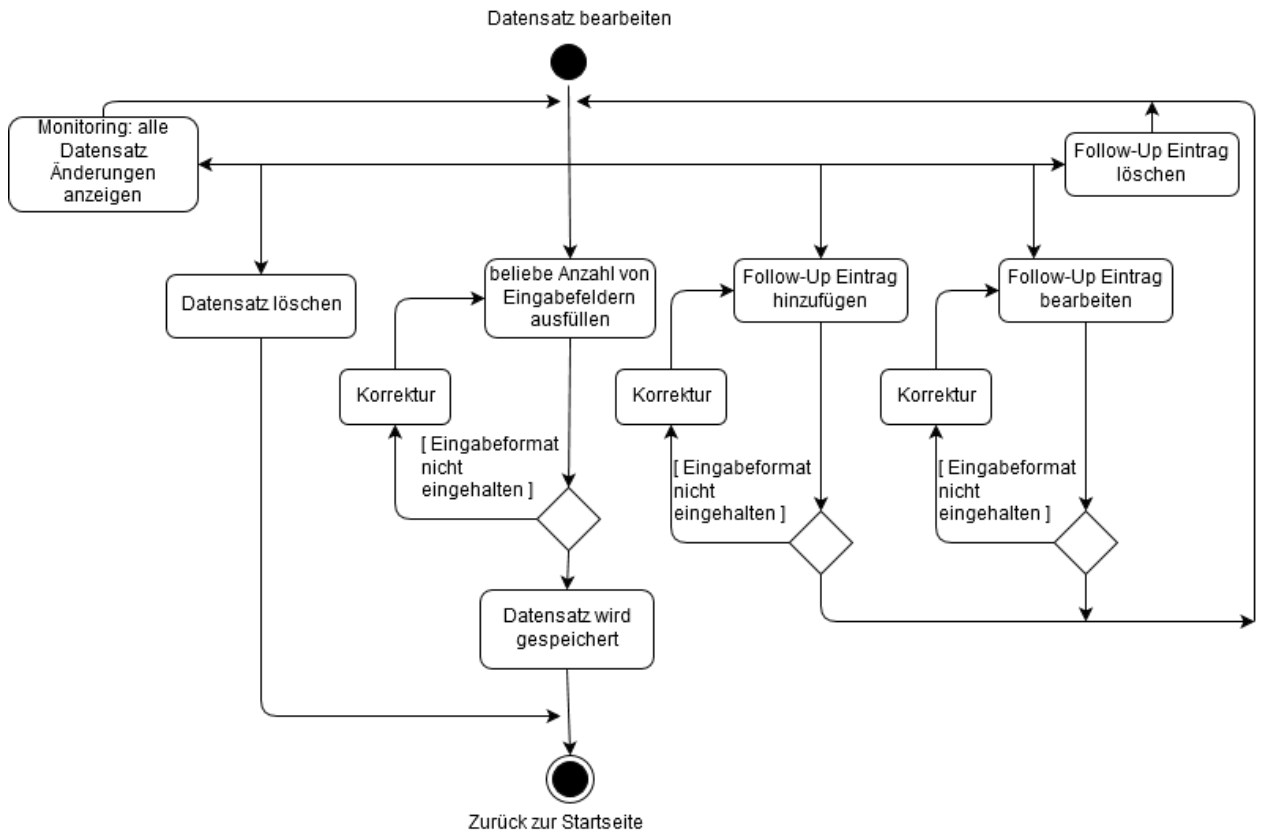


**Abb. 9:** Suchen und Auswählen eines Datensatzes über die Case ID

Nach erfolgreichem Login kann der Benutzer auf der Startseite einen Datensatz anhand von ausgewählten Suchkriterien suchen oder einen der 10 neusten Datensätze direkt aufrufen (Abb. 9). Einem Datensatz wird dabei immer eine eindeutige Case ID zugeordnet, anhand derer er aufgerufen wird.

Zusätzlich kann die erzeugte Liste der Suchergebnisse exportiert werden und in einer separaten Bearbeitungsansicht die Einträge der Suchergebnis-Liste bearbeitet werden.





**Abb. 10:** Bearbeiten und Löschen eines Datensatzes

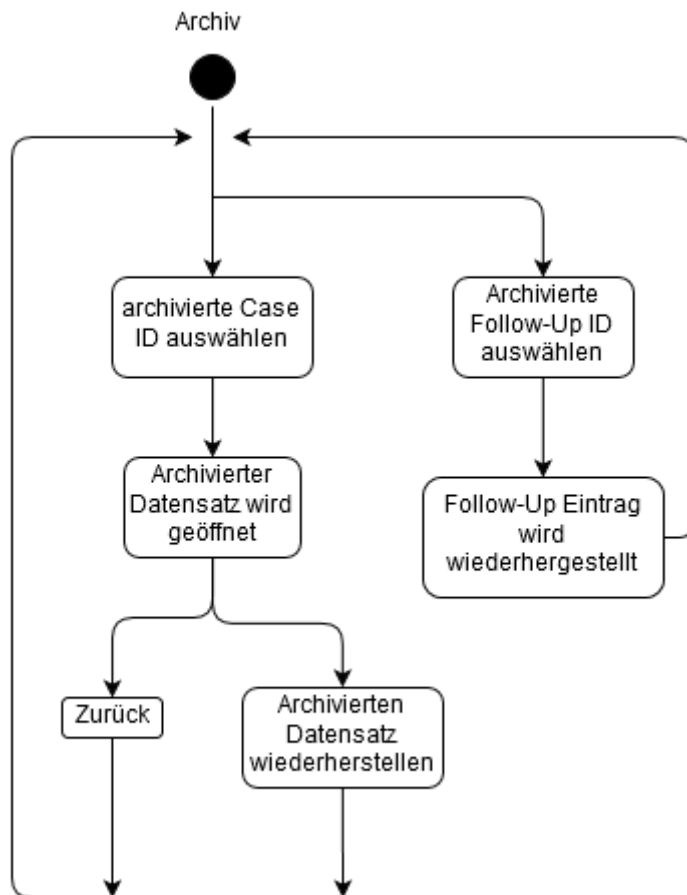
Nach Auswahl einer Case ID kann der Benutzer den zugehörigen Datensatz bearbeiten, löschen sowie alle Änderungen an Einträgen in einer Liste ansehen (Abb. 10).

Zudem kann eine beliebige Anzahl an Nachfolgeuntersuchungen (*Follow-Ups*) hinzugefügt, bearbeitet oder gelöscht werden. Das Einhalten von Eingabeformaten ist auch hier notwendig, bevor eine Abspeicherung stattfinden kann.



**Abb. 11:** Anlegen eines neuen Datensatzes

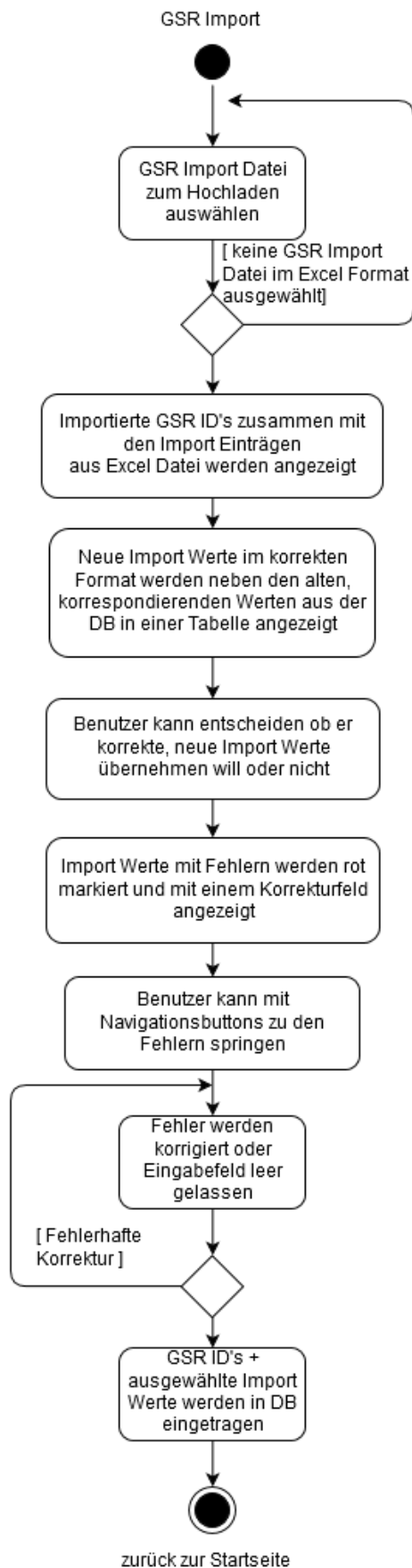
Der Benutzer kann ebenso einen neuen, leeren Datensatz anlegen (Abb. 11) und dabei eine beliebige Anzahl an Eingabefeldern ausfüllen, d.h. die Felder können auch unbearbeitet bleiben. Jedoch muss er sich beim Ausfüllen an vorgegebene Eingabeformate halten, da sonst die Abspeicherung des Datensatzes verhindert wird. Bei erfolgreichem Abspeichern wird dem Datensatz eine neue Case ID zugeordnet.



**Abb. 12:** Archivierung und Wiederherstellung von Datensätzen

Gelöschte Datensätze werden in ein Archiv verschoben und erscheinen nicht mehr im aktuellen Datenbestand, etwa wenn der Benutzer die Suchfunktion benutzt oder Daten aus der DB exportiert werden.

Im Archiv kann der Benutzer gelöschte Datensätze anhand ihrer Case ID ansehen und bei Bedarf wiederherstellen (Abb. 12). Analog verhält es sich mit gelöschten Follow-Up Einträgen.



**Abb. 13:** Darstellung des Importvorgangs einer GSR Datei

Zum Import von Daten aus der GSR Datenbank (Abb. 13) kann der Benutzer eine Importdatei im Excelformat auswählen und zum Server hochladen.

Falls die hochgeladene Datei ein anderes Dateiformat besitzt oder keine GSR Importdaten enthält, wird der Vorgang abgebrochen.

Nach erfolgreichem Hochladen werden in einer Übersicht die Importdaten dargestellt. Hier wird unterschieden, ob eine importierte GSR ID mit zugehörigen Daten schon in der iStrokeDB existiert oder nicht. Wenn dies der Fall ist, wird durch eine farbliche Hervorhebung angezeigt, ob die Importdaten von den korrespondierenden Daten aus der Datenbank abweichen oder sogar fehlerhaft sind, d.h. das Eingabeformat für dieses Frage-Item nicht einhalten.

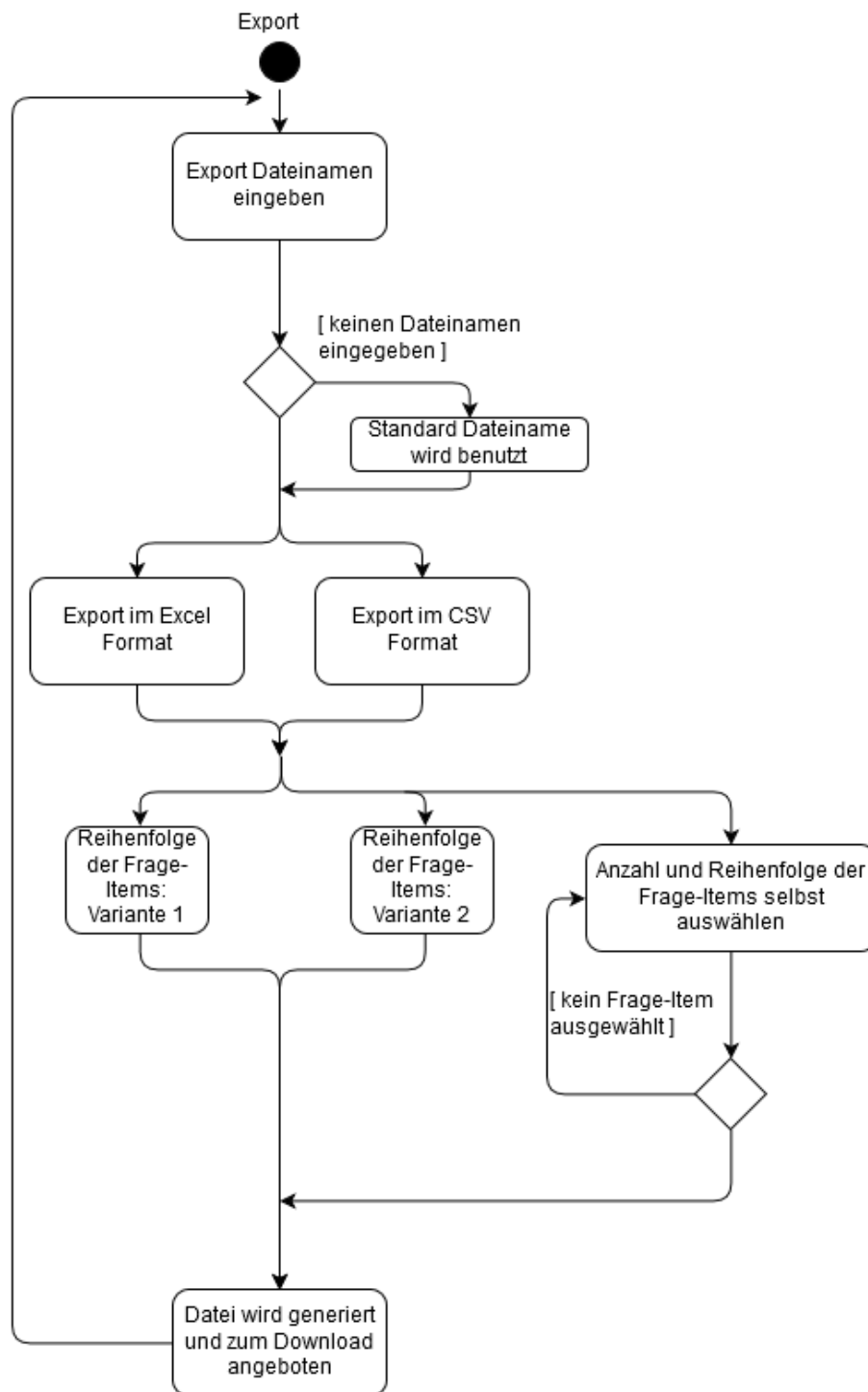
Der Benutzer kann entscheiden welche Daten er übernehmen will und ob er fehlerhafte Importdaten manuell korrigieren möchte. Mithilfe einer Fehlernavigation kann direkt zum Korrekturfeld eines fehlerhaften Eintrags gesprungen werden.

Stimmen zu einer GSR ID der Importeintrag und der korrespondierende Eintrag in der DB überein, so wird dieser in der Übersicht nicht angezeigt.

Nach abschließender Bearbeitung werden die ausgewählten Daten in die Datenbank importiert.

Für neu importierte GSR ID's wird automatisch ein neuer Datensatz mit Case ID angelegt und die zugehörigen Daten der importierten GSR ID eingetragen.

Bereits in der DB vorhandene GSR ID's werden mit den importierten Werten aktualisiert, falls der Benutzer diese für die Übernahme in die DB ausgewählt hat.



**Abb. 14:** Export von Daten aus der iStrokeDB

Für den Export von Daten aus der Datenbank (Abb. 14) bestimmt der Benutzer den Dateinamen und das Export-Dateiformat. Danach kann aus 2 Varianten zur Reihenfolge der Frage-Items ausgewählt werden oder die Anzahl und Reihenfolge der Frage-Items in einer neuen Ansicht selbst zusammengestellt werden. Nach erfolgreicher Auswahl wird die Exportdatei erzeugt und kann heruntergeladen werden.

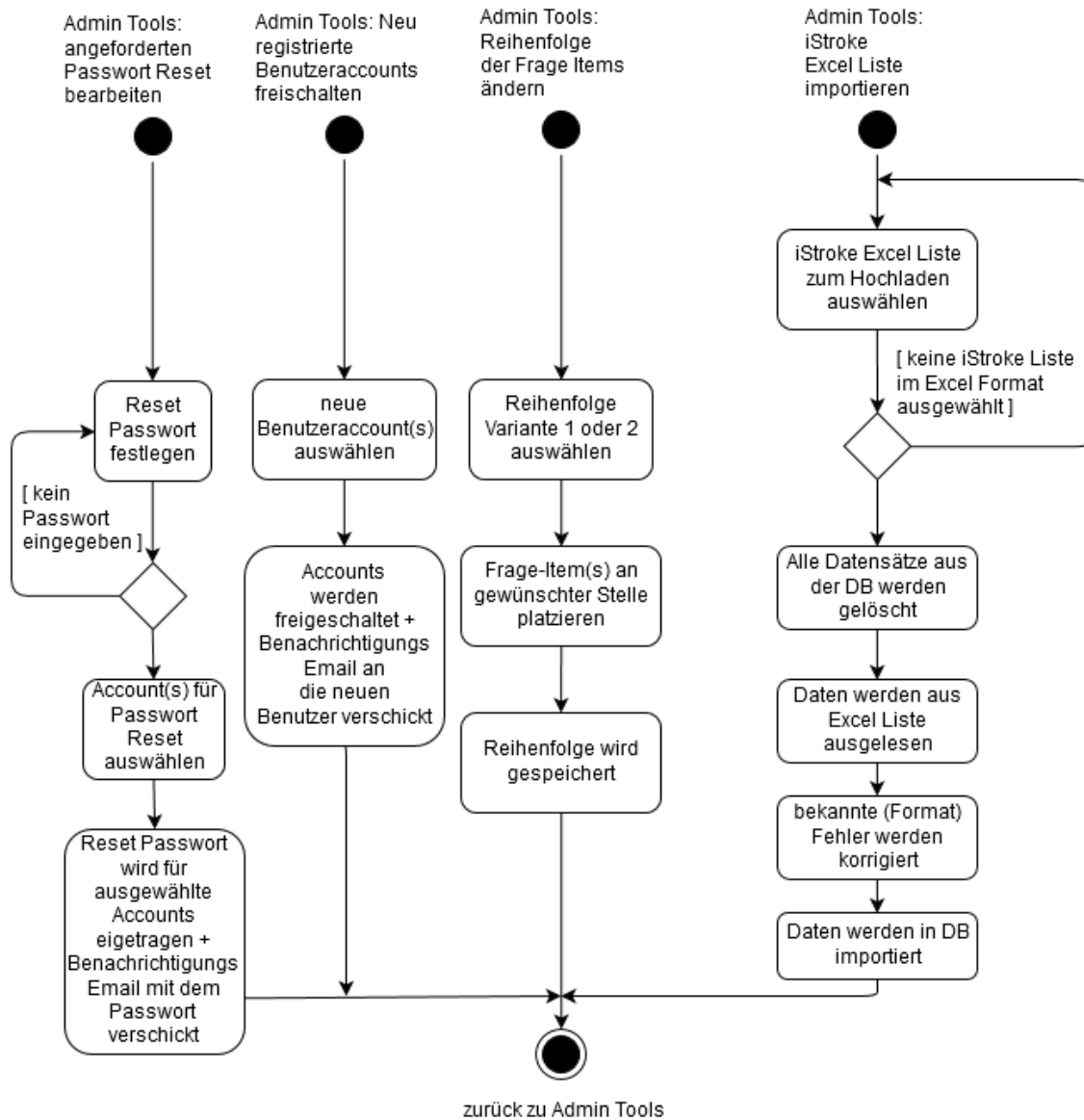
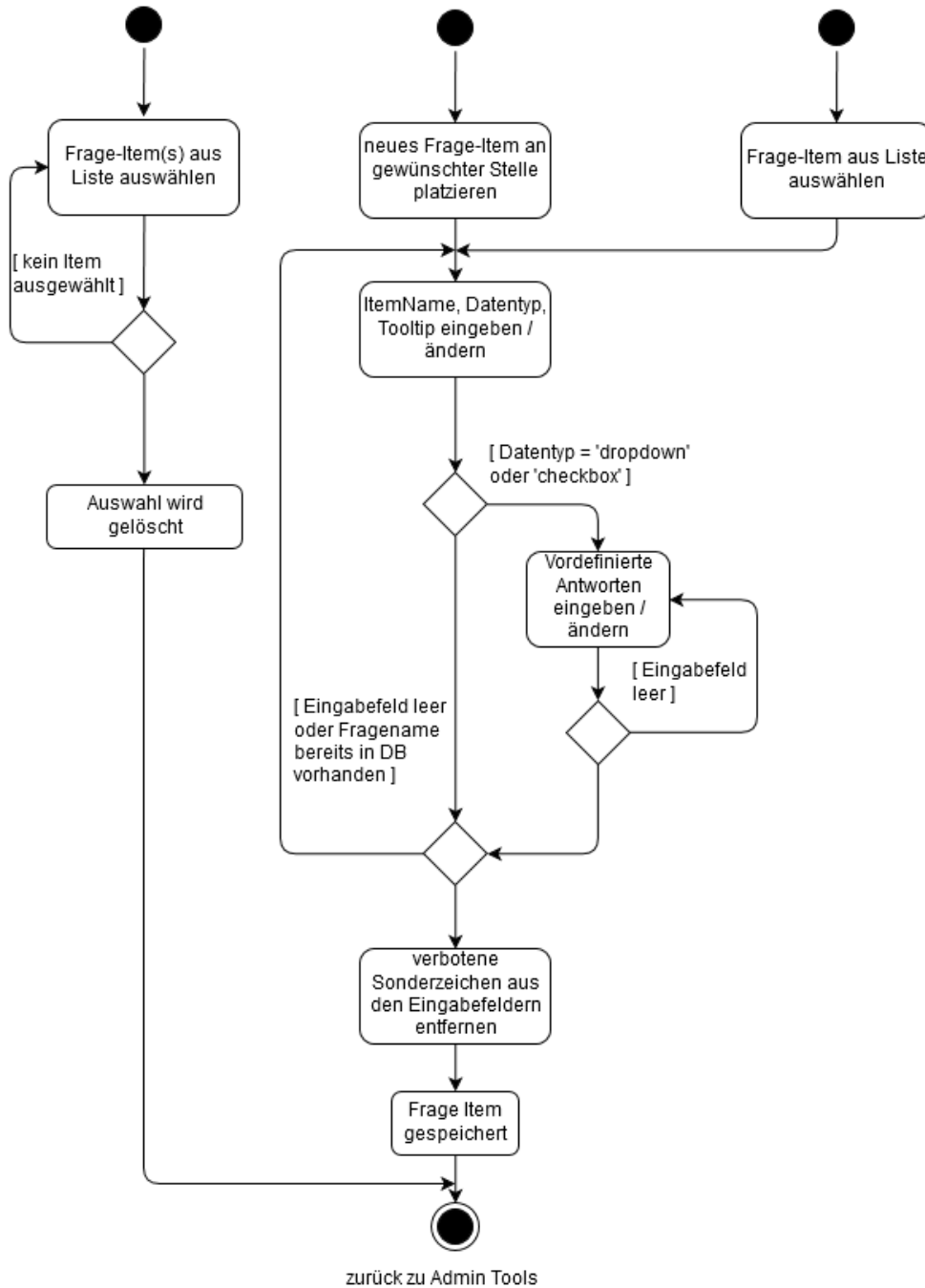


Abb. 15: Erweiterte Funktionen der Admin Oberfläche

Benutzer mit Admin-Status haben Zugriff auf die „Admin Tools“ Oberfläche (Abb. 15) mit weiteren Funktionen. Dort kann die Reihenfolge der Frage-Items von 2 Varianten verändert werden. Änderungen an Variante 1 wirken sich auf die Darstellung der Frage-Items in den Ansichten „Neuen Datensatz hinzufügen“ und „Datensatz ändern“ aus. Die Reihenfolge der Variante 2 hängt mit dem „GSR Import“ zusammen, hier werden nur Änderungen vorgenommen, falls sich Frage-Items in der GSR Datenbank geändert haben.

Für einen kompletten Import einer iStroke Excel-Liste wählt der Benutzer eine gültige Exceldatei aus. Anschließend werden alle vorhandenen Datensätze aus der DB gelöscht und mit den importierten ersetzt. Bekannte Formatfehler der Importeinträge werden beim Importvorgang korrigiert, bevor die Daten in die DB geschrieben werden. Wie in 2.2 beschrieben, musste beim Auslesen der Zelleneinträge beachtet werden, dass diese eine Excel-Formatierung für Datum oder Uhrzeit besitzen können.

Zusätzlich können in dieser Oberfläche neue Benutzeraccounts freigeschaltet und Passwort-Reset-Anfragen bearbeitet werden.



**Abb. 16:** Hinzufügen, Ändern und Löschen von Frage-Items in der Admin Oberfläche

Des Weiteren können Benutzer mit Admin-Status neue Frage-Items hinzufügen, verändern und löschen (Abb. 16). Grund hierfür kann eine Veränderung von Frage-Items in der GSR Datenbank sein. Nach dem Platzieren eines neuen oder Auswahl eines vorhandenen Frage-Items kann der Item-Name, Datentyp, Tooltip sowie eventuelle vordefinierte Antworten eingegeben und bearbeitet werden. Falls ein neu eingegebener Item-Name bereits in der DB vorhanden ist, muss eine entsprechende Korrektur stattfinden. Vor dem Speichern werden noch verbotene Sonderzeichen aus den Eingabefeldern entfernt bzw. ersetzt (z.B. Leerzeichen wird durch Unterstrich ersetzt). Das Löschen von 1 oder mehreren Frage-Items kann der Benutzer in einer separaten Ansicht durchführen.

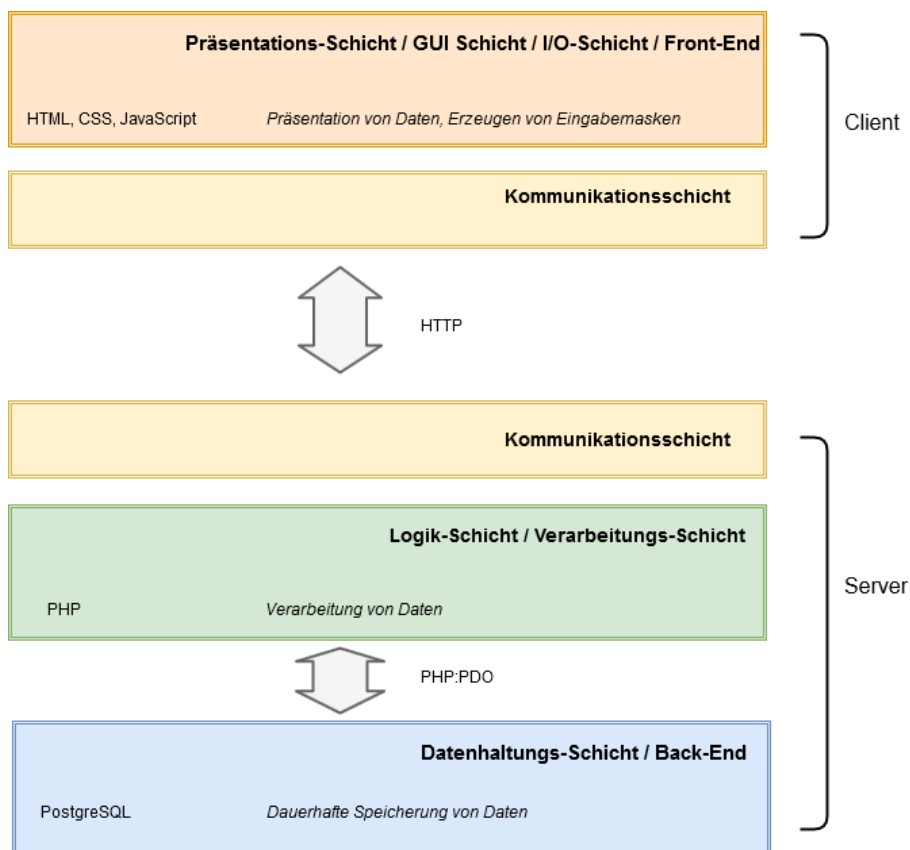
### 3.4 Gesamtarchitektur

Zur Veranschaulichung der Gesamtarchitektur wurde das Schichten-Modell gewählt.

Ziel war die Darstellung des Software-Systems (Abb. 17) als eine überschaubare Anzahl von Teilsystemen. Pro Schicht wurden miteinander verwandte Aspekte des Systems zusammengefasst. Jede Schicht stellt eine Abstraktion der in ihr enthaltenen Funktionalitäten dar und repräsentiert eine Teilaufgabe des Gesamtsystems (z.B. Datenspeicherung) [S. 290] [43].

Weiterhin musste beim Einteilen der Schichten berücksichtigt werden, dass es sich nicht um ein Einrechner-System (*stand-alone system*), sondern um ein verteiltes Client-Server System handelt. Durch die Verteilung auf einen Server-Rechner und mindestens einen Client-PC wird jeweils eine zusätzliche Schicht zur Kommunikation und Datenübertragung benötigt. [S. 298] [43].

Die Abhängigkeiten der Schichten zueinander folgt ebenfalls dem Client-Server Prinzip, d.h. eine Schicht stellt für die über ihr liegende Schicht Dienste zur Verfügung und kann auf Dienste der unter ihr liegenden Schicht zugreifen [S. 290] [43].



**Abb. 17:** Schichten-Modell eines Software Systems [S. 299] [43]

Da sich die Verarbeitungsschicht auf dem Server-Rechner befindet, spricht man auch von einer *Thin-Client*-Architektur [S. 299] [43]. Dadurch ist auf dem Client-Rechner bis auf die Installation eines Webbrowsers keine weitere Software nötig, um das System benutzen zu können.



### 3.5 Funktionskonzept

Ein primäres Ziel bei der Speicherung der Untersuchungsdaten war die fehlerfreie und einheitliche Übernahme der Benutzereingaben in die Datenbank.

Ein Datum beispielsweise wird je nach Land oder Herkunft des Benutzers in einem anderen Format eingegeben. Während in englischsprachigen Ländern das Format YYYY-MM-DD verbreitet ist, wird in Deutschland das Datum üblicherweise im Format DD.MM.YYYY geschrieben. Analog verhält es sich mit Uhrzeiten (13:00 Uhr vs. 01:00 p.m.). Diese unterschiedlichen Schreibweisen müssen sowohl bei Benutzereingaben, als auch beim Import von Daten aus anderen Quellen erkannt und vereinheitlicht werden.

Benutzereingaben werden in der *hypertext markup language* (HTML) üblicherweise über ein Formular entgegengenommen. Dabei stellt HTML diverse Möglichkeiten zur Verfügung, um diese zu validieren. Damit ein Eintrag wirklich dem gewünschten Format entspricht, reicht es in vielen Fällen nicht, einem HTML Eingabefeld nur das Attribut *type* (z.B. *type="date"* oder *type="number"*) zuzuweisen. Ebenso ist zu beachten, dass viele HTML Validierungsmöglichkeiten erst mit der HTML Version 5 eingeführt wurden und von älteren Browsern wie z.B. Internet Explorer 10 eventuell nicht vollständig unterstützt werden [44].

PHP selbst stellt vorgefertigte Funktionen bereit, um mittels POST Methode übertragene Daten auf Korrektheit zu überprüfen. Während des Registriervorgangs eines neuen Benutzers wurde beispielsweise die Funktion *filter\_var()* in Kombination mit dem *FILTER\_VALIDATE\_EMAIL* Filter verwendet, um eine eingegebene Email Adresse auf ihre Korrektheit hin zu überprüfen [45].

```
if (!(filter_var($email, FILTER_VALIDATE_EMAIL)))
{
    ?>
        <script>
            Message("Bitte eine gültige E-Mail-Adresse eingeben");
        </script>
    <?php
        $error = true;
    }
```

Möchte man nun zusätzlich überprüfen, dass alle eingegebenen Registrierungs-Emailadressen mit der Domain *@uke.de* enden, so müsste man die eingegebene Zeichenkette zerlegen und die einzelnen Bestandteile überprüfen.

Diese Überprüfung kann man allerdings auch in einer einzigen Codezeile mithilfe von *regular expressions* durchführen:

```
if (!(filter_var($email, FILTER_VALIDATE_EMAIL) AND
preg_match("/@uke.de$/i", $email)))
{
    ?>
        <script>
            Message("Bitte eine gültige E-Mail-Adresse (@uke.de) eingeben");
        </script>
    <?php
        $error = true;
    }
```

Als Werkzeug für eine detaillierte Fehlererkennung, -korrektur und -verhinderung wurden somit in der Mehrzahl der Fälle *regular expressions* (dt. Reguläre Ausdrücke) verwendet.

Da PHP eine serverseitige Programmiersprache ist, erfolgt die Fehlererkennung erst, nachdem die Daten zum Server geschickt wurden. Um fehlerhafte Benutzereingaben schon vor dem Absenden der Daten zum Server zu verhindern, bietet HTML für Eingabefelder das *pattern* (dt. Muster) Attribut an. Die Eingabe des Musters erfolgt in Form eines Regulären Ausdrucks. Wenn das Muster nicht eingehalten wird, erscheint sofort eine optische Markierung der fehlerhaften Eingabe und das Formular kann bis zu Korrektur nicht abgesendet werden.

Es wurde eine Funktion in PHP erstellt, um für alle Frage-Items des Fragenkatalogs sicherzustellen, dass Benutzereingaben zu dieser Frage nach dem Absenden fehlerfrei und einheitlich zum Server übertragen werden. Die Funktion bekommt Namen und Datentyp des Frage-Items übergeben und gibt einen regulären Ausdruck zurück, welcher bei der Generierung des Formulars an der entsprechenden Stelle in das *pattern* Attribut des Eingabefeldes geschrieben wird.

```
function column_input_validation($data_type, $column)
{
    if(preg_match("/^gser/i", $column))
    {
        // GSR ID Format
        return '^ (GSR) ([0-9]) {6}$';
    }
    elseif(preg_match("/^age/i", $column))
    {
        // Altersangabe bis max. 150 möglich
        return '^ (([0-9]) {1} | ([0-9] [0-9]) {1} | (1[0-4] [0-9]) {1} | (150))$';
    }
    elseif(preg_match("/admission.year/i", $column))
    {
        // Nur YYYY Eingabe möglich von 1900 bis 2199
        return '^ (19|20|21) \d{2}$';
    }
    elseif(preg_match("/admission.month/i", $column))
    {
        // Monatseingabe 1 bis 12 möglich
        return '^ (([1-9] {1}) | (1[0-2]) {1})$';
    }
    elseif(preg_match("/blood.pressure/i", $column))
    {
        // Blutdruck min. 10 max. 400
        return '^ (([1-9] [0-9]) {1} | ([1-3] [0-9] [0-9]) {1} | (400))$';
    }
    elseif(preg_match("/heart.rate/i", $column))
    {
        // Herzfrequenz min. 10 max. 500
        return '^ (([1-9] [0-9]) {1} | ([1-4] [0-9] [0-9]) {1} | (500))$';
    }
    elseif(preg_match("/0-42/i", $column))
    {
        // NIHSS on admission[0-42] min.0 max. 42
        return '^ (([0-9]) {1} | ([1-3] [0-9]) {1} | (4[0-2] {1}))$';
    }
    elseif($data_type == 'smallint')
    {
        // nur max. fünfstellige Zahlen möglich
    }
}
```

```

        return '[0-9]{1,5}';
    }
    elseif($data_type == 'integer')
    {
        // bis zu zehnstellige Zahlen möglich
        return '[0-9]{1,10}';
    }
    elseif($data_type == 'date')
    {
        // DD.MM.YYYY
        return
            '(0[1-9]|1[0-9]|2[0-9]|3[01])\.(0[1-9]|1[012])\.(19|20|21)\d{2}';
    }
    elseif($data_type == 'time without time zone')
    {
        // HH:MM
        return '(0[0-9]|1[0-9]|2[0-3])(:[0-5][0-9]){1}';
    }
    elseif($data_type == 'character(1)')
    {
        // 1 Zeichen erlaubt
        return '^.$';
    }
    else
    {
        // alle Zeichen erlaubt
        return '.*';
    }
}

```

Ein Vorteil der Funktion liegt in der beliebigen Erweiterbarkeit für zukünftige Änderungen am Fragenkatalog.

In einer ähnlichen Weise wurde beim Import von Daten in die *iStrokeDB* verfahren. Ziel einer weiteren Funktion war nicht nur die Erkennung von häufig auftretenden Fehlern sondern auch deren automatische Korrektur. Eine automatische Korrektur ist jedoch nur möglich, wenn der Fehler bzw. die Stelle des Fehlers bekannt ist. Für die Erkennung von Fehlermustern wurden erneut *regular expressions* benutzt.

Im Folgenden ein Ausschnitt aus der Funktion, der Fehler in eingelesenen Uhrzeiten erkennen und berichtigen soll:

```

if($data_type == 'time without time zone')
{
    // Korrektur für falsche Uhrzeitangaben
    // Zuerst wird nach einem Fehlermuster gesucht

    if(preg_match("/^\d:\d\d$/i", $value))
    {
        // falls eine vorangestellte Null fehlt
        // z.B. '1:35'

        $value = '0' . $value;
    }
    elseif(preg_match("/^\d\d;\d\d$/i", $value))
    {
        // falls anstatt Doppelpunkt ein Semikolon benutzt wurde

        $value = str_replace(";", ":", $value);
    }
    elseif(preg_match("/^\d\d:\d\d:\d\d$/i", $value) OR
    preg_match("/^\d:\d:\d\d$/i", $value))
    {
        // 'groin puncture time' im iStroke Excel Sheet
        // dort ist Uhrzeit im Format HH:MM:SS
        // PHPExcel:getFormatValue() kann z.B. '12:16:36', aber
        // auch '8:30:12' ergeben
        // String wird in Uhrzeit konvertiert
        // Format HH:MM

        $value = date("H:i", strtotime($value));
    }
    elseif(!preg_match("/^\d:\d:\d\d$/i", $value))
    {
        // z.B. '10:17 extern'
        // Uhrzeit wird aus String ausgelesen

        preg_match("/\d\d:\d\d/i", $value, $matches);

        $value = $matches[0];
    }

    return $value;
}

```

Hier wird ein übergebener Wert nacheinander mit vordefinierten Fehlermustern verglichen. Falls ein Fehlermuster erkannt wurde, wird ein passender Korrekturvorgang durchgeführt. Die Funktion kann bei neu auftretenden Fehlermustern entsprechend erweitert werden. Nicht bekannte bzw. erkannte Fehler müssen vom Benutzer selbstständig korrigiert werden.

Bei dieser Funktion ist zu beachten, dass durch die automatische Korrektur importierte Untersuchungsergebnisse verfälscht oder Informationen verloren gehen können. Dies wird vor allem in der letzten if – Anweisung deutlich, welche einen eingelesenen String automatisch auf die Uhrzeitangabe reduziert, falls er noch weitere Bemerkungen enthält. Welche Fehler automatisch korrigiert werden sollen, wird in 4.1 diskutiert.

Ein weiteres Problem beim Import von Daten bestand im Einlesen von Ergebnissen mit Mehrfachauswahl in Form von Checkboxen. Ein Eintrag mit Mehrfachauswahl im Benutzerinterface der GSR (*german stroke registry*) Datenbank (Abb. 18) wurde innerhalb einer GSR Export-Exceldatei in einer einzelnen Zelle (Tab. 8) abgespeichert.

Medication before admission

- ASS
- Clopidogrel
- Aggrenox
- Marcumar
- Eliquis® (Apixaban)
- Pradaxa® (Dabigatran)
- Xarelto® (Rivaroxaban)
- LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation)
- none
- Lixiana® (Edoxaban)

**Abb. 18:** Checkbox-Auswahl in der GSR Datenbank

16:55	Marcumar, ASS	no
21:44	None	no
22:39	LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation), ASS	no
13:49		yes

**Tab. 8:** Ausschnitt Checkbox-Ergebnis in GSR Export-Exceldatei

Die gesetzten Checkbox-Ergebnisse wurden also mit einem Komma und einem Leerzeichen separiert in die Zelle geschrieben. Hier tritt nun das Problem auf, dass einzelne Checkbox-Ergebnisse bereits die Zeichenkette „Komma + Leerzeichen“ enthalten. Dadurch kann dieser Eintrag nicht ohne weiteres mittels einer einfachen String-Teilungs-Funktion (z.B. *explode()* in PHP) anhand der „Komma + Leerzeichen“ Zeichenkette separiert werden kann.

Zur Lösung des Problems kamen wieder *regular expressions* in Kombination mit der PHP Funktion *preg\_split()* zum Einsatz. Für diesen Fall musste der reguläre Ausdruck, welcher das Trennzeichen zur Teilung des Strings beschreibt, mit einem Zusatz in Form eines *negated lookahead* erweitert werden.

```
if($data_type == 'checkbox')
{
    $chk_array = preg_split('/, (?!(zB|M1|M2|intracranial
    \(|transient|persistent)/', $value);
```

Dies ist ein angefügter Ausdruck innerhalb der *RegExp* Syntax, welcher von Klammern umschlossen mit den Startzeichen *?!* definiert wird. Dadurch wird eine *RegExp* Bedingung nur erfüllt, wenn nach dem Trennzeichen keine der im Ausdruck enthaltenen Zeichen(ketten) folgt [46].

Im o.g. Beispiel wird der Import-Wert nicht an der Stelle vor ‚zB‘ geteilt, da diese Zeichenkette im *negated lookahead* aufgeführt ist. Somit konnte der Separator „Komma + Leerzeichen“ zum Einlesen der Ergebnisse genutzt werden.

Beachtet werden muss jedoch, dass in Zukunft bei neu hinzukommenden Fragen mit Mehrfachauswahl in der GSR Datenbank die angegebenen Auswahlmöglichkeiten auf Vorhandensein von „Komma + Leerzeichen“ überprüft werden müssen und, falls erforderlich, der benutzte *negated lookahead* erweitert werden muss.

Wie in 1.2 beschrieben, wurde eine Möglichkeit zur Bildung von Daten-Kollektiven durch gezielte Auswahl und Eingrenzung von den in der Datenbank enthaltenen Daten gefordert.

Zur Lösung wurde eine Suchfunktion entworfen, welche eine beliebige Anzahl an Suchkriterien miteinander kombiniert, um ein gewünschtes Kollektiv von Daten zu erzeugen.

Dabei ergeben sich die Suchkriterien aus den Frage-Items (Abb. 19) des Fragenkatalogs.

Suchkriterien auswählen:

ID = 2016 Suchen

Weiteres Suchkriterium

UND  ODER

Age <= 70

Weiteres Suchkriterium

UND  ODER

Location\_of\_occlusion = A. cerebri media, M2

Weiteres Suchkriterium

**Abb. 19:** Suchmaske

Als Ergebnis der Suchanfrage wird eine Übersicht (Abb. 20) aller Datensätze erzeugt, die die eingegebenen Suchkriterien erfüllen. Über einen Link kann direkt in die Bearbeitungsansicht für einen ausgewählten Datensatz gewechselt werden oder die Suchergebnisse exportiert werden.

4 result(s) - Case ID zum Bearbeiten / Löschen auswählen:

Case ID	Progress	ID	ID alt	UKE case	GSR ID	Age	Location_of_occlusion
<a href="#">448</a>	64 / 108	uke_20160106_1949		563264637	GSR140054	64	A. cerebri media, M2
<a href="#">500</a>	65 / 108	uke_20160506_1527		564458535	GSR140127	48	A. cerebri media, M2
<a href="#">502</a>	71 / 108	uke_20160509_0313		564470316	GSR140125	55	A. cerebri media, M1 distal , A. cerebri media, M2
<a href="#">504</a>	73 / 108	uke_20160515_0920		564537114	GSR140106	58	A. cerebri media, M2 , A. cerebri media, M1 proximal

**Abb. 20:** Suchergebnis einer Suchanfrage. Zum Case zugehörige ID's (Spalte 3 bis 6) werden in den Suchergebnissen immer mit angezeigt. Die Spalte Progress gibt Auskunft über die Anzahl der bereits bearbeiteten Frage-Items.

Im Zuge der Benutzerfreundlichkeit muss kein exakter Suchbegriff eingegeben werden, es reichen auch Stichwörter, einzelne Zeichen oder man lässt das Suchfeld leer. Mithilfe des SQL Ausdrucks *LIKE* wird anschließend in der DB nach Treffern gesucht, wobei die Groß- /Kleinschreibung des Suchbegriffs keine Rolle spielt. War das Suchfeld leer, so werden alle nicht-leeren Einträge zum ausgewählten Suchkriterium angezeigt [47].

Bei der Eingabe des Suchkriteriums werden zudem Suchvorschläge zu passenden Frage-Items (Abb. 21) anhand der eingegebenen Zeichenkette angezeigt, um das gesuchte Frage-Item schnell zu finden. Ist das Eingabefeld für das Suchkriterium leer, so wird in der Suchvorschlagsliste der komplette Fragenkatalog aufgeführt.

The screenshot shows a search interface with the following elements:

- A header label: "Suchkriterien auswählen:"
- A search input field containing "ID" followed by an equals sign and a dropdown arrow.
- A blue "Suchen" button.
- A checked checkbox labeled "Weiteres Suchkriterium".
- Radio buttons for logical operators: "UND" (selected) and "ODER".
- A second search input field containing "iv" followed by an equals sign and a dropdown arrow.
- A dropdown menu listing search suggestions:
  - Living\_status\_before\_admission
  - iv-thrombolysis
  - iv-thrombolysis\_comments
  - Exclusion\_criteria\_iv-thrombolysis?
  - iv-thrombolysis\_dosis\_[0,9mg/\_kg\_KG]
  - iv-thrombolysis\_on\_site** (highlighted in blue)
  - iv-thrombolysis\_on\_site\_comments
  - iv\_thrombolysis\_time\_hh:mm
  - iv\_thrombolysis\_time\_comments
  - iv-thrombolysis\_drip\_and\_ship
  - iv-thrombolysis\_dosis\_stopped\_before\_intervention?

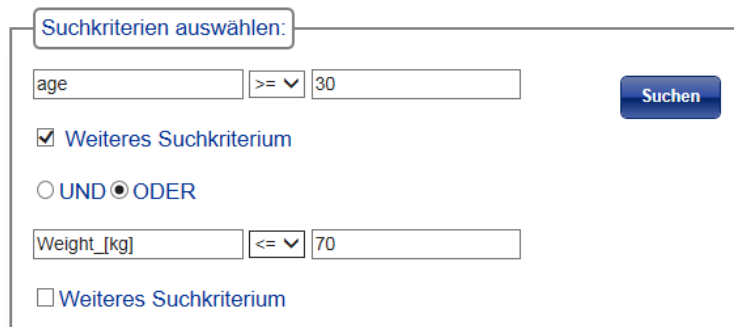
**Abb. 21:** Auswahl eines Suchkriteriums anhand von Suchvorschlägen.

Für numerische Suchkriterien (z.B. Alter, Gewicht) können die Operatoren *größer-gleich* oder *kleiner-gleich* verwendet werden, um die Suche weiter einzuzugrenzen. Zusätzlich dient der NICHT Operator (<> oder !=) dazu, eingegebene Suchbegriffe aus den Suchergebnissen auszuschließen.

Der für die Suchfunktion verwendete Suchalgorithmus lässt sich vereinfacht folgendermaßen beschreiben:

1. Ein Datensatz / Case wird über seinen Primärschlüssel (*Case ID*) eindeutig identifiziert. Für jedes einzelne Suchkriterium werden alle Datensätze gesucht, die das Kriterium erfüllen. Die Case ID's der gefundenen Datensätze werden als eine Menge zwischengespeichert.
2. Zur Verknüpfung von zwei Suchkriterien kann der Benutzer zwischen logischem UND bzw. ODER auswählen. Wenn die Suchkriterien mit UND verknüpft wurden, wird der SQL Ausdruck *INTERSECT* verwendet, um eine Schnittmenge zwischen den erzeugten Mengen von Case ID's der beiden Suchkriterien zu erzeugen. Als Ergebnis bleiben nur die Case ID's erhalten, welche beide Suchkriterien erfüllt haben. Bei einer ODER Verknüpfung wird der SQL Ausdruck *UNION* verwendet, welcher die Mengen der beiden Suchkriterien zu einer Gesamtmenge vereint. Beide SQL Ausdrücke entfernen zudem Duplikate im Ergebnis [48].
3. Diese Verknüpfungsoperationen werden für alle eingegebenen Suchkriterien durchgeführt. Die zum Schluss übriggebliebene Menge an Case ID's erfüllt somit alle Suchkriterien.
4. Schließlich werden für diese Case ID's die Ergebnisse anhand der eingegebenen Suchkriterien aus der DB ausgelesen und in einer Übersicht (Abb. 20) tabellarisch dargestellt.

Da die Formulierung einer Suchabfrage beliebig detailliert sein kann, wurde zusätzlich eine Möglichkeit zur Speicherung von Suchabfragen implementiert. Die einzelnen Suchkriterien und Operatoren sowie Suchbegriffe (Abb. 22) werden im JSON (*JavaScript Object Notation*) Format in der DB gespeichert. Dabei werden alle Suchparameter als eine einzelne, zusammenhängende Zeichenkette (Abb. 23) mit spezieller Notation in einer Tabellenzelle abgelegt.



**Abb. 22:** Beispiel Speicherung einer Suchanfrage

```
{"input_autocomplete_1":"Age","input_operator_1": ">=", "input_field_1": "30", "check_1": "on", "and_or_2": "UNION", "input_autocomplete_2": "Weight_[kg]", "input_operator_2": "<=", "input_field_2": "70", "hidden_amount_search_fields": "2"}
```

**Abb. 23:** Speicherung der Suchanfrage als Zeichenkette im JSON Format.

Wenn eine Suchabfrage geladen werden soll, wird die gespeicherte Zeichenkette von JavaScript mithilfe einer Parser Funktion in ein JavaScript Objekt umgewandelt.

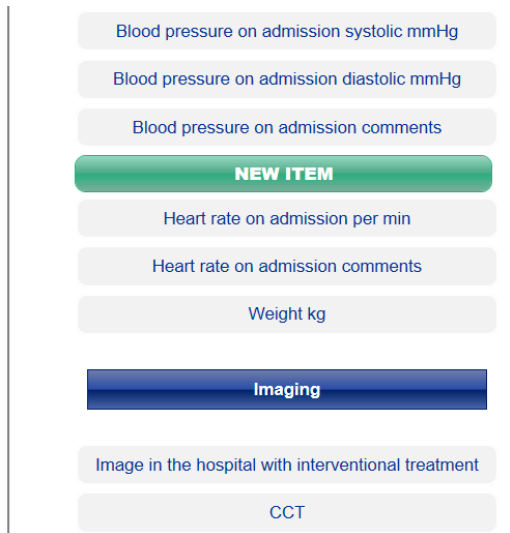
Der Zugriff auf das Objekt erfolgt ähnlich wie bei einem assoziativen Array. Die Daten sind als Schlüssel-Wert Paar (engl. *key-value pair*) abgespeichert [49]. Mithilfe des Schlüssels wird das HTML Element angesprochen, auf das der Wert des Schlüssels übertragen werden soll.

Für die Zukunft sind weitere Änderungen am Fragenkatalog der Datenbank zu erwarten. Dabei können neue Frage-Items hinzukommen, vorhandene können sich ändern oder auch wegfallen.

Damit diese Änderungen ohne Programmierkenntnisse von den Anwendern selbst durchgeführt werden können, wurde eine Möglichkeit geschaffen, den Fragenkatalog über die Weboberfläche selbstständig anzupassen. Zugriff auf diese Funktionen haben jedoch nur Anwender mit erweiterten Rechten, da durch das Löschen von Frage-Items auch assoziierte Untersuchungsdaten gelöscht werden.



Für die Auswahl der Items wurde mithilfe der JavaScript Erweiterung *jQuery UI* eine Liste aller vorhandenen Frage-Items generiert. In dieser können per Drag & Drop neue Items an der gewünschten Stelle eingefügt (Abb. 24) oder bereits vorhandene Items zum Verändern oder Löschen ausgewählt werden.



**Abb. 24:** Neues Item zum Fragenkatalog hinzufügen.

Nach erfolgreicher Platzierung oder Auswahl kann der Benutzer im nächsten Schritt die Parameter (Abb. 25) für das Item festlegen oder verändern.

The image shows a form titled 'Item Question ID: 17'. It contains three fields: 'Item Name:' with a text input field containing 'Admission\_time\_hh:mm', 'Item Datentyp:' with a dropdown menu showing 'Uhrzeit', and 'Item Tooltip:' with a text area containing two empty lines.

**Abb. 25:** Item Parameter bearbeiten.

Falls ein Item-Name verwendet wird, der bereits in der Datenbank existiert, so wird das Abspeichern verhindert. Ansonsten würden Datenbankabfragen, etwa bei der Suchfunktion, zu diesem Item keine eindeutigen Ergebnisse liefern.

Item Question ID: 106

Item Name:  
DICOM\_perfusion

Item Datentyp:  
DropDown Menü

Item Tooltip:  
[Empty field]

Antwort: performed

Antwort: not performed ✘

Antwort: saved ✘

Antwort: processed ✘

[Pencil icon]

**Abb. 26:** Vordefinierte Antwortmöglichkeiten für ein Item festlegen.

Falls Eingrenzungen bei den Eingaben gewünscht sind, kann für Items mit mehrfacher Checkbox Auswahl oder Drop-Down Menü eine beliebige Anzahl an vordefinierten Antworten (Abb. 26) festgelegt werden. Die Eingabe von *RegExp* Mustern für zusätzliche Restriktionen ist in dieser Ansicht jedoch nicht möglich.

Zusätzlich kann die Reihenfolge der Frage-Items im Fragenkatalog über eine weitere Ansicht (Abb. 27) vom Anwender ebenfalls per Drag & Drop angepasst werden.

Reihenfolge der Items ändern:

General Information

ID

ID alt

UKE case

GSR ID

Soarian NRAD

Soarian NL

**Abb. 27:** Reihenfolge der Items im Fragenkatalog verändern.

### 3.6 Testpläne und -ergebnisse

Die Hauptziele von Software-Tests sind zum einen die Qualitätssicherung der entwickelten Software und der Nachweis über die Erfüllung der Anforderungen [S. 17] [50].

Die ersten Tests erfolgten bereits während der Entwicklung der Software, um die Anzahl an enthaltenen Fehlern durchgehend so gering wie möglich zu halten. Nachdem eine einzelne Programmkomponente fertiggestellt war, wurde deren Lauffähigkeit und Funktionalität vom Entwickler selbst getestet.

Nachdem ein funktionierender Prototyp des Systems bereitstand, wurden ebenfalls Tests von Seiten der späteren Anwender durchgeführt. Dies ist besonders wichtig, da Anwender meistens eine völlig andere Sicht auf das System haben und es auch anders benutzen als der Entwickler selbst. Dadurch erhält man wertvolles Feedback zum Thema Gebrauchstauglichkeit der Benutzeroberfläche. Zudem werden bei Anwendertests am ehesten Lücken und Unvollständigkeiten in den Anforderungen an das System aufgedeckt [S. 15] [50], welche dann zeitnah behoben werden können.

Der vollständige Test eines komplexen Software-Systems ist jedoch kaum machbar, da man theoretisch jede mögliche Kombination von Eingangsparametern durchtesten müsste [S. 11] [50]. Stattdessen werden stichprobenartige Testfälle erstellt und durchgeführt, die am ehesten der Praxis entsprechen. Wenn diese Testfälle fehlerfrei ablaufen, heißt das allerdings nicht, dass die Software fehlerfrei ist.

Zur Erfüllung der o.g. Ziele wurde ein Testplan erstellt, welcher zunächst die grundlegende Funktionalität des Systems (Regelfall) beweisen soll. Die Testfälle orientieren sich dabei an der in 3.1 vorgestellten Anforderungsliste.

Anschließend werden bewusst Fehler (Grenzfälle) bei der Eingabe und Bedienung der Weboberfläche provoziert.

Der spätere Einsatzort - ein KIS-PC im Intranet der NRAD mit installiertem Internet Explorer 10 – stand als Testumgebung wegen der täglichen Nutzung durch anderes Personal nicht zur Verfügung. Stattdessen wurden die Testfälle mit einem separat installiertem Internet Explorer 10 durchgeführt.

Aufgrund von Zeit- und Personalmangel wurde der Testplan vom Entwickler selbst erstellt und durchgeführt. Es ist jedoch empfehlenswert, dass diese Aufgabe immer von jemandem durchgeführt wird, der nicht zum Entwicklerteam gehört [S. 6] [50].

Während der Durchführung des Testplans wurden vereinzelt Fehler bzw. Schwachstellen entdeckt, welche direkt bei Auffindung behoben wurden. Jedoch verliefen die meisten Testfälle weitestgehend erfolgreich.

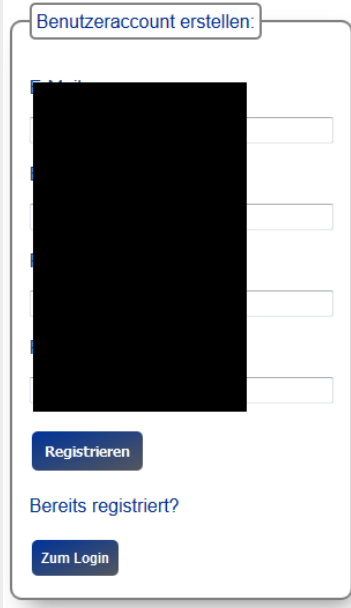
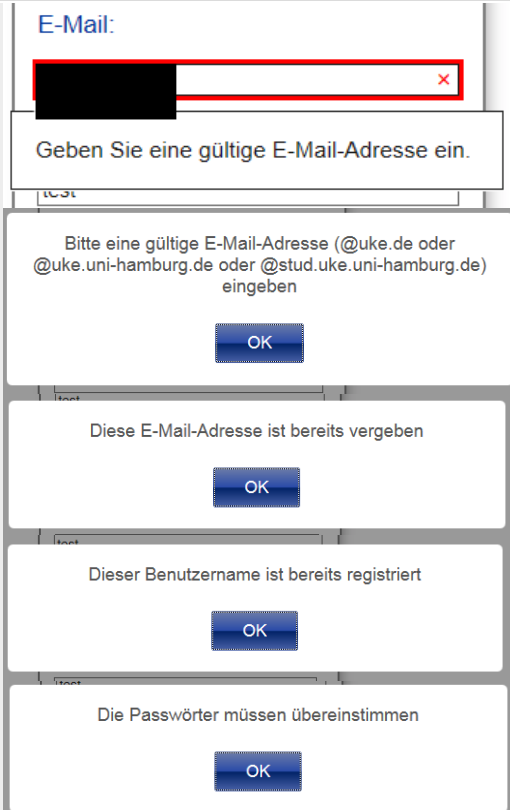
An dieser Stelle sollen nun exemplarisch zwei Testfälle aus dem Testplan vorgestellt und erläutert werden. Der vollständige Testplan ist im Anhang aufgeführt.

Test ID #T_0101	
Prozedur / Eingabe	Registrieren eines neuen Benutzeraccounts
Eingabedaten	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;">           Email: <a href="mailto:test@uke.de">test@uke.de</a>            Benutzername: test            Passwort: testpasswort            Passwort 2: testpasswort         </div> <div style="width: 35%; border: 1px solid gray; padding: 5px;"> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;">             Benutzeraccount erstellen:           </div> <div style="margin-bottom: 5px;">             E-Mail:             <input style="width: 100%;" type="text"/> </div> <div style="margin-bottom: 5px;"> <input style="width: 100%; background-color: black; height: 20px;" type="text"/> </div> <div style="margin-bottom: 5px;"> <input style="width: 100%; background-color: black; height: 20px;" type="text"/> </div> <div style="margin-bottom: 5px;"> <input style="width: 100%; background-color: black; height: 20px;" type="text"/> </div> <div style="margin-bottom: 5px;"> <input style="width: 100%; background-color: black; height: 20px;" type="text"/> </div> <div style="text-align: center; margin-bottom: 5px;"> <input type="button" value="Registrieren"/> </div> <div style="text-align: center; margin-bottom: 5px;"> <a href="#">Bereits registriert?</a> </div> <div style="text-align: center;"> <input type="button" value="Zurück zum Login"/> </div> </div> </div>
Erwarteter Zustand / Ausgabe	Erfolgreiche Erstellung des Benutzeraccounts mit den eingegebenen Daten.
Test Ergebnis	bestanden <div style="border: 1px solid gray; padding: 5px; margin-top: 10px; width: fit-content; margin-left: auto;">             Benutzeraccount erfolgreich erstellt             <div style="text-align: center; margin-top: 5px;"> <input type="button" value="Zum Login"/> </div> </div>
Kommentar	

**Tab. 9:** Testfall: Grundfunktionalität

Jeder Testfall (Tab. 9) besitzt eine eindeutige ID. Als erstes wird die Prozedur beschrieben, welche getestet werden soll und die verwendeten Eingabedaten aufgeführt. Dann wird der erwartete Soll-Zustand bzw. Ausgabe des Systems beschrieben. Erst jetzt wird mit der Durchführung der Prozedur begonnen und das Testergebnis bzw. der Ist-Zustand dokumentiert.

Nach dem erfolgreichen Testen einer Grundfunktionalität wurden anschließend so viele Fehlerszenarien (Tab. 10) wie möglich durchgetestet, um die Robustheit des Systems zu gewährleisten.

Test ID #T_0102	
Prozedur / Eingabe	Unzulässige Eingaben während des Registriervorgangs
Eingabedaten	<p>Email: <a href="mailto:test@web.de">test@web.de</a> &amp; <a href="mailto:test@uke.de">test@uke.de</a> &amp; test.uke.de            Benutzername: test            Passwort: testpasswort            Passwort 2: testpasswort2</p> 
Erwarteter Zustand / Ausgabe	Ausgabe der Fehlermeldungen für falsches Email Adressen Format, falsche Email Domain, bereits registrierte Email Adresse, bereits registrierter Benutzername, Passwort ist ungleich Passwort 2
Test Ergebnis	<p>bestanden</p> 
Kommentar	Leergelassene Eingabefelder sind ebenfalls unzulässige Eingaben und erzeugen entsprechende Fehlermeldungen, welche hier aus Platzgründen nicht aufgeführt sind.

Tab. 10: Testfall: Provozieren von Falscheingaben

## 4. Diskussion

### 4.1 Vergleich der Ergebnisse mit Aufgabenstellung

Durch die Eingabe der Untersuchungsdaten über die Weboberfläche werden die Datensätze in Zukunft einheitliche Daten enthalten und somit besser auswertbar und vergleichbar sein.

Mit der Suchfunktion kann gezielt nach Informationen gesucht und eine Eingrenzung der Suchergebnisse vorgenommen werden. Ebenso sind individuelle Daten-Kollektive für gesonderte Zwecke schnell und einfach aus der Datenbank exportierbar.

Zudem können mithilfe des CSV Dateiformats exportierte Daten leicht in andere Anwendungen / Systeme importiert werden.

Änderungen an Datensätzen werden automatisch protokolliert und sind in einer Historie einsehbar. Ältere Einträge können somit nichtmehr verloren gehen. Vorgenommene Änderungen sind immer mit einem registrierten Datenbankbenutzer assoziiert und somit nichtmehr anonym.

Durch die individuell angepasste, benutzerfreundliche Navigation ist ein insgesamt schnellerer Workflow möglich. Dies gilt insbesondere für den Import von Daten aus anderen Quellen (*GSR*) und deren Abgleich mit korrespondierenden Daten aus der *iStroke* Datensammlung, was bisher sehr zeitaufwendig war.

Durch das verteilte Client-Server System kann die Datensammlung nun von mehreren Benutzern gleichzeitig bearbeitet werden.

Falls in Zukunft Änderungen am Fragenkatalog auftreten, können diese über die Weboberfläche schnell und komfortabel umgesetzt werden. Zusätzlicher Programmieraufwand von Entwicklerseite ist hierfür nicht erforderlich. Ausgenommen sind Änderungen an Frage-Items, bei denen ein Eingabemuster in Form einer *RegExp* zwingend nötig ist. Hierfür muss die in 3.5 beschriebene Funktion zur Eingabvalidierung erweitert werden.

Bei der Migration der Daten von der Excel-Liste zur Datenbank muss zusätzlich beachtet werden, dass die in 3.5 vorgestellte Funktion zur Fehler-Autokorrektur nur bekannte Fehler automatisch korrigiert. Ein fehlerfreier Import aller Daten ist somit nicht garantiert. Nicht erkannte Fehler müssen somit bei Auffindung manuell korrigiert werden.

Eine automatische Korrektur von fehlerhaften Untersuchungsergebnissen ohne vorherige Absprache mit den verantwortlichen Ärzten ist zudem nicht zulässig, da durch die automatische Veränderung eine Verfälschung / Informationsverlust des Eintrags entstehen kann.

Um trotzdem eine möglichst fehlerfreie Migration der Daten zu erreichen, wird die *iStroke* Excel-Liste vor der endgültigen Umstellung unter Absprache manuell aufbereitet und für den Import an die Datenbank angepasst, sodass keine Informationen verfälscht werden oder verloren gehen.

Zur besseren Wartbarkeit, Übersichtlichkeit und Erweiterung des Programmcodes ist eine konsequent Objekt-Orientierte Programmierung angebracht. Aufgrund von fehlender Programmiererfahrung wurde dieses Konzept nicht angewandt.

Insgesamt wurden die Anforderungen der Stakeholder erfolgreich umgesetzt und die *iStrokeDB* zum Projektende in einer Abschlusspräsentation abgenommen.

## 4.2 Ausblick / mögliche Erweiterung

Bevor das erstellte System die bestehende *iStroke* Excel-Liste vollständig ablösen kann, sind Langzeittests auf Gebrauchstauglichkeit, Stabilität, und korrekte Funktionalität nötig. Erst wenn das System über einen längeren Zeitraum hinweg stabil funktioniert, keinen Datenverlust verursacht und sich für den Praxisgebrauch bewährt hat, kann eine vollständige Umstellung erfolgen.

Um einen eventuell entstehenden Datenverlust aufgrund von Fehlfunktionen abzufangen, wird vom Serveradministrator der NRAD eine Prozedur zur täglichen Erstellung einer Sicherungskopie der aktuellen Datenbankinstanz eingerichtet.

Im Laufe der aktiven Nutzung des Systems sind weitere Änderungswünsche der Nutzer und bisher nicht entdeckte Fehler des Systems zu erwarten, welche bei Auftreten nachträglich bearbeitet werden müssen.

Der Einbau einer Funktion für eine statistische Analyse und Auswertung von gespeicherten Daten zur Ermittlung von Leistungskennzahlen (z.B. *door-to-imaging* Zeit, Rekanalisationsrate) ist für das Qualitätsmanagement der NRAD von Interesse [S. 10] [14].

Zusätzlich sollen in Zukunft über eine weitere Schnittstelle automatisch DICOM Metadaten (wie Alter, Geschlecht, Uhrzeit der Bildaufnahme) in die Datenbank eingespielt werden. Beim Aufruf der Startseite erscheint dann eine entsprechende Meldung, dass neue Datensätze mit importierten Metadaten angelegt wurden, welche anschließend vervollständigt werden können.

## Literaturverzeichnis

1. Geschäftsbericht und Jahrbuch 2015. [Online].; 2015 [cited 2017 Februar 15. Available from: <https://www.uke.de/dateien/einrichtungen/unternehmenskommunikation/dokumente/gesch%C3%A4ftsberichte/gesch%C3%A4ftsbericht-2015.pdf>.
2. Über uns - Das UKE. [Online].; 2016 [cited 2017 Februar 15. Available from: <https://www.uke.de/allgemein/ueber-uns/das-uke/index.html>.
3. Siemens inside: health IT SPECIAL. Mit Soarian zum digitalen Krankenhaus. [Online].; 2012 [cited 2017 Februar 15. Available from: [https://www.ucm.de/wp-content/uploads/2015/05/Siemens-inside\\_health-16-April-2012-S68-71-Anwenderbericht\\_Soarian-Clinicals-UKE-HIMSS-Award.pdf](https://www.ucm.de/wp-content/uploads/2015/05/Siemens-inside_health-16-April-2012-S68-71-Anwenderbericht_Soarian-Clinicals-UKE-HIMSS-Award.pdf).
4. About HIMSS Analytics | HIMSS Analytics. [Online]. [cited 2017 Februar 15. Available from: <http://www.himssanalytics.org/about>.
5. UKE - Neuroradiologische Diagnostik und Intervention. [Online].; 2016 [cited 2017 Februar 16. Available from: <https://www.uke.de/kliniken-institute/kliniken/neuroradiologische-diagnostik-und-intervention/index.html>.
6. UKE - Neuroradiologische Diagnostik und Intervention - Zahlen & Fakten. [Online].; 2016 [cited 2017 Februar 16. Available from: <https://www.uke.de/kliniken-institute/kliniken/neuroradiologische-diagnostik-und-intervention/%C3%BCber-die-klinik/zahlen-fakten/index.html>.
7. UKE - Neuroradiologische Diagnostik und Intervention - Forschung. [Online].; 2015 [cited 2017 Februar 16. Available from: <https://www.uke.de/kliniken-institute/kliniken/neuroradiologische-diagnostik-und-intervention/forschung/index.html>.
8. UKE - Neuroradiologische Diagnostik und Intervention - Studien. [Online].; 2015 [cited 2017 Februar 16. Available from: <https://www.uke.de/kliniken-institute/kliniken/neuroradiologische-diagnostik-und-intervention/forschung/studien/index.html>.
9. dgn.org - Behandlung des akuten Schlaganfalls: Neue Leitlinie empfiehlt interventionelle Thrombektomie. [Online].; 2016 [cited 2017 März 29. Available from: [https://www.dgn.org/images/red\\_pressemitteilungen/2016/PM\\_160412\\_DSG\\_DGN\\_DGNR\\_LL\\_Akutversorgung.pdf](https://www.dgn.org/images/red_pressemitteilungen/2016/PM_160412_DSG_DGN_DGNR_LL_Akutversorgung.pdf).
10. Zeyfang A, Hagg-Grün U, Nikolaus T. Basiswissen Medizin des Alterns und des alten Menschen. 2nd ed.: Springer Berlin Heidelberg; 2013.
11. Berlit P. Basiswissen Neurologie. 6th ed.: Springer-Verlag Berlin Heidelberg; 2014.
12. Schwab S, Schellinger P, Unterberg A, Werner C, Hacke W. NeuroIntensiv. 3rd ed.: Springer-Verlag Berlin Heidelberg; 2015.
13. dgn.org - Die mechanische Thrombektomie: eine Revolution in der Schlaganfalltherapie. [Online].; 2015 [cited 2017 März 29. Available from: [https://www.dgn.org/images/red\\_pressemitteilungen/2015/150925\\_PM\\_Thrombektomie\\_fina\\_l\\_ok.pdf](https://www.dgn.org/images/red_pressemitteilungen/2015/150925_PM_Thrombektomie_fina_l_ok.pdf).



14. Ringleb P, Veltkamp R. Akuttherapie des ischämischen Schlaganfalls – Rekanalisierende Therapie (Ergänzung 2015). [Online].; 2015 [cited 2017 April 21. Available from: [https://www.dgn.org/images/red\\_leitlinien/LL\\_2015/PDFs\\_Download/030140\\_LL\\_akuter-ischaeemischer-schlaganfall\\_final.pdf](https://www.dgn.org/images/red_leitlinien/LL_2015/PDFs_Download/030140_LL_akuter-ischaeemischer-schlaganfall_final.pdf).
15. DeGIR. [Online]. [cited 2017 Februar 20. Available from: <http://www.degir.de/de-DE/2537/ziele-der-gesellschaft>.
16. Ubuntu Server - for scale out workloads | Ubuntu. [Online]. [cited 2017 Februar 21. Available from: <https://www.ubuntu.com/server>.
17. About the Apache HTTP Server Project - The Apache HTTP Server Project. [Online]. [cited 2017 Februar 21. Available from: [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html).
18. PHP: Was ist PHP? - Manual. [Online]. [cited 2017 Februar 21. Available from: <http://php.net/manual/de/intro-what-is.php>.
19. PostgreSQL: About. [Online]. [cited 2017 Februar 21. Available from: <https://www.postgresql.org/about/>.
20. phpPgAdmin - Start. [Online]. [cited 2017 Februar 21. Available from: <http://phpPgAdmin.sourceforge.net/doku.php>.
21. PHP: Einführung - Manual. [Online]. [cited 2017 Februar 21. Available from: <http://php.net/manual/de/intro.pdo.php>.
22. GitHub - PHPOffice/PHPExcel. [Online]. [cited 2017 Februar 21. Available from: <https://github.com/PHPOffice/PHPExcel>.
23. jQuery. [Online]. [cited 2017 Februar 21. Available from: <https://jquery.com/>.
24. jQuery UI. [Online]. [cited 2017 Februar 21. Available from: <http://jqueryui.com/>.
25. SweetAlert2. [Online]. [cited 2017 Februar 21. Available from: <https://limonte.github.io/sweetalert2/>.
26. Virtual Box - Chapter 1. First steps. [Online]. [cited 2017 Februar 21. Available from: <https://www.virtualbox.org/manual/ch01.html>.
27. Louis D, Müller P. Das Java 6 Codebook. 1st ed.: Addison-Wesley Verlag; 2007.
28. PHP: fgetcsv - Manual. [Online]. [cited 2017 Februar 17. Available from: <http://php.net/manual/de/function.fgetcsv.php>.
29. Yakov S. RFC 4180 - Definition of the CSV Format. [Online].; 2005 [cited 2017 Februar 17. Available from: <https://tools.ietf.org/html/rfc4180#section-2>.
30. Wagner K, Hüttl T, Backin D. Einführung Wirtschaftsinformatik: Gabler Verlag | Springer Fachmedien Wiesbaden GmbH.
31. Seemann J, von Gudenberg JW. Software Entwurf mit UML 2. 2nd ed.: Springer-Verlag Berlin Heidelberg; 2006.

32. Schäling B. Highscore - Der moderne Softwareentwicklungsprozess mit UML. [Online]. [cited 2017 April 11. Available from: <http://www.highscore.de/uml/>.
33. Kleuker S. Grundkurs Software-Engineering mit UML. 3rd ed.: Vieweg + Teubner; 2010.
34. Wagenknecht C, Hielscher M. Formale Sprachen, abstrakte Automaten und Compiler. 2nd ed.: Springer Vieweg; 2014.
35. Reimers N. Reguläre Ausdrücke - PHP lernen. [Online]. [cited 2017 März 1. Available from: <https://www.php-einfach.de/php-tutorial/regulaere-ausdruecke/>.
36. JavaScript Regular Expressions. [Online]. [cited 2017 März 1. Available from: [https://www.w3schools.com/js/js\\_regexp.asp](https://www.w3schools.com/js/js_regexp.asp).
37. Goll J. Methoden und Architekturen der Softwaretechnik. 1st ed.: Vieweg+Teubner; 2011.
38. Meier A. Relationale und postrelationale Datenbanken. 7th ed.: Springer Berlin Heidelberg; 2010.
39. Unterstein M, Matthiessen G. Relationale Datenbanken und SQL in Theorie und Praxis. 5th ed.: Springer Berlin Heidelberg; 2012.
40. Meier A, Kaufmann M. SQL- & NoSQL-Datenbanken. 8th ed.: Springer-Verlag Berlin Heidelberg; 2016.
41. van Randen HJ, Bercker C, Fiendl J. Einführung in UML - Analyse und Entwurf von Software: Springer Vieweg; 2016.
42. PostgreSQL 9.6.2 Documentation: 8.1. Numeric Types. [Online]. [cited 2017 März 3. Available from: <https://www.postgresql.org/docs/9.6/static/datatype-numeric.html>.
43. Goll J. Architektur- und Entwurfsmuster der Softwaretechnik. 2nd ed.: Springer Vieweg; 2014.
44. input - HTML | MDN. [Online].; 2017 [cited 2017 März 10. Available from: [https://developer.mozilla.org/de/docs/Web/HTML/Element/Input#Browser\\_compatibility](https://developer.mozilla.org/de/docs/Web/HTML/Element/Input#Browser_compatibility).
45. PHP: filter\_var - Manual. [Online]. [cited 2017 März 10. Available from: <http://php.net/manual/de/function.filter-var.php>.
46. Regular Expressions - JavaScript | MDN. [Online]. [cited 2017 März 13. Available from: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular\\_Expressions#special-negated-look-ahead](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions#special-negated-look-ahead).
47. PostgreSQL: Documentation: 9.5: Pattern Matching. [Online]. [cited 2017 März 16. Available from: <https://www.postgresql.org/docs/9.5/static/functions-matching.html>.
48. PostgreSQL: Documentation: 9.5: Combining Queries. [Online]. [cited 2017 März 18. Available from: <https://www.postgresql.org/docs/9.5/static/queries-union.html>.
49. JSON. [Online]. [cited 2017 März 19. Available from: <http://www.json.org/json-de.html>.
50. Witte F. Testmanagement und Softwaretest: Springer Vieweg; 2016.

## Anhänge

Die komplette Anforderungsliste aus 3.1 wird folgend aufgeführt:

ID	Beschreibung der Anforderung
#TR_0001	Die Benutzeroberfläche soll über Internet Explorer 10 und 11 in vollem Umfang benutzbar sein.
#FA_0001	Untersuchungsdatensätze sollen fallbasiert mit einer Case ID gespeichert werden.
#FA_0002	Zum Fall gehörende, unveränderliche ID's sollen zusammen mit der Case ID in einer eigenen Tabelle gespeichert werden.
#FA_0003	Alle anderen Untersuchungsergebnisse sollen mit Zeitstempel der Eintragung und Versionsnummer in einer eigenen Tabelle abgespeichert werden.
#FA_0004	Alle Untersuchungsfragen (Frage-Items) sollen als Fragenkatalog in einer eigenen Tabelle gespeichert werden.
#FA_0005	Vordefinierte, feste Antwortmöglichkeiten für die Untersuchungsfragen sollen in einer eigenen Tabelle abgespeichert werden.
#FA_0006	Alle Daten über die registrierten Benutzer soll in einer eigenen Tabelle abgespeichert werden.
#FA_0007	Nachuntersuchungen ( <i>Follow-Ups</i> ) sollen in einer eigenen Tabelle abgespeichert werden.
#FA_0100	Benutzer sollen sich selbstständig registrieren können.
#FA_0101	Eine eingegebene Email Adresse beim Registriervorgang soll einem gültigen Email Format entsprechen.
#FA_0102	Die Eingabe einer bereits vergebenen Email Adresse beim Registriervorgang soll eine entsprechende Fehlermeldung erzeugen.
#FA_0103	Die Eingabe eines bereits vergebenen Benutzernamens beim Registriervorgang soll eine entsprechende Fehlermeldung erzeugen.
#FA_0104	Ein Benutzer Password soll beim Registriervorgang zweimal eingegeben werden.
#FA_0105	Zwei unterschiedliche Passwort Eingaben beim Registriervorgang sollen eine entsprechende Fehlermeldung erzeugen.
#QA_0001	Das Passwort für jeden Benutzer soll verschlüsselt in der Datenbank gespeichert werden.
#FA_0106	Der Administrator soll neu registrierte Benutzer freischalten, bevor diese sich einloggen können.
#FA_0107	Nur registrierte und freigeschaltete Benutzer sollen Zugriff auf die Benutzeroberfläche haben.
#FA_0108	Benutzer sollen sich selbstständig ein- und ausloggen können.
#FA_0109	Eine falsche Benutzername + Passwort Kombination beim Login sollen eine entsprechende Fehlermeldung erzeugen.
#FA_0110	Ein vergessenes Benutzerpassword soll vom Admin im Rahmen einer Passwort-Reset Anfrage mit einem Ersatz Passwort ersetzt werden können
#FA_0111	Eingabe einer nicht registrierten Benutzer Email bei der Anfrage soll eine entsprechende Fehlermeldung erzeugen.

#FA_0112	Das Ersatz Passwort zum Neu Vergeben des Passworts soll automatisch an die registrierte Benutzer Email versendet werden.
#FA_0200	Die 10 neusten Datensätze sollen auf der Startseite in einer Tabelle angezeigt werden.
#FA_0201	Benutzer sollen Untersuchungsdatensätze anhand von ein oder mehreren Suchkriterien suchen können
#FA_0202	Die Suchkriterien sollen über UND / ODER Ausdrücke miteinander kombinierbar sein.
#FA_0203	Die Suchkriterien sollen sich aus den Untersuchungsfragen im Fragenkatalog ergeben.
#FA_0204	Ein nicht vollständig eingegebenes Suchkriterium soll automatisch vervollständigt werden.
#FA_0205	Ein leer gelassenes Suchkriterienfeld soll automatisch mit einem Standardwert versehen werden.
#SA_0001	Die durch eine Suche erzeugte Ergebnisliste sowie alle zugehörigen Daten der Suchtreffer sollen exportierbar sein
#FA_0206	Die Einträge der Ergebnisliste sollen in einer Extra Ansicht editierbar sein.
#FA_0207	Eingegebene Suchabfragen sollen speicherbar, aufrufbar und löscherbar sein.
#FA_0300	Benutzer sollen neue Datensätze anlegen und speichern können.
#FA_0301	Benutzer sollen bereits vorhandene Datensätze ändern können.
#FA_0302	Benutzer sollen zu bereits angelegten Datensätzen zusätzlich eine beliebige Anzahl von Nachuntersuchungen hinzufügen können.
#FA_0303	Angelegte Nachuntersuchungen sollen vom Benutzer nachträglich veränderbar sein
#FA_0304	Benutzer sollen vorhandene Datensätze mit den dazugehörigen Nachuntersuchungen löschen können.
#FA_0305	Benutzer sollen einzelne Nachuntersuchungen löschen können.
#FA_0306	Gelöschte Datensätze und Nachuntersuchungen sollen archiviert werden und nichtmehr im aktuellen Datenbestand sichtbar sein.
#FA_0307	Gelöschte Datensätze und Nachuntersuchungen sollen vom Benutzer bei Bedarf wiederherstellbar sein.
#FA_0400	Beim Anlegen eines neuen Datensatzes oder Verändern eines vorhandenen Datensatzes sollen alle Untersuchungsfragen aus dem Fragekatalog zusammen mit dem zugehörigem Eingabefeld in einer festgelegten Reihenfolge angezeigt werden.
#FA_0401	Die Untersuchungsfragen sollen in definierte Gruppen eingeteilt werden.
#FA_0402	Mit einem seitlichen Navigationsmenü soll zu den einzelnen Gruppen gesprungen werden können.
#FA_0403	Zu den einzelnen Fragen soll ein zugehöriger Tooltip angezeigt werden.
#FA_0404	Die Untersuchungsfragen im Fragekatalog sollen einen bestimmten Daten- / Eingabetyp zugeordnet bekommen.
#FA_0405	Die Untersuchungsfragen im Fragekatalog sollen abhängig von ihrem Daten- / Eingabetyp ein bestimmtes Eingabefeld mit Eingabebeschränkungen zugewiesen bekommen.
#FA_0406	Bei Untersuchungsfragen vom Typ Datum soll nur die Eingabe im Format DD.MM.YYYY möglich sein.
#FA_0407	Bei Untersuchungsfragen vom Typ Uhrzeit soll nur die Eingabe im Format HH:MM möglich sein.

#FA_0408	Bei Untersuchungsfragen vom Typ Bool soll nur die Auswahl "ja" / "nein" möglich sein.
#FA_0409	Bei Untersuchungsfragen mit numerischen Werten soll nur die Eingabe von Zahlen möglich sein.
#FA_0410	Bei Untersuchungsfragen mit vordefinierten Antworten soll nur eine Antwort über ein DropDown Menü auswählbar sein.
#FA_0411	Bei Untersuchungsfragen mit vordefinierten Antworten und Mehrfachauswahl sollen diese über Checkboxes auswählbar sein.
#FA_0412	Die Untersuchungsfrage "GSR ID" soll ein Eingabeformat für die korrekte Eingabe der GSR ID enthalten.
#FA_0413	Die Untersuchungsfrage "Age" soll Alterseingaben zwischen 0 und 150 zulassen.
#FA_0414	Die Untersuchungsfrage "Admission year" soll Jahreseingaben zwischen 1900 und 2199 zulassen.
#FA_0415	Die Untersuchungsfrage "Admission month" soll Monateingaben zwischen 1 und 12 zulassen.
#FA_0416	Die Untersuchungsfrage "Blood pressure on admission systolic mmHg" soll Blutdruckeingaben zwischen 10 und 400 zulassen.
#FA_0417	Die Untersuchungsfrage "Blood pressure on admission diastolic mmHg" soll Blutdruckeingaben zwischen 10 und 400 zulassen.
#FA_0418	Die Untersuchungsfrage "Heart rate on admission per min" soll Herzrateneingaben zwischen 10 und 500 zulassen.
#FA_0419	Die Untersuchungsfrage "NIHSS on admission[0-42]" soll Eingaben zwischen 0 und 42 zulassen.
#FA_0500	Vom Benutzer geänderte Einträge sollen mit fortlaufender Versionsnummer abgespeichert werden.
#FA_0501	Alle Änderungen eines Datensatzes sollen in einer Gesamtübersicht aufrufbar sein.
#FA_0502	Bei Einträgen mit mehreren Versionen soll ein Pop Up Fenster mit allen älteren Einträgen + Versionsnummer aufrufbar sein.
#FA_0503	Die Menge an ausgefüllten Feldern eines Datensatzes soll farblich und numerisch als Fortschritt angezeigt werden.
#FA_0504	Ein leeres Eingabefeld soll als „bearbeitet“ markiert werden können, falls es aufgrund von fehlenden Informationen nicht ausgefüllt werden kann.
#SA_0002	Daten aus der Datenbank sollen als Excel- oder CSV-Datei exportierbar sein.
#FA_0600	Die aktuelle Instanz der Datenbank soll mit den Einträgen in der jeweils höchsten Versionsnummer exportiert werden.
#FA_0601	Einzelne Untersuchungsfragen mitsamt allen Einträgen in der höchsten Version sollen exportierbar sein.
#FA_0602	Die zu exportierenden Untersuchungsfragen und deren Reihenfolge sollen vom Benutzer auswählbar sein.
#FA_0603	Die ausgewählte Kombination der Untersuchungsfragen soll speicherbar, aufrufbar und löschar sein.
#SA_0003	Eine GSR Import Datei im Excel Format soll in die Datenbank importiert werden können.
#FA_0700	Eine Importdatei, welche keine GSR-Importdatei im Excel Format darstellt, soll eine entsprechende Fehlermeldung erzeugen.

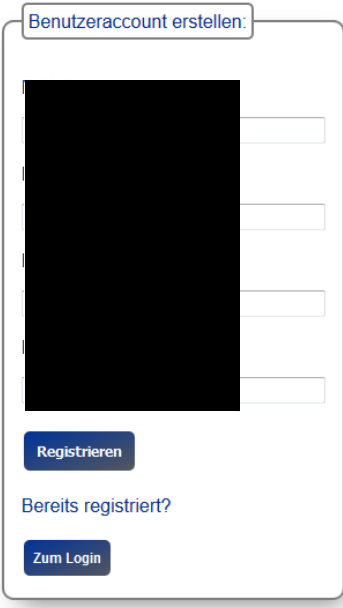
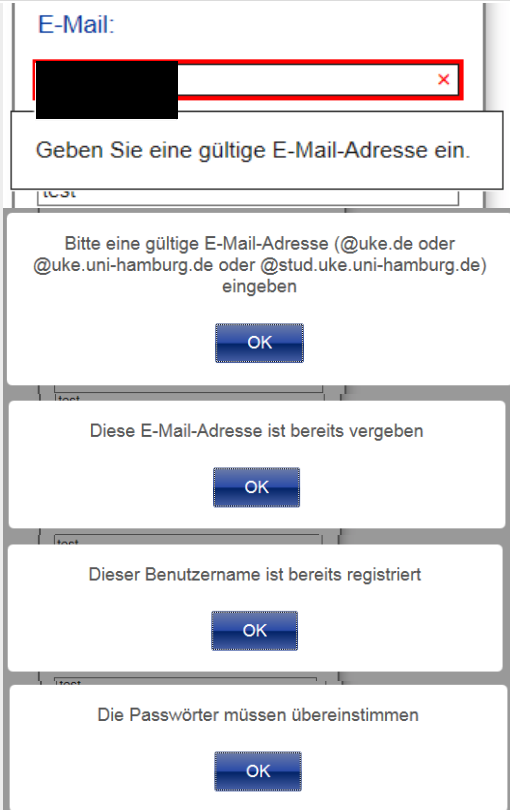
#FA_0701	Benutzer sollen auswählen können welche Importeinträge sie importieren wollen.
#FA_0702	Bekannte Fehler von Importeinträgen sollen automatisch vorkorrigiert werden.
#FA_0703	Benutzer sollen fehlerhafte Importeinträge manuell korrigieren können.
#FA_0704	Mit einem seitlichen Navigationsmenü soll direkt zu einzelnen Importfehlern gesprungen werden können.
#LA_0001	Der Upload der GSR-Importdatei soll nicht länger als 2 Minuten dauern.
#FA_0800	Benutzer mit Admin Status sollen neue Untersuchungsfragen zum Fragekatalog hinzufügen können.
#FA_0801	Neue Fragen sollen an einer vom Benutzer ausgewählten Stelle im Fragekatalog platziert werden können.
#FA_0802	Benutzer mit Admin Status sollen bereits vorhandene Fragen bearbeiten können.
#FA_0803	Die zu bearbeitende Frage soll vom Benutzer aus dem Fragekatalog auswählbar sein.
#FA_0804	Bei der Neueingabe / Bearbeitung soll der Frage Name, Daten- / Eingabetyp und Tooltip bestimmbar sein.
#FA_0805	Benutzer soll für den Eingabetyp <i>dropdown</i> eine beliebige Anzahl von vordefinierte Antworten festlegen können.
#FA_0806	Benutzer soll für den Eingabetyp <i>checkbox</i> eine beliebige Anzahl von vordefinierte Antworten festlegen können.
#FA_0807	Die vordefinierten Antworten sollen auch löschtbar sein.
#FA_0808	Ein leeres Eingabefeld zum Frage Namen soll eine entsprechende Fehlermeldung vor dem Abspeichern erzeugen.
#FA_0809	Falls der eingegebene Frage Name schon im Fragenkatalog existiert, soll eine entsprechende Fehlermeldung angezeigt werden.
#FA_0810	Ein leeres Eingabefeld zu einer vordefinierten Antwort soll eine entsprechende Fehlermeldung vor dem Abspeichern erzeugen.
#FA_0811	Benutzer mit Admin Status sollen 1 oder mehrere bereits vorhandene Frage(n) auswählen und löschen können.
#FA_0812	Benutzer mit Admin Status sollen die in der Weboberfläche benutzte Anzeige-Reihenfolge der Fragen verändern können.
#FA_0813	Benutzer mit Admin Status sollen eine iStroke Excel-Liste in die Datenbank importieren können.

**Tab. 11:** Anforderungsliste iStrokeDB

Alle Testfälle des Testplans aus 3.6 werden folgend aufgeführt:

Test ID #T_0101	
Prozedur / Eingabe	Registrieren eines neuen Benutzeraccounts
Eingabedaten	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>Email: [REDACTED]            Benutzername: [REDACTED]            Passwort: [REDACTED]            Passwort 2: [REDACTED]</p> </div> <div style="width: 35%; border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; border: 1px solid #0070C0; border-radius: 3px; margin-bottom: 5px;">Benutzeraccount erstellen:</p> <p>E-Mail:</p> <div style="background-color: black; height: 100px; width: 100%;"></div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <input style="width: 80%; height: 20px; border: 1px solid #ccc;" type="text"/> <input style="width: 15%; height: 20px; border: 1px solid #ccc;" type="text"/> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <input style="width: 60%; height: 20px; background-color: #0070C0; color: white; border: none;" type="button" value="Registrieren"/> </div> <p style="text-align: center; margin-top: 5px; color: #0070C0;">Bereits registriert?</p> <div style="display: flex; justify-content: center; margin-top: 5px;"> <input style="width: 60%; height: 20px; background-color: #0070C0; color: white; border: none;" type="button" value="Zurück zum Login"/> </div> </div> </div>
Erwarteter Zustand / Ausgabe	Erfolgreiche Erstellung des Benutzeraccounts mit den eingegeben Daten.
Test Ergebnis	<div style="display: flex; justify-content: space-between;"> <div style="width: 40%;">bestanden</div> <div style="width: 55%; border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin-bottom: 10px;">Benutzeraccount erfolgreich erstellt</p> <div style="display: flex; justify-content: center; margin-top: 5px;"> <input style="width: 60%; height: 20px; background-color: #0070C0; color: white; border: none;" type="button" value="Zum Login"/> </div> </div> </div>
Kommentar	

**Tab. 12:** Testfall - Registrieren eines neuen Benutzeraccounts

Test ID #T_0102	
Prozedur / Eingabe	Unzulässige Eingaben während des Registriervorgangs
Eingabedaten	<p>Email: [redacted] [redacted] [redacted]</p> 
Erwarteter Zustand / Ausgabe	Ausgabe der Fehlermeldungen für falsches Email Adressen Format, falsche Email Domain, bereits registrierte Email Adresse, bereits registrierter Benutzername, Passwort ist ungleich Passwort 2
Test Ergebnis	<p>bestanden</p> 
Kommentar	Leergelassene Eingabefelder sind ebenfalls unzulässige Eingaben und erzeugen entsprechende Fehlermeldungen, welche hier aus Platzgründen nicht aufgeführt sind.

Tab. 13: Testfall - Unzulässige Eingaben während des Registriervorgangs



Test ID #T_0103	
Prozedur / Eingabe	Login mit neu erstelltem Benutzeraccount
Eingabedaten	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>Benutzername: [redacted]</p> <p>[redacted]</p> </div> <div style="width: 35%; border: 1px solid gray; padding: 5px;"> <p style="text-align: center; border: 1px solid gray; border-radius: 3px;">Login</p> <p>Benutzername: [redacted] <input type="text"/></p> <p>[redacted] <input type="password"/></p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 50px; margin: 5px auto;">Login</p> <p>Noch keinen Benutzeraccount?</p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 50px; margin: 5px auto;">Registrieren</p> <p>Passwort vergessen?</p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 80px; margin: 5px auto;">Neues Passwort anfordern</p> </div> </div>
Erwarteter Zustand / Ausgabe	Erfolgreicher Login und Zugriff auf die Weboberfläche
Test Ergebnis	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;">bestanden</div> <div style="width: 35%; border: 1px solid gray; padding: 5px;"> <p>Sie sind angemeldet als: <b>test</b></p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 100px; margin: 5px auto;">User Panel Logout</p> </div> </div>
Kommentar	

**Tab. 14:** Testfall - Login mit neu erstelltem Benutzeraccount

Test ID #T_0104	
Prozedur / Eingabe	Login mit falschen & leeren Benutzerdaten
Eingabedaten	<div style="display: flex; justify-content: space-between;"> <div style="width: 50%;"> <p>Benutzername: [redacted]</p> <p>[redacted] [redacted]</p> <p>[redacted]</p> </div> <div style="width: 45%; border: 1px solid gray; padding: 5px;"> <p style="text-align: center; border: 1px solid gray; border-radius: 3px;">Login</p> <p>Benutzername: [redacted] <input type="text"/></p> <p>[redacted] <input type="password"/></p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 50px; margin: 5px auto;">Login</p> </div> </div>
Erwarteter Zustand / Ausgabe	Fehlermeldung über fehlerhafte Login Eingaben
Test Ergebnis	<div style="display: flex; justify-content: space-between;"> <div style="width: 50%;">bestanden</div> <div style="width: 45%; border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">Benutzername oder Passwort ungültig!</p> <p style="text-align: center; border: 1px solid gray; border-radius: 3px; width: 50px; margin: 5px auto;">OK</p> </div> </div>
Kommentar	

**Tab. 15:** Testfall - Login mit falschen & leeren Benutzerdaten

Test ID #T_0201																			
Prozedur / Eingabe	Anlegen eines neuen Datensatzes mit 3 Eingabewerten																		
Eingabedaten	<div style="display: flex;"> <div style="flex: 1;"> <p>Eingabefeld 1: testID</p> <p>Eingabefeld 2: testID2</p> <p>Eingabefeld 3: monday</p> </div> <div style="flex: 2;"> </div> </div>																		
Erwarteter Zustand / Ausgabe	Datensatz wird mit neuer Case ID angelegt und speichert die eingegebenen Werte																		
Test Ergebnis	<p>bestanden</p> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Case ID 600 wurde neu hinzugefügt</p> <p style="text-align: center;"><a href="#">Zur Startseite</a></p> </div> <div style="border: 1px solid gray; padding: 10px;"> <p>Last 10 records - Total: 600 record(s)</p> <table border="1"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> </tr> </thead> <tbody> <tr> <td>600</td> <td>3 / 108</td> <td>testID</td> <td>testID2</td> <td></td> <td></td> </tr> <tr> <td>599</td> <td>12 / 108</td> <td>uke 20161231 0251</td> <td></td> <td>566686802</td> <td></td> </tr> </tbody> </table> </div>	Case ID	Progress	ID	ID alt	UKE case	GSR ID	600	3 / 108	testID	testID2			599	12 / 108	uke 20161231 0251		566686802	
Case ID	Progress	ID	ID alt	UKE case	GSR ID														
600	3 / 108	testID	testID2																
599	12 / 108	uke 20161231 0251		566686802															
Kommentar																			

Tab. 16: Testfall - Anlegen eines neuen Datensatzes mit 3 Eingabewerten

Test ID #T_0202													
Prozedur / Eingabe	Bearbeiten des neu angelegten Datensatzes												
Eingabedaten	<div style="display: flex;"> <div style="flex: 1;"> <p>Eingabefeld 1: testID#neu</p> <p>Eingabefeld 2: testID#neu2</p> <p>Eingabefeld 3: friday</p> <p>Eingabefeld 4: 12:00</p> </div> <div style="flex: 2;"> </div> </div>												
Erwarteter Zustand / Ausgabe	Die veränderten und neu eingegebenen Werte werden gespeichert												
Test Ergebnis	<p>bestanden</p> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Case ID 600 wurde aktualisiert und gespeichert</p> <p style="text-align: center;"><a href="#">Zurück zur Case Ansicht</a></p> </div> <div style="border: 1px solid gray; padding: 10px;"> <p>Last 10 records - Total: 600 record(s)</p> <table border="1"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> </tr> </thead> <tbody> <tr> <td>600</td> <td>4 / 108</td> <td>testID#neu</td> <td>testID#neu2</td> <td></td> <td></td> </tr> </tbody> </table> </div>	Case ID	Progress	ID	ID alt	UKE case	GSR ID	600	4 / 108	testID#neu	testID#neu2		
Case ID	Progress	ID	ID alt	UKE case	GSR ID								
600	4 / 108	testID#neu	testID#neu2										
Kommentar													


Tab. 17: Testfall - Bearbeiten des neu angelegten Datensatzes

<b>Test ID #T_0203</b>																													
<b>Prozedur / Eingabe</b>	Monitoring der veränderten Eingabewerte																												
<b>Eingabedaten</b>																													
<b>Erwarteter Zustand / Ausgabe</b>	Die veränderten und neu eingegebenen Werte werden protokolliert																												
<b>Test Ergebnis</b>	<p>bestanden</p> <div style="border: 1px solid black; padding: 5px;"> <p>Monitoring: Alle Änderungen chronologisch sortiert</p> <table border="1"> <thead> <tr> <th>result date</th> <th>result time</th> <th>question</th> <th>old result</th> <th>new result</th> <th>version</th> <th>username</th> </tr> </thead> <tbody> <tr> <td>13.04.2017</td> <td>13:57</td> <td>Admission time hh:mm</td> <td></td> <td>12:00</td> <td>1</td> <td>test</td> </tr> <tr> <td>13.04.2017</td> <td>13:57</td> <td>Admission weekday</td> <td>monday</td> <td>friday</td> <td>2</td> <td>test</td> </tr> <tr> <td>13.04.2017</td> <td>13:55</td> <td>Admission weekday</td> <td></td> <td>monday</td> <td>1</td> <td>test</td> </tr> </tbody> </table> </div>	result date	result time	question	old result	new result	version	username	13.04.2017	13:57	Admission time hh:mm		12:00	1	test	13.04.2017	13:57	Admission weekday	monday	friday	2	test	13.04.2017	13:55	Admission weekday		monday	1	test
result date	result time	question	old result	new result	version	username																							
13.04.2017	13:57	Admission time hh:mm		12:00	1	test																							
13.04.2017	13:57	Admission weekday	monday	friday	2	test																							
13.04.2017	13:55	Admission weekday		monday	1	test																							
<b>Kommentar</b>	Die veränderten ID's werden nicht mitprotokolliert, da diese normalerweise eindeutig sind, d.h. nie geändert werden.																												

**Tab. 18:** Testfall - Monitoring der veränderten Eingabewerte

Test ID #T_0204		
Prozedur / Eingabe	Unzulässige Eingaben während der Datensatz Bearbeitung	
Eingabedaten		
Erwarteter Zustand / Ausgabe	Die fehlerhaften Eingaben werden optisch markiert und das Abspeichern verhindert	
Test Ergebnis	bestanden	
	GSR ID	GSF000000 x
	Soarian NRAD	Bitte im Format GSR***** eingeben
	Age	Sehr alt! x
		Altersangabe bis max. 150 möglich
	Basic data	
	Admission year	1742 x
	Admission month	Nur YYYY Eingabe möglich von 1900 bis 2199
	Symptom onset date	99.99.9999 x
	Time of recognition comments	Datumseingabe im Format DD.MM.YYYY
	Admission time hh:mm	12:00 Uhr! x
		Uhrzeiteingabe im Format HH:MM
	Admission time hh:mm	25:00 x
	Blood pressure on admission systolic mmHg	500
	Blood pressure on admission diastolic mmHg	3 x
Blood pressure on admission comments	Blutdruck min. 10 max. 400	
Heart rate on admission per min	6 x	
Heart rate on admission comments	Herzfrequenz min. 10 max. 500	
Kommentar		

Tab. 19: Testfall - Unzulässige Eingaben während der Datensatz Bearbeitung

Test ID #T_0205																							
Prozedur / Eingabe	Datensatz löschen & wiederherstellen																						
Eingabedaten																							
Erwarteter Zustand / Ausgabe	Datensatz wird gelöscht und ist nur noch im Archiv einsehbar. Nach Wiederherstellung ist er wieder Teil des aktiven Datenbestands.																						
Test Ergebnis	<p>bestanden</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Case ID 600 wurde gelöscht (archiviert)</p> <p style="text-align: center;"><a href="#">Zur Startseite</a></p> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Suchkriterien auswählen:</p> <p>Case ID <input type="text" value="="/> <input type="text" value="600"/> <a href="#">Suchen</a></p> <p><input type="checkbox"/> Weiteres Suchkriterium</p> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px; text-align: center;">  <p>Keine Suchergebnisse gefunden!</p> <p style="text-align: center;"><a href="#">Zurück zur Startseite</a></p> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;"> <a href="#">Import - GSR Data</a> <a href="#">Export</a> <a href="#">Archive</a> <a href="#">Last Changes</a> <a href="#">Statistics</a> </p> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Case ID zum Wiederherstellen auswählen:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Case ID</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><a href="#">600</a></td> <td>testID#neu</td> <td>testID2#neu2</td> <td></td> <td>GSF000000</td> </tr> </tbody> </table> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px; text-align: center;"> <p>Case 600 wurde erfolgreich wiederhergestellt</p> <p style="text-align: center;"><a href="#">Zurück zur Archiv Ansicht</a></p> </div> <div style="border: 1px solid gray; padding: 5px;"> <p>Last 10 records - Total: 600 record(s)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><a href="#">600</a></td> <td style="text-align: center;">5 / 108</td> <td>testID#neu</td> <td>testID</td> </tr> <tr> <td style="text-align: center;"><a href="#">599</a></td> <td style="text-align: center;">12 / 108</td> <td>uke_20161231_0251</td> <td></td> </tr> </tbody> </table> </div>	Case ID	ID	ID alt	UKE case	GSR ID	<a href="#">600</a>	testID#neu	testID2#neu2		GSF000000	Case ID	Progress	ID	ID	<a href="#">600</a>	5 / 108	testID#neu	testID	<a href="#">599</a>	12 / 108	uke_20161231_0251	
Case ID	ID	ID alt	UKE case	GSR ID																			
<a href="#">600</a>	testID#neu	testID2#neu2		GSF000000																			
Case ID	Progress	ID	ID																				
<a href="#">600</a>	5 / 108	testID#neu	testID																				
<a href="#">599</a>	12 / 108	uke_20161231_0251																					
Kommentar																							

**Tab. 20:** Testfall - Datensatz löschen & wiederherstellen

Test ID #T\_0301

Prozedur / Eingabe / Exportieren der aktuellen Datenbank Instanz

Eingabedaten

Export-Parameter:

Dateiname:

Markierung für leere & bearbeitete Items:

leere & bearbeitete Comments Felder ebenfalls markieren:

Follow-Up Daten exportieren:

Datei Format:  EXCEL (.xlsx)  CSV (.csv)

Aktuelle Instanz der DB herunterladen:

Reihenfolge der Spalten in der Export Datei auswählen:

iStroke Excel Liste (GSR Block einheitlich)  
 iStrokeDB-order-no

Erwarteter Zustand / Ausgabe / Erzeugung einer Excel Datei mit den ausgewählten Export Parametern.

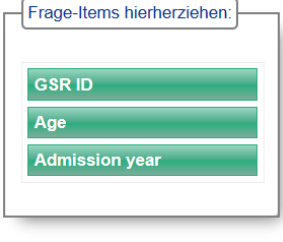
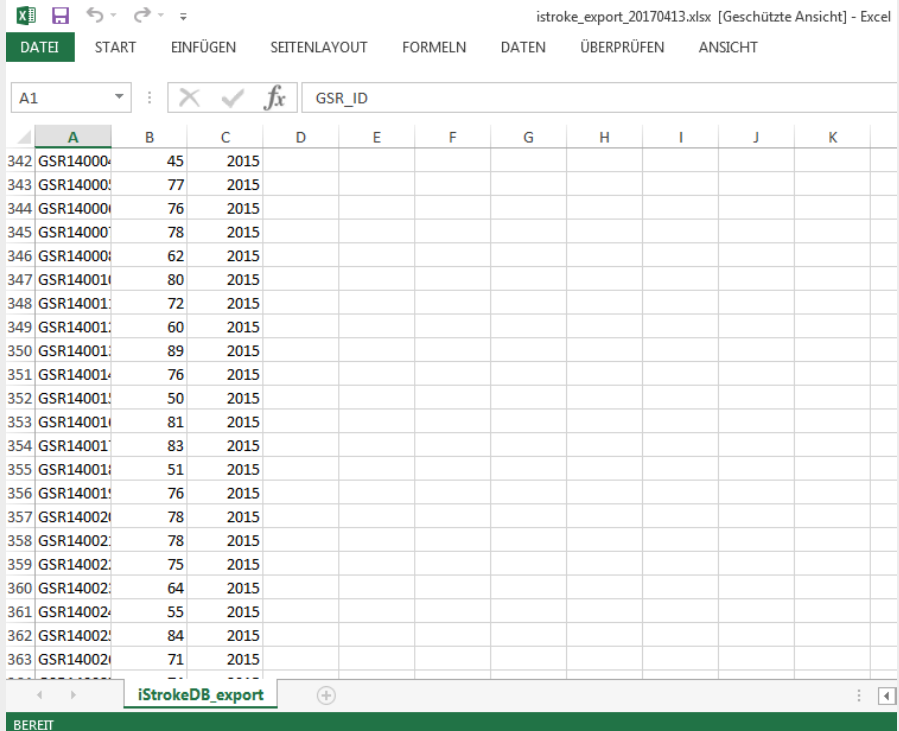
Test Ergebnis

bestanden

	A	B	C	D	E	F	G	H	I	J	K
580	579	uke_20161112_1336		5,66E+08		yes	yes				
581	580	uke_20161114_1503		5,66E+08		yes	yes				
582	581	uke_20161117_2144		5,66E+08		yes	yes				
583	582	uke_20161118_1232		5,66E+08		yes	no				Zeitverlauf NEURO fehlt
584	583	uke_20161121_1611		5,66E+08		yes	yes				
585	584	uke_20161121_1944		5,66E+08		yes	no				Zeitverlauf Schlaganfall fehlt kon
586	585	uke_20161125_0800		5,66E+08		yes	yes				
587	586	uke_20161126_1257		5,66E+08		yes	yes				
588	587	uke_20161129_0842		5,66E+08		yes	yes				FU CT noch ausstehend
589	588	uke_20161129_0941		5,66E+08		yes	yes				
590	589	uke_20161201_0114		5,66E+08		yes	yes				2 HEST Bögen mit geringen Abwe
591	590	uke_20161201_2230		5,66E+08		yes	yes				
592	591	uke_20161204_1944		5,66E+08		yes	yes				
593	592	uke_20161208_1909		5,67E+08		yes	yes				
594	593	uke_20161210_2211		5,67E+08		yes	yes				
595	594	uke_20161201_1118		5,66E+08		yes	no				
596	595	uke_20161215_1354		5,66E+08		yes	yes				
597	596	uke_20161226_0903		5,67E+08		yes	no				Zeitverlauf NEURO lückenhaft
598	597	uke_20161226_1428		5,67E+08		yes	no				Zeitverlauf NEURO lückenhaft
599	598	uke_20161230_1336		5,67E+08		yes	yes				
600	599	uke_20161231_0251		5,67E+08		yes	yes				FU noch ergänzen -CCT vom 31.12
601	600	testID#ne	testID2#neu2		#bearbeitet						

Kommentar

Tab. 21: Testfall - Exportieren der aktuellen Datenbank Instanz

Test ID #T_0302	
Prozedur / Eingabe	Individueller Export von Daten
Eingabedaten	
Erwarteter Zustand / Ausgabe	Erzeugung einer Excel Datei mit den ausgewählten Frage-Items und Export Parametern.
Test Ergebnis	<p>Bestanden</p> 
Kommentar	

Tab. 22: Testfall - Individueller Export von Daten

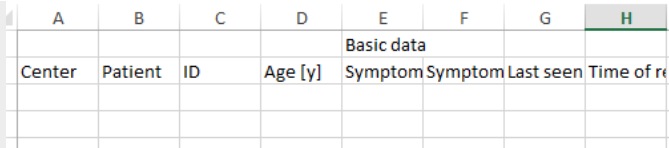

Test ID #T_0401																																																																		
Prozedur / Eingabe	Import von GSR Testdaten. Die Testdaten enthalten eingebaute Fehler, welche erkannt werden sollen.																																																																	
Eingabedaten	<p>2 GSR Datensätze im Excel Format</p> <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> <th>H</th> <th>I</th> <th>J</th> <th>K</th> <th>L</th> <th>M</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td colspan="9">Basic data</td> </tr> <tr> <td>Center</td> <td>Patient ID</td> <td>Age [y]</td> <td>Symptom</td> <td>Symptom</td> <td>Last seen</td> <td>Time of r</td> <td>Admissio</td> <td>Admissio</td> <td>Admissio</td> <td>Admissio</td> <td>Medicatio</td> <td></td> </tr> <tr> <td>14 GSR15000</td> <td>1</td> <td>85</td> <td>yes</td> <td>19:00</td> <td>19:00</td> <td>19:00</td> <td>2015</td> <td>6</td> <td>monday</td> <td>22:31</td> <td>ASS2</td> <td></td> </tr> <tr> <td>14 GSR14044</td> <td>2</td> <td>78</td> <td>yes</td> <td>10:30</td> <td>10:20</td> <td>10:30</td> <td>2015</td> <td>6</td> <td>wednesd</td> <td>10:59</td> <td>noneee</td> <td></td> </tr> </tbody> </table>	A	B	C	D	E	F	G	H	I	J	K	L	M					Basic data									Center	Patient ID	Age [y]	Symptom	Symptom	Last seen	Time of r	Admissio	Admissio	Admissio	Admissio	Medicatio		14 GSR15000	1	85	yes	19:00	19:00	19:00	2015	6	monday	22:31	ASS2		14 GSR14044	2	78	yes	10:30	10:20	10:30	2015	6	wednesd	10:59	noneee	
A	B	C	D	E	F	G	H	I	J	K	L	M																																																						
				Basic data																																																														
Center	Patient ID	Age [y]	Symptom	Symptom	Last seen	Time of r	Admissio	Admissio	Admissio	Admissio	Medicatio																																																							
14 GSR15000	1	85	yes	19:00	19:00	19:00	2015	6	monday	22:31	ASS2																																																							
14 GSR14044	2	78	yes	10:30	10:20	10:30	2015	6	wednesd	10:59	noneee																																																							
Erwarteter Zustand / Ausgabe	Formatfehler werden erkannt und optisch rot hervorgehoben. Vom Benutzer ausgewählte Daten werden importiert.																																																																	
Test Ergebnis	<p>bestanden</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>GSR150000</b></p> <table border="1"> <thead> <tr> <th>GSR Item</th> <th>DB value</th> <th>IMPORT value</th> <th>✓</th> </tr> </thead> <tbody> <tr> <td>GSR ID</td> <td></td> <td>NEW: GSR150000</td> <td>✓</td> </tr> <tr> <td>Age</td> <td></td> <td>85</td> <td>✓</td> </tr> <tr> <td>Symptom_onset_known</td> <td></td> <td>yes</td> <td>✓</td> </tr> <tr> <td>Symptom_onset_time_hh:mm</td> <td></td> <td>19.00</td> <td>✓</td> </tr> </tbody> </table>   <table border="1"> <tr> <td>Medication_before_admission</td> <td>ASS2</td> <td> <input checked="" type="checkbox"/> ASS  <input type="checkbox"/> Clopidogrel  <input type="checkbox"/> Aggrenox  <input type="checkbox"/> Marcumar  <input type="checkbox"/> Eliquis® (Apixaban)  <input type="checkbox"/> Pradaxa® (Dabigatran)  <input type="checkbox"/> Xarelto® (Rivaroxaban)  <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation)  <input type="checkbox"/> none  <input type="checkbox"/> Lixiana® (Edoxaban)         </td> </tr> </table>   <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>GSR Import ausgeführt:</p> <p>0 GSR record(s) aktualisiert            4 GSR record(s) neu eingetragen</p> <p>Im Archiv wurden kein Case aktualisiert</p> <p><a href="#">Zurück zur Startseite</a></p> </div>   <div style="display: flex;"> <div style="width: 20%; border: 1px solid gray; padding: 5px;"> <p>Case ID - 603</p> <p>Zurück</p> <p>Case löschen</p> <p>Case speichern</p> <p>Monitoring: Alle Änderungen anzeigen</p> <p>4 / 108</p> <p>General Information</p> <p>Basic data</p> <p>Imaging</p> <p>Acute stroke treatment</p> <p>24h follow-up</p> <p>Discharge</p> <p>90 days</p> <p>Additional Information</p> <p>Image Processing</p> <p>Follow-Up</p> <p>Neurofunktion</p> </div> <div style="width: 80%; border: 1px solid gray; padding: 5px;"> <table border="1"> <tr> <td>GSR ID</td> <td>GSR150000</td> <td>✓</td> </tr> <tr> <td>Soarian NRAD</td> <td></td> <td>□</td> </tr> <tr> <td>Soarian NL</td> <td></td> <td>□</td> </tr> <tr> <td>Soarian comments</td> <td></td> <td>✓</td> </tr> <tr> <td>General comments</td> <td></td> <td>✓</td> </tr> <tr> <td>Missing value comments</td> <td></td> <td>✓</td> </tr> <tr> <td>Workflow comments</td> <td></td> <td>✓</td> </tr> <tr> <td>Sex</td> <td></td> <td>□</td> </tr> <tr> <td>Age</td> <td>85</td> <td>✓</td> </tr> </table>   <table border="1"> <tr> <td>Medication before admission</td> <td> <input checked="" type="checkbox"/> ASS  <input type="checkbox"/> Clopidogrel  <input type="checkbox"/> Aggrenox  <input type="checkbox"/> Marcumar  <input type="checkbox"/> Eliquis® (Apixaban)  <input type="checkbox"/> Pradaxa® (Dabigatran)  <input type="checkbox"/> Xarelto® (Rivaroxaban)  <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation)  <input type="checkbox"/> none  <input type="checkbox"/> Lixiana® (Edoxaban)         </td> </tr> </table> </div> </div> </div>	GSR Item	DB value	IMPORT value	✓	GSR ID		NEW: GSR150000	✓	Age		85	✓	Symptom_onset_known		yes	✓	Symptom_onset_time_hh:mm		19.00	✓	Medication_before_admission	ASS2	<input checked="" type="checkbox"/> ASS <input type="checkbox"/> Clopidogrel <input type="checkbox"/> Aggrenox <input type="checkbox"/> Marcumar <input type="checkbox"/> Eliquis® (Apixaban) <input type="checkbox"/> Pradaxa® (Dabigatran) <input type="checkbox"/> Xarelto® (Rivaroxaban) <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation) <input type="checkbox"/> none <input type="checkbox"/> Lixiana® (Edoxaban)	GSR ID	GSR150000	✓	Soarian NRAD		□	Soarian NL		□	Soarian comments		✓	General comments		✓	Missing value comments		✓	Workflow comments		✓	Sex		□	Age	85	✓	Medication before admission	<input checked="" type="checkbox"/> ASS <input type="checkbox"/> Clopidogrel <input type="checkbox"/> Aggrenox <input type="checkbox"/> Marcumar <input type="checkbox"/> Eliquis® (Apixaban) <input type="checkbox"/> Pradaxa® (Dabigatran) <input type="checkbox"/> Xarelto® (Rivaroxaban) <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation) <input type="checkbox"/> none <input type="checkbox"/> Lixiana® (Edoxaban)													
GSR Item	DB value	IMPORT value	✓																																																															
GSR ID		NEW: GSR150000	✓																																																															
Age		85	✓																																																															
Symptom_onset_known		yes	✓																																																															
Symptom_onset_time_hh:mm		19.00	✓																																																															
Medication_before_admission	ASS2	<input checked="" type="checkbox"/> ASS <input type="checkbox"/> Clopidogrel <input type="checkbox"/> Aggrenox <input type="checkbox"/> Marcumar <input type="checkbox"/> Eliquis® (Apixaban) <input type="checkbox"/> Pradaxa® (Dabigatran) <input type="checkbox"/> Xarelto® (Rivaroxaban) <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation) <input type="checkbox"/> none <input type="checkbox"/> Lixiana® (Edoxaban)																																																																
GSR ID	GSR150000	✓																																																																
Soarian NRAD		□																																																																
Soarian NL		□																																																																
Soarian comments		✓																																																																
General comments		✓																																																																
Missing value comments		✓																																																																
Workflow comments		✓																																																																
Sex		□																																																																
Age	85	✓																																																																
Medication before admission	<input checked="" type="checkbox"/> ASS <input type="checkbox"/> Clopidogrel <input type="checkbox"/> Aggrenox <input type="checkbox"/> Marcumar <input type="checkbox"/> Eliquis® (Apixaban) <input type="checkbox"/> Pradaxa® (Dabigatran) <input type="checkbox"/> Xarelto® (Rivaroxaban) <input type="checkbox"/> LMWH, zB Clexane® (Enoxaprin) (zur Antikoagulation) <input type="checkbox"/> none <input type="checkbox"/> Lixiana® (Edoxaban)																																																																	
Kommentar																																																																		

Tab. 23: Testfall - Import von GSR Testdaten



Test ID #T_0402	
Prozedur / Eingabe	Import einer GSR Testdatei, welche nicht dem Import Format entspricht.
Eingabedaten	
Erwarteter Zustand / Ausgabe	Erkennung der Abweichung und Abbruch des Vorgangs.
Test Ergebnis	bestanden
Kommentar	

Tab. 24: Testfall - Import einer GSR Testdatei, welche nicht dem Import Format entspricht

Test ID #T_0403	
Prozedur / Eingabe	Import einer GSR Testdatei, welche dem Import Format entspricht, aber keine Import Daten enthält.
Eingabedaten	
Erwarteter Zustand / Ausgabe	Es werden keine Import Daten in der Übersicht angezeigt und somit auch nichts importiert.
Test Ergebnis	bestanden
Kommentar	

Tab. 25: Testfall - Import einer GSR Testdatei ohne Importdaten

Test ID #T_0501																																																																	
Prozedur / Eingabe	Suche mit festgelegten Suchparametern																																																																
Eingabedaten	<div style="border: 1px solid gray; padding: 5px;"> <p>Suchkriterien auswählen:</p> <p>Age &gt;= 70 <span style="float: right;">Suchen</span></p> <p><input checked="" type="checkbox"/> Weiteres Suchkriterium</p> <p><input type="radio"/> UND <input type="radio"/> ODER</p> <p>Admission_year &gt;= 2015</p> <p><input checked="" type="checkbox"/> Weiteres Suchkriterium</p> <p><input type="radio"/> UND <input type="radio"/> ODER</p> <p>Admission_year &lt;= 2017</p> <p><input type="checkbox"/> Weiteres Suchkriterium</p> </div>																																																																
Erwarteter Zustand / Ausgabe	Es werden alle Datensätze angezeigt, bei denen das Aufnahmejahr zwischen 2015 und 2017 liegt und die Patienten älter als 70 sind.																																																																
Test Ergebnis	<p>bestanden</p> <p>115 result(s) - Case ID zum Bearbeiten / Löschen auswählen:</p> <table border="1"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> <th>Age</th> <th>Admission_year</th> </tr> </thead> <tbody> <tr> <td>326</td> <td>76 / 110</td> <td>uke_20150615_2251</td> <td></td> <td>554774014</td> <td>GSR140001</td> <td>85</td> <td>2015</td> </tr> <tr> <td>327</td> <td>76 / 110</td> <td>uke_20150617_1119</td> <td></td> <td>554789491</td> <td>GSR140002</td> <td>78</td> <td>2015</td> </tr> <tr> <td>330</td> <td>85 / 110</td> <td>uke_20150626_1737</td> <td></td> <td>554878237</td> <td>GSR140005</td> <td>77</td> <td>2015</td> </tr> <tr> <td>331</td> <td>67 / 110</td> <td>uke_20150705_1925</td> <td></td> <td>554958526</td> <td>GSR140006</td> <td>76</td> <td>2015</td> </tr> <tr> <td>332</td> <td>68 / 110</td> <td>uke_20150707_0135</td> <td></td> <td>554976338</td> <td>GSR140007</td> <td>78</td> <td>2015</td> </tr> <tr> <td>335</td> <td>75 / 110</td> <td>uke_20150722_1252</td> <td></td> <td>555141254</td> <td>GSR140010</td> <td>80</td> <td>2015</td> </tr> <tr> <td>336</td> <td>69 / 110</td> <td>uke_20150729_1534</td> <td></td> <td>555203369</td> <td>GSR140011</td> <td>72</td> <td>2015</td> </tr> </tbody> </table>	Case ID	Progress	ID	ID alt	UKE case	GSR ID	Age	Admission_year	326	76 / 110	uke_20150615_2251		554774014	GSR140001	85	2015	327	76 / 110	uke_20150617_1119		554789491	GSR140002	78	2015	330	85 / 110	uke_20150626_1737		554878237	GSR140005	77	2015	331	67 / 110	uke_20150705_1925		554958526	GSR140006	76	2015	332	68 / 110	uke_20150707_0135		554976338	GSR140007	78	2015	335	75 / 110	uke_20150722_1252		555141254	GSR140010	80	2015	336	69 / 110	uke_20150729_1534		555203369	GSR140011	72	2015
Case ID	Progress	ID	ID alt	UKE case	GSR ID	Age	Admission_year																																																										
326	76 / 110	uke_20150615_2251		554774014	GSR140001	85	2015																																																										
327	76 / 110	uke_20150617_1119		554789491	GSR140002	78	2015																																																										
330	85 / 110	uke_20150626_1737		554878237	GSR140005	77	2015																																																										
331	67 / 110	uke_20150705_1925		554958526	GSR140006	76	2015																																																										
332	68 / 110	uke_20150707_0135		554976338	GSR140007	78	2015																																																										
335	75 / 110	uke_20150722_1252		555141254	GSR140010	80	2015																																																										
336	69 / 110	uke_20150729_1534		555203369	GSR140011	72	2015																																																										
Kommentar																																																																	


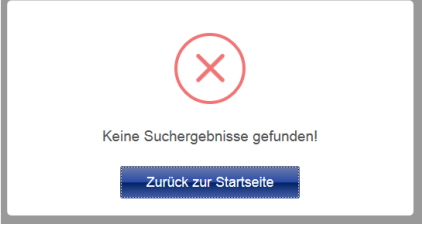
Tab. 26: Testfall - Suche mit festgelegten Suchparametern

Test ID #T_0502																									
Prozedur / Eingabe	Suche mit festgelegten Suchparametern #2																								
Eingabedaten	<div style="border: 1px solid gray; padding: 5px;"> <p>Suchkriterien auswählen:</p> <p>Age = 30 <span style="float: right; border: 1px solid gray; padding: 2px 5px;">Suchen</span></p> <p><input checked="" type="checkbox"/> Weiteres Suchkriterium</p> <p><input checked="" type="radio"/> UND <input type="radio"/> ODER</p> <p>Admission_weekday &lt;&gt; wednesday</p> <p><input type="checkbox"/> Weiteres Suchkriterium</p> </div>																								
Erwarteter Zustand / Ausgabe	Es werden alle Datensätze angezeigt, bei denen der Patient 30 Jahre alt ist und nicht an einem Mittwoch aufgenommen wurde.																								
Test Ergebnis	<p>bestanden</p> <div style="border: 1px solid gray; padding: 5px;"> <p>2 result(s) - Case ID zum Bearbeiten / Löschen auswählen:</p> <table border="1"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> <th>Age</th> <th>Admission_weekday</th> </tr> </thead> <tbody> <tr> <td>372</td> <td>71 / 110</td> <td>uke_20151126_0857</td> <td></td> <td>556296602</td> <td>GSR140044</td> <td>30</td> <td>saturday</td> </tr> <tr> <td>493</td> <td>64 / 110</td> <td>uke_20160920_1921</td> <td></td> <td>565738075</td> <td>GSR140166</td> <td>30</td> <td>tuesday</td> </tr> </tbody> </table> </div>	Case ID	Progress	ID	ID alt	UKE case	GSR ID	Age	Admission_weekday	372	71 / 110	uke_20151126_0857		556296602	GSR140044	30	saturday	493	64 / 110	uke_20160920_1921		565738075	GSR140166	30	tuesday
Case ID	Progress	ID	ID alt	UKE case	GSR ID	Age	Admission_weekday																		
372	71 / 110	uke_20151126_0857		556296602	GSR140044	30	saturday																		
493	64 / 110	uke_20160920_1921		565738075	GSR140166	30	tuesday																		
Kommentar																									


**Tab. 27:** Testfall - Suche mit festgelegten Suchparametern #2

Test ID #T_0503	
Prozedur / Eingabe	Exportieren aller Untersuchungsdaten zu den in #T_0100 gefundenen Datensätzen.
Eingabedaten	
Erwarteter Zustand / Ausgabe	Erzeugung einer Export Datei mit allen Untersuchungsdaten zu den Suchtreffern.
Test Ergebnis	<p>bestanden</p>  
Kommentar	

**Tab. 28:** Testfall - Exportieren aller Untersuchungsdaten zu den gefundenen Datensätzen

Test ID #T_0504	
Prozedur / Eingabe	Suche mit fehlerhaften Suchparametern.
Eingabedaten	
Erwarteter Zustand / Ausgabe	Meldung , dass keine Suchergebnisse gefunden wurden.
Test Ergebnis	bestanden 
Kommentar	

Tab. 29: Testfall - Suche mit fehlerhaften Suchparametern

Test ID #T_0505	
Prozedur / Eingabe	Suche mit unvollständigen Suchparametern.
Eingabedaten	
Erwarteter Zustand / Ausgabe	Wenn die Suchkriterien-Felder leer sind, wird standardmäßig nach Case ID gesucht.
Test Ergebnis	bestanden 
Kommentar	

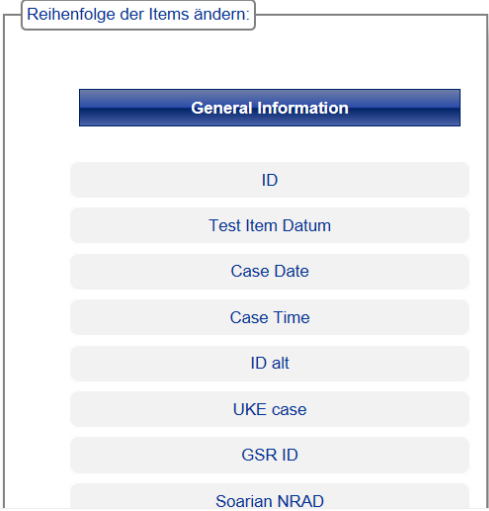
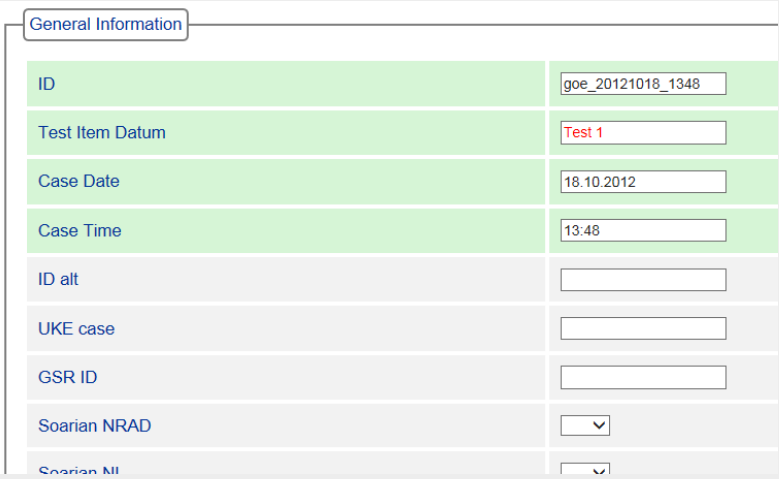
Tab. 30: Testfall - Suche mit unvollständigen Suchparametern

Test ID #T_0601																																																																																					
Prozedur / Eingabe	Hinzufügen eines neuen Frage-Items																																																																																				
Eingabedaten																																																																																					
Erwarteter Zustand / Ausgabe	Neues Frage-Item erscheint an der ausgewählten Stelle mit den gewählten Parametern. Zudem kann in der Suchfunktion nach dem neuen Item gesucht werden und es wird beim Export mitexportiert.																																																																																				
Test Ergebnis	<p>bestanden</p> <table border="1" data-bbox="475 1512 1149 1668"> <thead> <tr> <th>Case ID</th> <th>Progress</th> <th>ID</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> <th>Test_Item</th> </tr> </thead> <tbody> <tr> <td>658</td> <td>17 / 111</td> <td>goe_20121018_1348</td> <td></td> <td></td> <td></td> <td>Test 1</td> </tr> </tbody> </table> <table border="1" data-bbox="446 1713 1173 1899"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> <th>H</th> <th>I</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Case ID</td> <td>ID</td> <td>Case Date</td> <td>Case Time</td> <td>ID alt</td> <td>UKE case</td> <td>GSR ID</td> <td>Test Item</td> <td>Soarian</td> </tr> <tr> <td>2</td> <td></td> <td>658</td> <td>goe_2012:</td> <td>18.10.2012</td> <td>13:48</td> <td></td> <td></td> <td>Test 1</td> <td></td> </tr> <tr> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>5</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>6</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Case ID	Progress	ID	ID alt	UKE case	GSR ID	Test_Item	658	17 / 111	goe_20121018_1348				Test 1		A	B	C	D	E	F	G	H	I	1	Case ID	ID	Case Date	Case Time	ID alt	UKE case	GSR ID	Test Item	Soarian	2		658	goe_2012:	18.10.2012	13:48			Test 1		3										4										5										6									
Case ID	Progress	ID	ID alt	UKE case	GSR ID	Test_Item																																																																															
658	17 / 111	goe_20121018_1348				Test 1																																																																															
	A	B	C	D	E	F	G	H	I																																																																												
1	Case ID	ID	Case Date	Case Time	ID alt	UKE case	GSR ID	Test Item	Soarian																																																																												
2		658	goe_2012:	18.10.2012	13:48			Test 1																																																																													
3																																																																																					
4																																																																																					
5																																																																																					
6																																																																																					
Kommentar	Für das Hinzufügen des Frage-Items wird ein Benutzeraccount mit Admin Rechten benötigt.																																																																																				

Tab. 31: Testfall - Hinzufügen eines neuen Frage-Items

Test ID #T_0602																	
Prozedur / Eingabe	Bearbeiten des Frage-Items aus #T_0601																
Eingabedaten																	
Erwarteter Zustand / Ausgabe	Veränderte Parameter werden übernommen.																
Test Ergebnis	<p>bestanden</p> <table border="1"> <tr> <td>ID</td> <td>goe_20121018_1348</td> </tr> <tr> <td>Case Date</td> <td>18.10.2012</td> </tr> <tr> <td>Case Time</td> <td>13:48</td> </tr> <tr> <td>ID alt</td> <td></td> </tr> <tr> <td>UKE case</td> <td></td> </tr> <tr> <td>GSR ID</td> <td></td> </tr> <tr> <td>Test Item Datum</td> <td>Test 1</td> </tr> <tr> <td>Soarian NRAD</td> <td>Datumseingabe im Format DD.MM.YYYY</td> </tr> </table> 	ID	goe_20121018_1348	Case Date	18.10.2012	Case Time	13:48	ID alt		UKE case		GSR ID		Test Item Datum	Test 1	Soarian NRAD	Datumseingabe im Format DD.MM.YYYY
ID	goe_20121018_1348																
Case Date	18.10.2012																
Case Time	13:48																
ID alt																	
UKE case																	
GSR ID																	
Test Item Datum	Test 1																
Soarian NRAD	Datumseingabe im Format DD.MM.YYYY																
Kommentar	<p>Wenn der Datentyp eines Frage-Items geändert wird, können Konflikte mit bereits abgespeicherten Daten zu diesem Item auftreten (siehe rote Markierung). Für das Bearbeiten des Frage-Items wird ein Benutzeraccount mit Admin Rechten benötigt.</p>																

**Tab. 32:** Testfall - Bearbeiten des Frage-Items aus #T\_0601

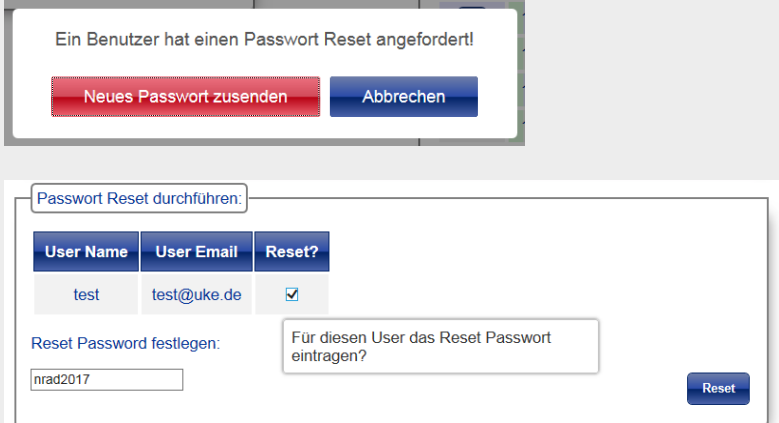
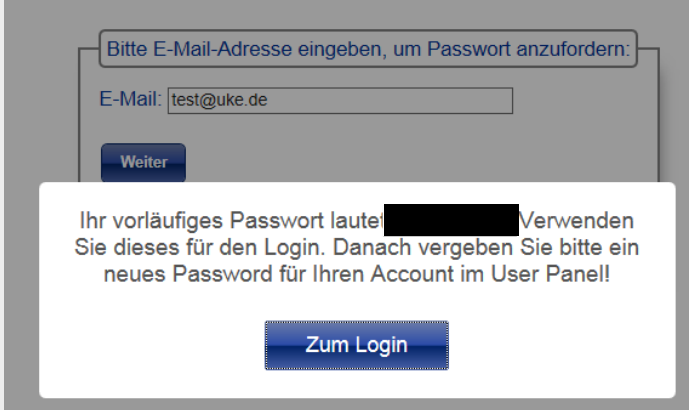
Test ID #T_0603																					
Prozedur / Eingabe	Verschieben der Position des Frage-Items aus #T_0601 im Fragenkatalog an die zweite Stelle.																				
Eingabedaten																					
Erwarteter Zustand / Ausgabe	Veränderte Position wird übernommen.																				
Test Ergebnis	<p>bestanden</p>  <table border="1" data-bbox="459 1574 1428 1765"> <thead> <tr> <th>Case ID</th> <th>ID</th> <th>Test Item</th> <th>Case Date</th> <th>Case Time</th> <th>ID alt</th> <th>UKE case</th> <th>GSR ID</th> <th>Soarian NI</th> <th>Soarian</th> </tr> </thead> <tbody> <tr> <td>658</td> <td>goe_2012</td> <td>Test 1</td> <td>18.10.2012</td> <td>13:48</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Case ID	ID	Test Item	Case Date	Case Time	ID alt	UKE case	GSR ID	Soarian NI	Soarian	658	goe_2012	Test 1	18.10.2012	13:48					
Case ID	ID	Test Item	Case Date	Case Time	ID alt	UKE case	GSR ID	Soarian NI	Soarian												
658	goe_2012	Test 1	18.10.2012	13:48																	
Kommentar	Für das Verschieben des Frage-Items wird ein Benutzeraccount mit Admin Rechten benötigt.																				

**Tab. 33:** Testfall- Verschieben der Position des Frage-Items aus #T\_0601 im Fragenkatalog



Test ID #T_0604																									
Prozedur / Eingabe	Löschen des Frage-Items aus #T_0601.																								
Eingabedaten																									
Erwarteter Zustand / Ausgabe	Frage-Item wird aus dem Fragenkatalog gelöscht. Zum Item assoziierte Einträge werden ebenfalls gelöscht.																								
Test Ergebnis	<p>Bestanden</p>  <table border="1" data-bbox="453 801 1353 1106"> <thead> <tr> <th colspan="3">General Information</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>goe_20121018_1348</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Case Date</td> <td>18.10.2012</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Case Time</td> <td>13:48</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>ID alt</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>UKE case</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>GSR ID</td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>Soarian NRAD</td> <td></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	General Information			ID	goe_20121018_1348	<input checked="" type="checkbox"/>	Case Date	18.10.2012	<input checked="" type="checkbox"/>	Case Time	13:48	<input checked="" type="checkbox"/>	ID alt		<input type="checkbox"/>	UKE case		<input type="checkbox"/>	GSR ID		<input type="checkbox"/>	Soarian NRAD		<input type="checkbox"/>
General Information																									
ID	goe_20121018_1348	<input checked="" type="checkbox"/>																							
Case Date	18.10.2012	<input checked="" type="checkbox"/>																							
Case Time	13:48	<input checked="" type="checkbox"/>																							
ID alt		<input type="checkbox"/>																							
UKE case		<input type="checkbox"/>																							
GSR ID		<input type="checkbox"/>																							
Soarian NRAD		<input type="checkbox"/>																							
Kommentar	Für das Löschen des Frage-Items wird ein Benutzeraccount mit Admin Rechten benötigt.																								

**Tab. 34:** Testfall - Löschen des Frage-Items aus #T\_0601

Test ID #T_0701							
Prozedur / Eingabe	Neuvergabe eines Passworts bei Passwort Reset Anfrage.						
Eingabedaten							
Erwarteter Zustand / Ausgabe	Das alte Passwort wird durch das Reset Passwort ersetzt. Zusätzlich wird das Reset Passwort an die eingetragene Email verschickt, wenn diese in der DB registriert ist.						
Test Ergebnis	<p><b>Bestanden</b></p>  <table border="1" data-bbox="480 981 778 1066"> <thead> <tr> <th>User Name</th> <th>User Email</th> <th>Reset?</th> </tr> </thead> <tbody> <tr> <td>test</td> <td>test@uke.de</td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table> <p>Reset Passwort festlegen: <input type="text" value="nrad2017"/> <input type="button" value="Reset"/></p> <p>Für diesen User das Reset Passwort eintragen?</p> <p>Für 1 Benutzer wurde das Reset Passwort eingetragen!</p> <p><input type="button" value="Zurück zu Admin Tools"/></p> 	User Name	User Email	Reset?	test	test@uke.de	<input checked="" type="checkbox"/>
User Name	User Email	Reset?					
test	test@uke.de	<input checked="" type="checkbox"/>					
Kommentar	Für das Vergeben eines Reset Passworts wird ein Benutzeraccount mit Admin Rechten benötigt. Die Versendung des Passworts per Email kann erst getestet werden, wenn das System auf einem Server im Intranet der NRAD installiert ist.						

Tab. 35: Testfall - Neuvergabe eines Passworts bei Passwort Reset Anfrage