HOCHSCHULE LANDSHUT
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

Bachelorarbeit zum Thema

ENTWICKLUNG EINES KONZEPTS FÜR EINE MODULARE APP FÜR MEDIZINTECHNISCHE ANWENDUNGEN – UMSETZUNG AM BEISPIEL RECIST

vorgelegt von

Michael Uhl
aus Ergolding

eingereicht am

27.02.2016

Betreuer: Prof. Dr. Stefanie Remmele

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

Erklärung zur Bachelorarbeit

(gemäß § 9 d, Abs. 3 APO)

Name, Vorname der/des

Student(in)en: .....

Hochschule Landshut
Fakultät Elektrotechnik und Wirtschaftsingenieurwesen

Hiermit erkläre ich, dass ich die Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

…… ……

(Datum) (Unterschrift der/des Student(in)en)

Fakultät für Elektrotechnik und Wirtschaftsingenieurwesen

FREIGABEERKLÄRUNG DER/DES STUDENT(IN)EN

Name, Vorname der/des Studentin/en:

.....

Hiermit erkläre ich, dass die vorliegende Bachelorarbeit in den Bestand der Hochschulbibliothek aufgenommen werden kann und

□ ohne Sperrfrist

oder nach einer Sperrfrist von

□ 1 Jahr
□ 2 Jahren
□ 3 Jahren
□ 5 Jahren oder länger

über die Hochschulbibliothek zugänglich gemacht werden darf.

.....
(Datum) (Unterschrift der/des Student(in)en)

Abkürzungsverzeichnis

CR	Complete Response
CSS	Cascading Style Sheets
DICOM	Digital Imaging and Communications in Medicine
HTML	Hypertext Markup Language
irRC	Immune-related Response Criteria
Nadir	arabisch: Fußpunkt
PACS	Picture Archiving and Communication System
PD	Progressive Disease
PHP	PHP: Hypertext Preprocessor
PR	Partial Response
RECIST	Response Evaluation Criteria in Solid Tumors
SD	Stable Disease
URL	Uniform Resource Locator

Inhaltsverzeichnis

1 Ziel

1.1 Entwicklung eines Konzepts zur Speicherung und Verarbeitung von textbasierten Daten

Ziel dieser Arbeit ist die Entwicklung eines Konzeptes für eine medizinische Software, deren modularer Aufbau eine einfache Wiederverwendung von Softwareteilen bei zukünftigen Softwareprojekten der Hochschule Landshut im Studiengang Biomedizinische Technik ermöglicht. Insbesondere wird in dieser Arbeit auf die Speicherung und Verarbeitung von Text- und Metadaten eingegangen.

1.2 Entwicklung einer Anwendung unter Verwendung des beschriebenen Softwarekonzepts

Im zweiten Teil der Arbeit wird eine Software zur Speicherung von Läsionsparametern und Bewertung des Verlaufs einer Tumorerkrankung entwickelt. RECIST-Kriterien ermöglichen die Bewertung des Tumoransprechens auf eine Behandlung durch die entwickelte Software. Das Programm soll Radiologen bei der Erstellung strukturierter Befundungen unter Verwendung wissenschaftlicher Erkenntnisse unterstützen.

2 Einleitung

2.1 Aufbau der Arbeit

Die Entwicklung des genannten Softwarekonzeptes wird aufgrund des großen Projektumfangs auf zwei Bachelorarbeiten aufgeteilt. Die vorliegende Arbeit setzt sich insbesondere mit der Speicherung und Verarbeitung von textbasierten Daten auseinander. Der Umgang mit medizinischen Bilddaten ist Gegenstand der Bachelorarbeit von Herrn Andreas Hintermaier [8].

2.2 Stand der Technik

Ärzte und Krankenhauspersonal sehen sich heute mit einer immer größer werdenden Anzahl an Softwareprodukten konfrontiert. Hierzu zählt u. A. Software zur Steuerung von medizinischen Geräten, Patienteninformationssysteme zur Verwaltung von Patientendaten und PACS-Systeme für die Archivierung und den Austausch von Bilddaten aus verschiedenen bildgebenden Verfahren. Durch Softwaretools können Mediziner komplexe biologische Parameter und Biomarker in Bilddaten sichtbar machen und vermessen. Diese modernen Methoden bieten das Potential zur Verbesserung der Diagnose und damit der erfolgreichen Behandlung von Krankheiten, stellen aber gleichzeitig neue Anforderungen an Mediziner und Personal.

Heute handelt es sich bei der verwendeten Software in der Regel um native Software, die auf einem entsprechenden Endgerät installiert wird. Ein großer Verwaltungsaufwand ist notwendig, um die Hard- und Softwarekomponenten bereitstellen zu können. Applikationen, die ausschließlich im Internet als Web-Applikation zur Verfügung gestellt werden, sind ein sich in der Entwicklung befindlicher Trend. Ärzte können von unterschiedlichen Arbeitsplätzen und Endgeräten auf diese Anwendungen zugreifen. Eine Installation auf dem Endgerät ist nicht notwendig. Daher ist es einfach, die Software nach der Veröffentlichung zu Warten, Softwarefehler zu beheben und neue Funktionen zu implementieren. Komplexe Anforderungen an den Datenaustausch zwischen Client und Server, sowie an die Verwahrung von Patientendaten, stellen Hersteller solcher Lösungen aber vor neue Herausforderungen.

Die Komplexität verwendeter Software reicht von sehr einfachen RECIST-Rechnern, die eine Auswertung von Daten lediglich zwischen zwei Messzeitpunkten erlauben, bis hin zu umfangreichen Systemen zur Befundung, die eine Erfassung notwendiger Messdaten direkt aus den medizinischen Bilddaten möglich machen und nach Anwendung diverser Kriterien bei der Erstellung von Berichten unterstützen.

Proprietäre Software, die umfangreich bei der Befundung unterstützen, sind u. A. Mint Lesion der Firma Mint Medical GmbH und syngo.via der Firma Siemens Healthcare GmbH [14].

Neben den genannten umfangreichen Softwarelösungen teils großer Medizinproduktehersteller gibt es einige kleine Apps direkt im Internet und für Smartphones. Sie zeichnen sich durch einen recht

geringen Funktionsumfang aus. Genannt seien an dieser Stelle der RECIST 1.1 Calculator von Louis Lassalle [13] und der Radiology Tutor Jonathan Colledge [12].

2.3 Motivation

Die Hochschule Landshut hat im Jahr 2012 den Studiengang Biomedizinische Technik eingerichtet. Das Studium bietet eine interdisziplinäre Ausbildung mit Schwerpunkten in der Elektrotechnik ergänzt um biologische und medizinische Inhalte. Die Studenten erlernen während des sieben Semester andauernden Bachelorstudiengangs zudem die Grundlagen der Entwicklung von Software in der Medizintechnik, darunter Informatik, verschiedene Programmiersprachen, Projektmanagement sowie Anforderungen an medizinische Produkte und Software.

Es ist daher naheliegend, ein grundlegendes Konzept für medizinische Software zu entwickeln, unter dessen Anwendung wissenschaftliche Hilfskräfte oder Studenten im Rahmen von Abschlussarbeiten Softwareprojekte verwirklichen können. Es soll eine Softwarelösung entstehen, die sowohl auf festen Arbeitsplätzen, als auch mobil über Smartphones und Tablets angewendet werden kann.

Die Softwareprojekte bieten außerdem Möglichkeiten für die Zusammenarbeit und den Informationsaustausch mit industriellen Partner und klinischen Institutionen. Die Hochschule kann auf diese Weise ihre Beziehungen zu externen Partnern aus Medizin und Medizintechnik stärken und zur Verbreitung innovativer Ideen beitragen.

Die Entwicklung einer RECIST-Applikation wurde maßgeblich durch Herrn Dr. Thorsten Persigehl von der Uniklinik Köln und von der Deutschen Röntgengesellschaft e.V. angestoßen. Die heute insbesondere in medizinischen Studien verwendete Sammlung von Kriterien ermöglicht eine standardisierte Bewertung des Outcomes von Tumorerkrankungen und stellt damit die Vergleichbarkeit von Studienergebnissen sicher. Ziel der Entwicklung ist die Bereitstellung eines Tools, welches einer großen Zahl von Ärzten die Vorteile der Befundungsmethode näherbringen kann. Existierende Softwarelösungen sind teuer in der Anschaffung und erfordern aufgrund ihrer hohen Komplexität lange Einarbeitungszeiten. Eine mobile Applikation mit einem auf das Wesentliche reduziertem Funktionsumfang, welche Ärzten kostenlos zur Verfügung gestellt werden kann, kann zur Verwendung moderner Befundungsmethoden motivieren.

3 Methode

3.1 Softwarearchitektur zur Speicherung und Verarbeitung von textbasierten Patientendaten

Die Entwicklung des hier beschriebenen Softwarekonzeptes wurde innerhalb von zwei Bachelorarbeiten durchgeführt. In der vorliegenden Arbeit wurde insbesondere die Speicherung und Verarbeitung von textbasierten Daten betrachtet. Eine weitere Arbeit von Herrn Andreas Hintermaier beleuchtet außerdem die Verarbeitung von Bilddaten [8].

3.1.1 Anforderungen

Abbildung 1 gibt zunächst eine kurze Übersicht über die grundlegenden Anforderungen an das System.

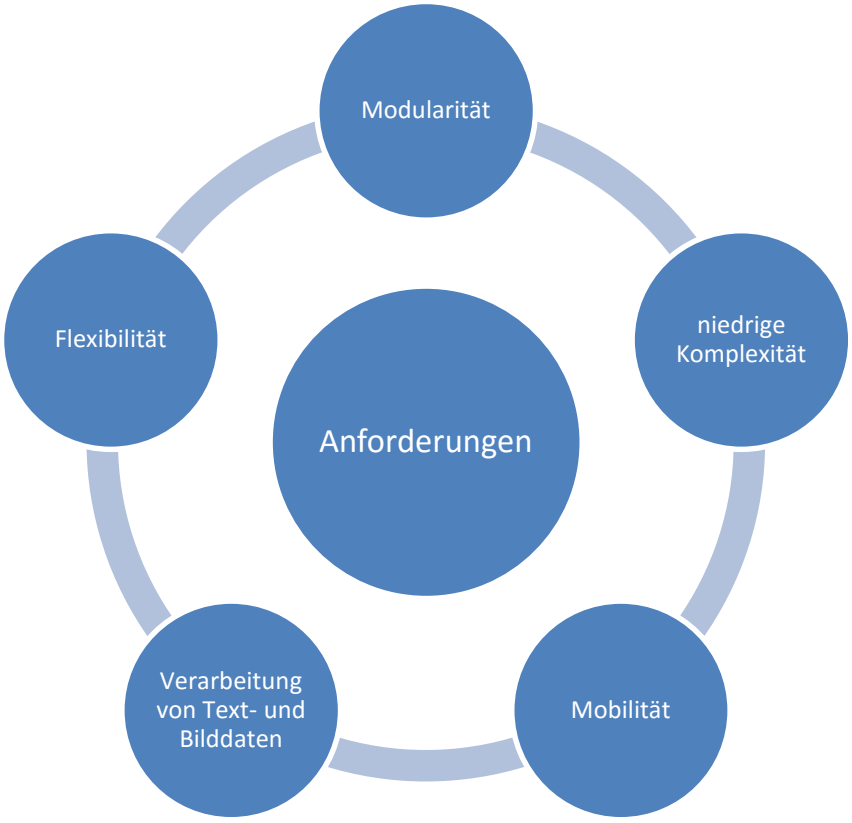

Abbildung 1: Grundlegende Anforderung an die zu entwickelnde Software

Die Anwendung soll auf möglichst vielen Endgeräten lauffähig und auch mobil nutzbar sein. Es soll also eine Verwendung auf einem Computer, als auch auf Smartphones und Tablets möglich sein. Die

Software soll außerdem modular aufgebaut sein. Dies bedeutet, dass eine Wiederverwendbarkeit von Softwareteilen in zukünftigen Projekten sichergestellt werden soll und neue Softwareteile möglichst einfach in das System integriert werden können. Eine eher geringe Komplexität ermöglicht die schnelle Einarbeitung von Studenten und Mitarbeitern in das bestehende System. Weiterhin muss das System eine ausreichende Leistung zur Verfügung stellen, damit auch komplexe medizinische Bildverarbeitungsaufgaben in einem sinnvollen Zeitrahmen durchgeführt werden können. Da das System zukünftig für eine Vielzahl von medizinischen Softwareprojekten genutzt werden soll, müssen die verwendeten Komponenten eine hohe Flexibilität aufweisen, um damit sehr unterschiedliche medizinische Problemstellungen lösen zu können.

Diese grundlegenden Anforderungen werden um Anforderungen an die RECIST-Anwendung im Speziellen ergänzt. Eine Auflistung aller Anforderungen an die Software befindet sich im Anhang der Arbeit.

3.1.2 Konformität zu gesetzlichen Anforderungen

Medizinische Software kann nach dem Medizinproduktegesetz ein Medizinprodukt sein. Ob es sich um ein Medizinprodukt handelt, wird im Gesetz aus der Zweckbestimmung abgeleitet. Ein Produkt wird zum Medizinprodukt, wenn es u. A. für die Erkennung, Überwachung oder Behandlung von Krankheiten eingesetzt wird. Für Medizinprodukte gelten weitreichende gesetzliche Bestimmungen. Verschiedene Normen müssen zu Rate gezogen werden, u. A. kann die Einrichtung eines Qualitätsmanagementsystems nach DIN EN ISO 13485 notwendig sein [1].

Bei der vorliegenden Entwicklung ist zunächst keine Vermarktung der entstehenden Produkte geplant. Es handelt sich um wissenschaftliche Softwaretools, die nicht für die Behandlung von Patienten eingesetzt werden sollen. Die entwickelte Softwarelösung ist daher nicht zu den gesetzlichen Bestimmungen für Medizinprodukte konform. Dieser Umstand wird vor Herausgabe der Software an externe Partner kommuniziert und deutlich in der App gekennzeichnet. Der Nutzer kann die Software erst verwenden, nachdem er einen entsprechenden Hinweis bestätigt hat.

3.1.3 Verwendete Technologien

Im Folgenden wird beschrieben, welche Technologien zur Umsetzung verwendet wurden. Dies umfasst sowohl die Entwicklungsumgebung, als auch Programmiersprachen und Datenbanksysteme. Abbildung 2 stellt die Systeme in einer Übersicht dar.

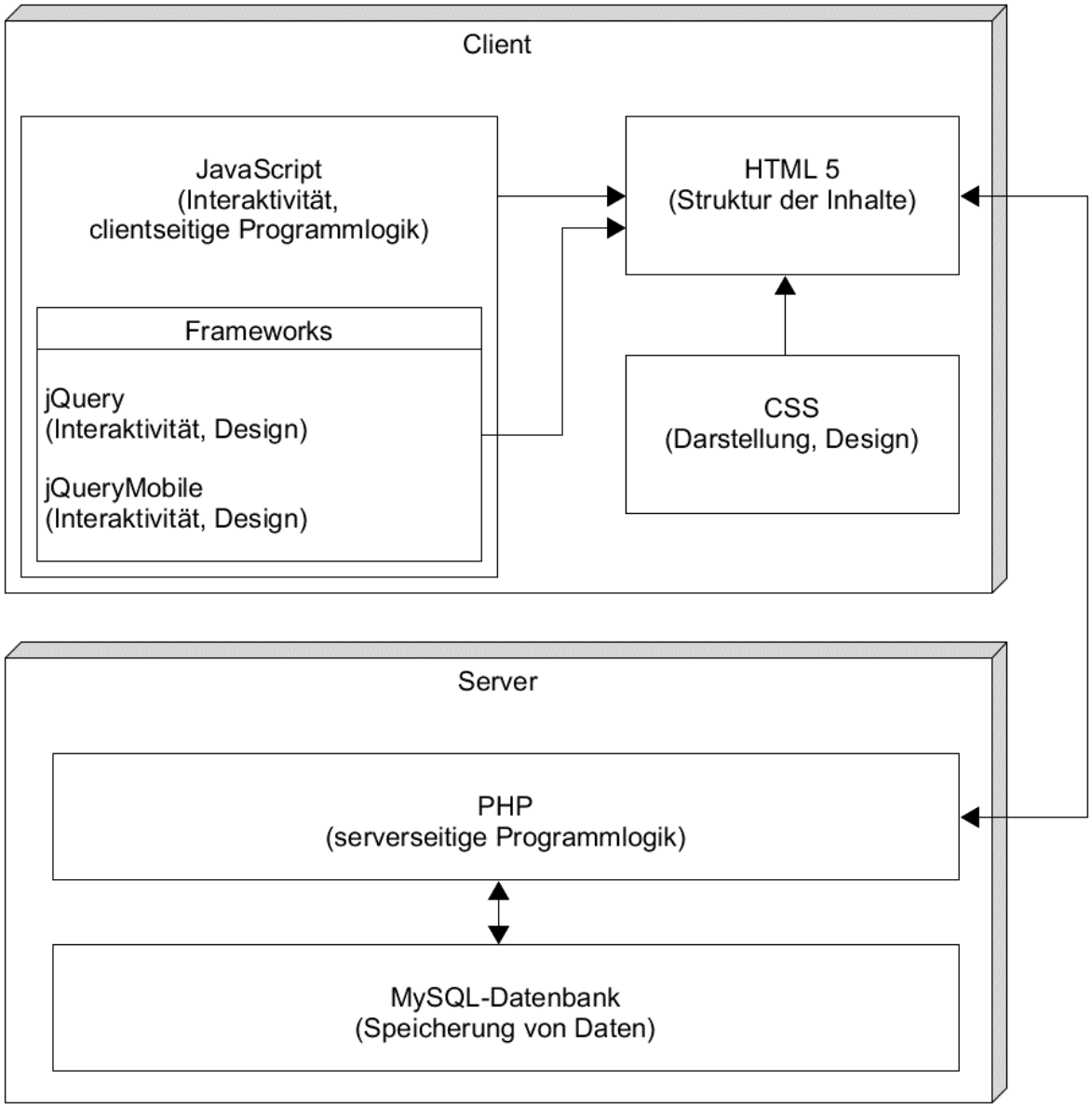

Abbildung 2: Verwendete Programmiersprachen, Frameworks und Datenbank mit jeweils typischer Anwendung.

3.1.3.1 Programmiersprachen

HTML 5

HTML (Hypertext Markup Language) ist eine Auszeichnungssprache. Sie erlaubt die Formatierung von Text und anderen im Web verwendeten Elementen. HTML ist im eigentlichen Sinne keine Programmiersprache. HTML kennt weder Variablen noch Funktionen. Mit HTML alleine können also ausschließlich Webseiten ohne dynamische Inhalte erzeugt werden. HTML befindet sich damit bezugnehmend auf die Architektur auf Präsentationsebene [3].

Java Script

JavaScript ist die Grundlage zur Erstellung von Webanwendungen. Java Script ermöglicht es, HTML-Seiten dynamisch zu gestalten und die Nutzerinteraktion zu verbessern. Die weitverbreitete Programmiersprache wird direkt im Browser ausgeführt. Es handelt sich also um eine clientseitige Sprache [2, S. 19, 19, S. 21 f.].

jQuery und jQueryMobile

Java-Script ist die Grundlage zum Erstellen moderner, interaktiver Webanwendungen. Das JavaScript-Framework jQuery stellt eine Sammlung verschiedener Methoden zur Verfügung, die eine Erstellung interaktiver Webanwendungen erleichtern. So wird es möglich HTML-Elemente zu manipulieren oder Inhalte direkt in eine Webseite zu streamen, ohne das die Seite neu geladen werden muss. Eine Animation der Webseitenübergänge, animierte Menüs oder interaktive Benutzereingabefelder können mit der Erweiterung jQueryMobile angepasst an die Anforderungen mobiler Endgeräte erzeugt werden. Damit kann ohne spezielle Designkenntnisse eine funktionale und ansehnliche Benutzeroberfläche geschaffen werden, die dem Nutzer das Gefühl gibt, eine native Software zu bedienen [2].

PHP

PHP ist die am häufigsten verwendete serverseitige Programmiersprache zum Erstellen dynamischer Webinhalte. Im Gegensatz zu anderen serverseitigen Sprachen wie Microsoft ASP.NET Framework handelt es sich bei PHP um freie Software. PHP unterstützt den Zugriff auf viele Datenbanken. Häufig wird PHP zusammen mit der Datenbank MySQL verwendet. Die Sprache ist zudem recht leicht erlernbar. So müssen Variablen z.B. weder initialisiert noch deklariert werden. [15, S. 17–19, 18].

Ein großer Vorteil ist das einfache Zusammenspiel mit HTML. So kann HTML Code direkt von PHP erzeugt werden und in den bestehenden Code eingefügt werden. Auf diese Weise wird es möglich, dass Webseiten mit dem Nutzer interagieren.

3.1.3.2 Datenbank

Daten werden mittels der Datenbank MySQL gespeichert. SQL ist hierbei eine Sprache, welche den Zugriff auf verschiede Datenbanken erlaubt. MySQL ist das weitverbreitetste Datenbanksystem und bei einer nicht-kommerziellen Verwendung kostenlos. Die relationale Datenbank speichert Datensätze in Tabellen ab, wobei jeder Spalte ein individueller Datentyp zugeordnet werden kann. Verknüpfungen zwischen Datensätzen werden durch die Verwendung gemeinsamer Spalten in verschiedenen Tabellen realisiert [15, S. 188 ff., 16, S. 21].

Zu Verwaltung der Tabellen steht die Benutzeroberfläche phpMyAdmin zur Verfügung. Der Administrator kann dort ohne Programmierkenntnisse die Struktur der Datenbank aufbauen, also

neue Tabellen anlegen und Datentypen vergeben. Auch können Datensätze angelegt, verändert oder gelöscht werden. Solche Änderungen müssen aber auch durch den Benutzer der Software erfolgen können. Hierzu bietet PHP eine Schnittstelle, um auf SQL-Datenbanken zuzugreifen zu können [15, S. 188 ff.].

Medizinische Bilddaten werden nicht direkt in der Datenbank abgespeichert. Stattdessen wird eine Verknüpfung in Form einer URL zur Bilddatei in der Datenbank abgelegt. So kann später das Ablageverzeichnis der Datei identifiziert werden. Auf diese Weise können beliebige Datenformate abgespeichert werden [8].

3.1.3.3 Entwicklungsumgebung

Alle Softwareteile wurden unter Verwendung des Texteditors Notepad++ von Don Ho geschrieben. Der Editor Notepad++ stellt zahlreiche Funktionen zum Schreiben von Programmcode zur Verfügung. Unter anderem markiert die Software die Syntax vieler Programmiersprachen und unterstützt den Anwender beim Layout des Programmcodes.

Der Test von Programmcode erfolgt mit den Browsern Mozilla Firefox und Google Chrome sowohl auf einem PC mit dem Betriebssystem Windows 10, als auch auf einem Samsung Galaxy S5 Smartphone mit dem Betriebssystem Android 5.0.

Als serverseitige Programmiersprache muss PHP-Code für seine Ausführung und daher auch zum Test auf einem Server abgelegt werden. Dort muss die verwendete PHP-Version installiert sein. Es ist möglich, eine geeignete Serverumgebung auf einem lokalen Computer zu installieren oder einen realen Server zu verwenden. Bei vorliegender Arbeit werden beide Methoden angewendet.

Mit XAMPP kann eine Serverumgebung auf einem lokalen Computer zum Test der Software installiert werden. XAMPP ist eine Apache-Distribution, die recht einfach auf einem lokalen Windows-System installiert werden kann. Neben anderen Komponenten wird auch eine PHP-Version und eine MySQL-Datenbank eingerichtet. Werden Dateien im Ordner htdocs des Anwendungsverzeichnisses abgelegt, können diese mittels eines Internetbrowser durch Eingabe von „localhost“ in der Adresszeile eines Internetbrowsers ausgeführt werden. Dies erlaubt den Test der Web-Applikation ohne der Notwendigkeit, einen externen Server einzurichten [15, S. 579 f.].

Weiterhin ist es möglich, die Software auf einem angemieteten Server zu testen. Im Internet gibt es zahllose Anbieter von Webhosting-Angeboten, die ein Testen des Programmcodes erlauben. Es ist darauf zu achten, das auf dem verwendeten Server PHP in der gewünschten Version installiert ist und der Server so konfiguriert werden kann, dass beim Auftreten von Programmfehlern eben diese angezeigt werden. Ein Debugging des Programmcodes ist sonst kaum möglich.

Falls nicht bereits geschehen, kann eine Fehleranzeige durch das Einfügen der Codezeile „php_flag display_errors on“ in die Datei .htaccess initiiert werden [17].

3.2 Beispielanwendung RECIST

Ziel ist es, ein Softwaremodul zu entwickeln, das einem Mediziner das Tracking von Läsionen und eine Bewertung der Befunde mittels RECIST-Kriterien ermöglicht.

3.2.1 Response Evaluation Criteria in Solid Tumors (RECIST)

RECIST (Response Evaluation Criteria in Solid Tumors) sind Kriterien, die eine Bewertung des Ansprechens von soliden Tumoren erlauben. Sie werden innerhalb einer strukturierten Befundung in der Radiologie heute insbesondere in Studien eingesetzt, um vergleichbare und wissenschaftlich fundierte Ergebnisse hervorbringen zu können. Es existieren eine Vielzahl weiterer veröffentlichter Bewertungskriterien, die sich Abhängig von der Art der Erkrankung unterschiedlich gut für eine Befundung eignen. Die RECIST – Kriterien wurden bisher in 2 Versionen, 1.0 und 1.1, veröffentlich. Beide Versionen basieren auf dem im Folgenden beschriebenen, grundlegenden Konzept [6].

Geeignete Läsionen werden als Target-Läsionen klassifiziert, deren Durchmessersumme zu verschiedenen Zeitpunkten gemessen wird. Die Messdaten werden aus Bildinformationen gewonnen, die mittels verschiedener Aufnahmemodalitäten akquiriert werden. In der Regel werden hierfür Computertomographie oder Magnetresonanztomographie verwendet. Aus dem Vergleich der über die Zeit gemessenen Summen der Durchmesser, insbesondere zum ersten gemessenen Wert (Baseline) und zum niedrigsten Wert (Nadir) der Messung, kann mittels der Kriterien das Ansprechen der Erkrankung auf eine Behandlung bewertet werden [6].

Läsionen, die sich für eine Verwendung als Target-Läsion nicht eignen, werden Non-Target Läsionen genannt. Deren genauer Durchmesserwert nimmt keinen Einfluss auf die Bewertung, Anwesenheit und Ansprechen werden jedoch dokumentiert. Auch neue Läsionen, die zum Zeitpunkt der ersten Messung nicht vorhanden waren, werden schriftlich erfasst [6].

Anforderungen an Target-Läsionen [5]:

- Maximal fünf, höchstens zwei pro Organ
- Lymphknoten, wenn Kurzachsendurchmesser ≥ 15 mm
- osteolytische Knochenläsion mit Weichteilanteil, wenn maximaler Durchmesser ≥ 10 mm
- zystische Tumorläsion, wenn maximaler Durchmesser LAD ≥ 10 mm

Anforderungen an Non-Target-Läsionen [5]:

- Alle Läsionen, die wegen Einschränkungen nicht als Target-Läsion aufgenommen werden können
- Lymphknoten nur bei Kurzachsendurchmesser zwischen 10 mm und 15 mm

- Bei Läsionen, die zu klein für eine exakte Messung sind, wird ein Durchmesser von 5 mm angenommen

Tabelle 1 und Tabelle 2 verdeutlichen, wie das Gesamtansprechen ermittelt wird.

Target Läsionen (TL)	**Non Target Läsion (Non-TL)**	**Neue Läsion**	**Gesamtansprechen**
CR	CR	Nein	CR
CR	Kein CR / kein PD	Nein	PR
CR	Nicht alle auswertbar	Nein	PR
CR	Kein PD / Nicht alle auswertbar	Nein	PR
SD	Kein PD / nicht alle auswertbar	Nein	SD
Nicht alle auswertbar	Kein PD	Nein	Nicht auswertbar
PD	Alle	Ja oder nein	PD
Alle	PD	Ja oder nein	PD
Alle	Alle	Ja	PD

Tabelle 1: Tumor-Response, wenn Target-Läsionen vorhanden sind. Abkürzungen: CR (Complete Response); PR (Partial Response); PD (Progressive Disease); SD (Stable Disease). Nach Quelle [6].

Non-Target-Läsionen (Non-TL)	**Neue Läsion**	**Gesamtansprechen**
CR	Nein	CR
Keine CR / kein PD	Nein	Keine CR / kein PD
Nicht alle auswertbar	Nein	Nicht auswertbar
Eindeutig progredient	Ja oder nein	PD
Alle	Ja	PD

Tabelle 2: Tumor-Response, wenn ausschließlich Non-Target-Läsionen vorhanden sind. Abkürzungen: CR (Complete Response); PR (Partial Response); PD (Progressive Disease); SD (Stable Disease). Nach Quelle [6].

3.2.2 Modellierung (UML)

Vor der Implementierung der RECIST-Anwendung müssen Strukturen implementiert werden, die das Anmelden von Nutzern, im medizinischen Kontext in der Regel Ärzten oder Krankenhauspersonal, erlauben und den angemeldeten Benutzern Zugriff auf Patientendaten und Programme gewähren.

Um eine geeignete Strukturierung des Programmablaufs entwickeln zu können, wurden verschiedene Anwendungsfälle mit UML beschrieben. Abbildung 3 zeigt beispielhaft den Anwendungsfall „Anlegen eines neuen Patienten in der Patientenverwaltung“.

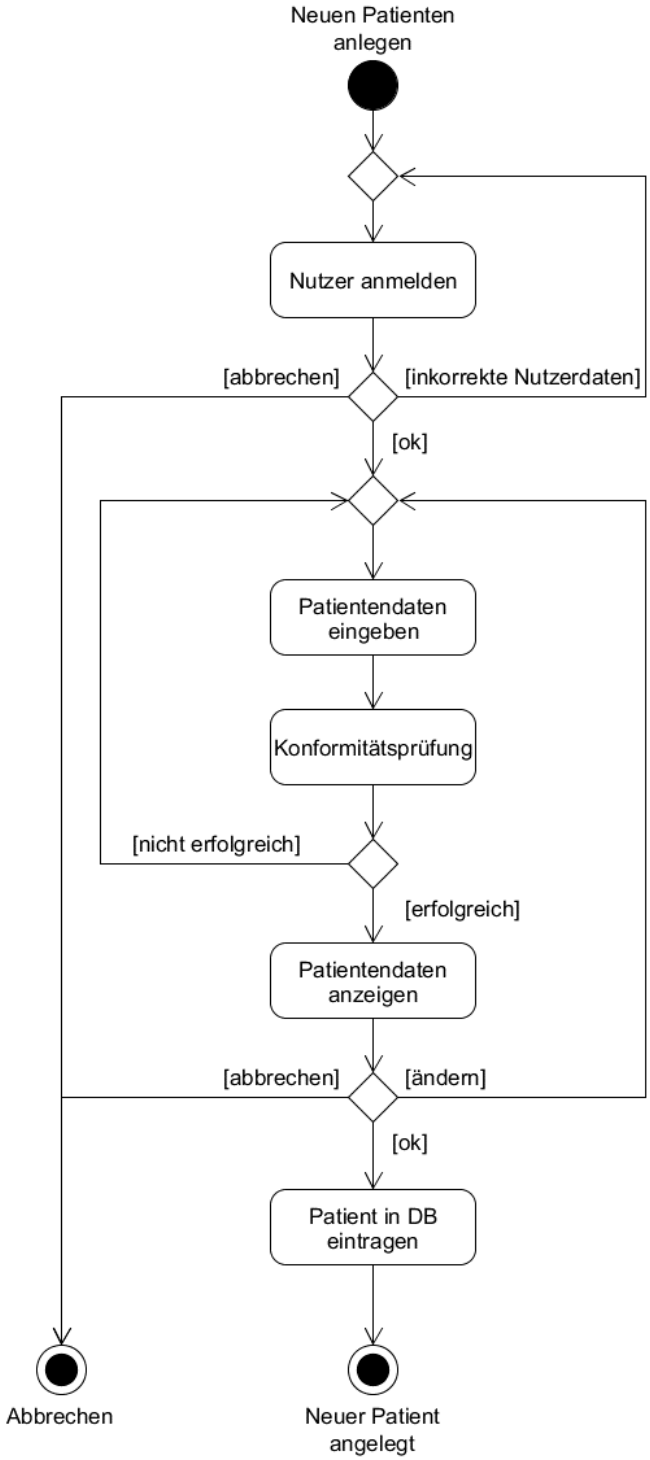

Abbildung 3: UML-Aktivitätsdiagramm für den Anwendungsfall „Anlegen eines neuen Patienten in der Patientenverwaltung“

4 Ergebnis

4.1 Beschreibung der entwickelten Softwarearchitektur

Die Software ist modular aufgebaut. Sie wird zunächst auf zwei Servern installiert.

Es gibt drei Module, die im Folgenden als Verwaltungsmodule bezeichnet werden. Sie bilden die Grundlage des Softwaresystems. Es handelt sich bei diesen Modulen um eine Benutzerverwaltung, einer Patientenverwaltung und um die Programmmodulverwaltung.

In der Benutzerverwaltung werden Systemnutzer abgespeichert. Später sollte hier auch eine Rechteverwaltung implementiert werden. Dies würde es möglich machen, den Zugriff auf Patienten- und Programmdaten verschiedenen Benutzern und Nutzergruppen zu erlauben bzw. zu verwehren. In der Patientenverwaltung werden Studien- und Patientendaten angelegt, bearbeitet und verwahrt. Die Programmmodulverwaltung erlaubt es dem Administrator, sogenannte Programmmodule, wie z.B. RECIST, in das bestehende System zu integrieren.

Abbildung 4 zeigt das Softwarekonzept.

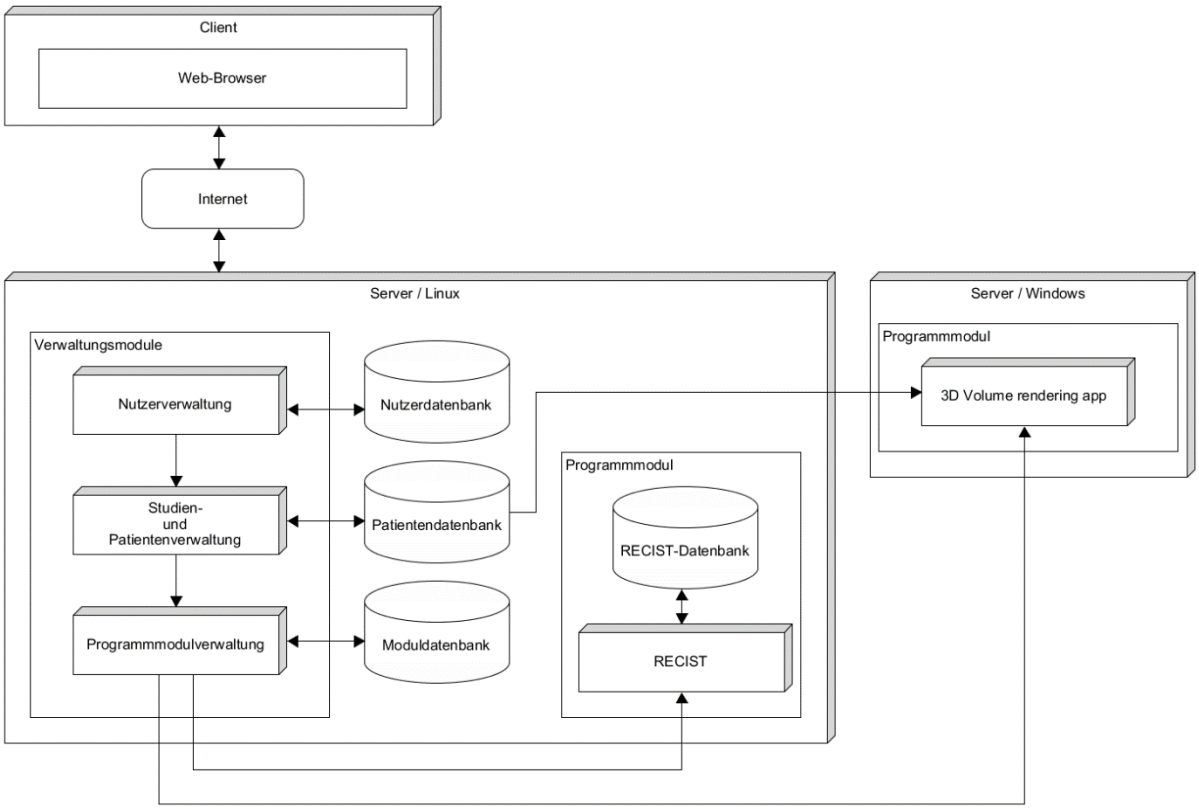

Abbildung 4: Übersicht über das Softwarekonzept. Die Grafik zeigt die Aufteilung der Verwaltungs- und Programmmodule auf die beiden verwendeten Server, sowie genutzte Datenbanken.

4.2 Beschreibung der implementierten Verwaltungsmodule

4.2.1 Benutzerverwaltung

Die rudimentär implementierte Nutzerverwaltung ermöglicht die Registrierung von neuen Nutzern und die Anmeldung eines Nutzers an einer Session nach Eingabe eines Benutzernamens und eines Passwortes. Das Passwort wird in der Nutzerdatenbank als Hashwert abgelegt. Bei einer Anmeldung wird das eingegebene Passwort ebenfalls in einen Hashwert umgewandelt und mit dem in der Datenbank vorliegenden Passwort verglichen. Nur, wenn beide Hashwerte gleich sind, wird der Nutzer an der Session angemeldet und an das Modul Patientenverwaltung weitergeleitet. Ist der Wert unterschiedlich, so kommt der Nutzer zurück zur Anmeldungsseite [15, S. 413 ff.].

Übersicht über den Aufbau der Datenbank des Moduls Benutzerverwaltung.

Spaltenname	Typ	Beschreibung
user_id	int	Automatisch vergebene ID, die eine eindeutige Nutzeridentifikation durch die Software erlaubt. Sie wird beim Anlegen eines neuen Benutzers automatisch um eins inkrementiert.
user_selfid	int	Dem angemeldeten Nutzer kann eine weitere ID zugeordnet werden. Diese kann durch den Nutzer selbst vergeben werden.
user_last_name	varchar	Nachname des Nutzers.
user_first_name	varchar	Vorname des Nutzers.
user_name	varchar	Benutzername des Nutzers.
user_mail	varchar	E-Mail-Adresse des Nutzers.
userPasswordHash	varchar	Passwort des Nutzers als Hashwert abgespeichert.

Tabelle 3: Übersicht der Spalten in der Tabelle user_data.

4.2.2 Patientenverwaltung

In der Patientenverwaltung können neue Patienten mit einer Patienten-ID und einer Patientenbeschreibung angelegt werden. Patienten werden grundsätzlich einer Studie zugeordnet, die ebenso durch eine ID und eine Beschreibung charakterisiert werden kann. Studien können auf vergleichbare Weise angelegt und wieder gelöscht werden. Die Patientenverwaltung verfügt über eine Instant-Suche, die angezeigte Studien- und Patientendaten nach einer Zeichenfolge durchsuchen kann und dem Nutzer das Ergebnis ohne neues Laden der Webseite anzeigen kann. Die Suchfunktion ist in jQuery Mobile bereits implementiert. Es ist zu beachten, dass die Suchfunktion lediglich bereits in der HTML-Seite angezeigte Datensätze durchsuchen kann und keinen direkten Zugriff auf die Datenbank besitzt. Patienten und Studien können als Favoriten markiert werden und werden daraufhin in einem eigenen Tab angezeigt. Außerdem können kürzlich aufgerufene Patienten in einem weiteren Tab angezeigt werden, wobei diese Funktion aber noch nicht vollständig implementiert worden ist [10].

Die Struktur der mit der Patientenverwaltung verbundenen Datenbank wird in Tabelle 4 und Tabelle 5 dargestellt.

Spaltenname	Typ	Beschreibung
patient_id	int	Automatisch vergebene ID, die eine eindeutige Patientenidentifikation durch die Software erlaubt. Sie wird beim Anlegen eines neuen Patienten automatisch um eins inkrementiert.
patient_selfid	int	Dem Patienten kann eine weitere ID zugeordnet werden. Diese kann durch den Nutzer selbst vergeben werden.
study_id	varchar	ID der Studie, in welcher sich der Patient befindet.
patient_description	text	Beschreibung des Patienten.
patient_fav	boolean	0 = Der Patient wurde nicht als Favorit markiert; 1 = Der Patient wurde als Favorit markiert.
patient_date_last_visit	date	Datum des letzten Aufrufs dieses Patienten.
patient_time_last_visit	time	Uhrzeit des letzten Aufrufs dieses Patienten.

Tabelle 4: Tabelle patient_data zeigt den Aufbau der mit der Patientenverwaltung verknüpften Datenbank.

Spaltenname	Typ	Beschreibung
study_id	int	Automatisch vergebene ID, die eine eindeutige Identifikation der Studie durch die Software erlaubt. Sie wird beim Anlegen einer neuen Studie automatisch um eins inkrementiert.
study_selfid	int	Der Studie kann eine weitere ID zugeordnet werden. Diese kann durch den Nutzer selbst vergeben werden.
study_fav	boolean	0 = Die Studie wurde nicht als Favorit markiert; 1 = Die Studie wurde als Favorit markiert.
study_description	text	Beschreibung der Studie.

Tabelle 5: Tabelle study_data zeigt den Aufbau der mit der Patientenverwaltung verknüpften Datenbank.

4.2.3 Programmmodulverwaltung

In der Programmmodulverwaltung werden alle verfügbaren Programmmodule angezeigt. Ein Modul muss zuvor vom Administrator in die entsprechende Tabelle über phpMyAdmin in die Datenbank eingetragen werden. In die Datenbank kann dazu der Name des Moduls, eine Beschreibung, der Link zum Modul, wie auch ein Link für eine passende Grafik eingetragen werden. Abbildung 5 zeigt die Tabelle der Programmmodulverwaltung in phpMyAdmin.

	program_id	program_name	program_description	program_url	program_pic_url
	1	RECIST	Track Tumors with RECIST	recist.php	recist.png
	2	DICOMITE	DICOM-Betrachter		dicomite.png
	3	3DVideoCreator	Create an 3D-Video out of DICOM-Files		3DVideoBetrachter.png

Abbildung 5: Screenshot der Tabelle program_data in phpMyAdmin. Ein Programm bzw. Modul wird durch einen Namen und eine Beschreibung charakterisiert. Die Spalte program_url enthält die Verlinkung zum Modul, program_pic_url die Verlinkung zu einem Icon. Die Programme DICOMITE und 3DVideoCreater wurden nicht implementiert und sind nur zur Veranschaulichung abgespeichert worden.

4.3 Beschreibung der implementierten Programmmodule

4.3.1 RECIST

Das RECIST-Modul übernimmt Daten aus der Nutzerverwaltung und der Patientenverwaltung. Der angemeldete Nutzer kann unter einem ausgewählten Patienten Target-Läsionen anlegen. Das Modul kann aus den abgespeicherten Läsionen die Summe der Durchmesser sowie Veränderungen zur Baseline und Nadir berechnen und anzeigen. Der Verlauf der Durchmessersumme kann in einem Graphen angezeigt werden. Der Graph wurde mit Hilfe der JavaScript-Bibliothek chart.js implementiert.

In der Datenbank werden einzelne Messungen abgespeichert. Der Datensatz beschreibt eine Messung einer Läsion zu einem bestimmten Messdatum. Die Zuordnung der Läsion zu einem Patienten erfolgt

durch eine einmalig vergebene Patienten-ID. Ein Datensatz hat dabei den in Tabelle 6 beschriebenen Aufbau.

Spaltenname	Typ	Bemerkung	Beispiel
lesion_id	int	Einmalige ID für Messung einer Läsion zu einem bestimmten Messdatum.	52
patient_id	int	ID des Patienten, dem diese Läsion zugeordnet ist.	3
lesion_date	date	Datum, an dem die Messung vorgenommen wurde.	2015-05-04
lesion_nr	int	Nummer der gemessenen Läsion.	2
lesion_type	boolean	0 = Target-Läsion; 1 = Non-Target-Läsion	0
lesion_active	boolean	1 = Läsion wird in Berechnung mit einbezogen; 0 = Läsion wird nicht in Berechnung mit einbezogen	1
lesion_name	varchar	Name der Läsion	Lungenmetastase
lesion_node	int	0 = Läsion ist ein Lymphknoten; 1 = Läsion ist kein Lymphknoten	0
lesion_size	varchar	Durchmesser der Läsion in mm	23
lesion_description	text	Beschreibung der Läsion	Metastase im rechten Lungenflügel
lesion_set_sum	int	Durchmessersumme aller zu diesem Zeitpunkt gemessener Target-Läsionen	56
lesion_baseline	int	Abweichung der Durchmessersumme zur Durchmessersumme bei Baseline in %	-21
lesion_nadir	int	Abweichung der Durchmessersumme zur Durchmessersumme bei Nadir in %	+16
lesion_last	int	Abweichung der Durchmessersumme zur Durchmessersumme der vorhergehenden Messung in %	+12

Tabelle 6: Übersicht der Spalten in der Tabelle lesion_data. Ein Datensatz entspricht der Messung einer Läsion zu einem bestimmten Messdatum.

4.3.2 Segmentierung mit Python

Dieses Modul ermöglicht die Segmentierung eines Tumors in einem zweidimensionalen DICOM-Bilddatensatz und die Ausgabe eines Videos, welches den Datensatz mit der erfolgten Segmentierung in einer 3D-Darstellung zeigt. Die erzeugte Volumengraphik dreht sich in einem Video einmal um 360°. Die Implementierung dieses Moduls erfolgt im Zuge der Bachelorarbeit von Andreas Hintermaier [8].

4.4 Beschreibung der Modulschnittstellen

Die Schnittstelle zwischen den einzelnen Komponenten soll nun anhand der Darstellung eines Anwendungsfalles verdeutlicht werden. Der Anwendungsfall zeigt vereinfacht das Erstellen einer Läsion im Programmmodul RECIST. Hierbei wird insbesondere die Kommunikation zwischen den Modulen beschrieben.

Der Arzt meldet sich zunächst mit einem Benutzernamen und Passwort am System an. Die Daten sind in der Benutzerverwaltung hinterlegt und werden überprüft. Bei einer korrekten Eingabe wird der Arzt an das Verwaltungsmodul Patientenverwaltung weitergeleitet. Das Modul Benutzerverwaltung speichert die eindeutige ID des aktuellen Nutzers in der SESSION-Variable. Die Patientenverwaltung kann nun den Nutzer anhand der gespeicherten ID identifizieren. Dem Arzt werden daraufhin alle Studien- und Patientendaten angezeigt, für die er Zugriffsrechte besitzt. Der Arzt wählt einen Patienten aus und öffnet die Programmmodulverwaltung. Hierbei wird die ID des Patienten über die GET-Methode an die Programmmodulverwaltung übergeben. Der Arzt kann nun beispielsweise das Programmmodul RECIST auswählen. Dabei wird die Patienten-ID wieder durch die GET-Methode über die Adresszeile weitergegeben. Das Programmodul RECIST kann den Patienten anhand der Patienten-ID identifizieren. Der Arzt trägt die Messung einer neuen Läsion im Programmmodul RECIST ein. Diese wird zusammen mit der eindeutigen Patienten-ID in der Datenbank abgelegt, um später eine erneute Zuordnung der Läsion zum Patienten möglich zu machen.

Der Datenfluss zwischen den Modulen bei diesem Anwendungsfall wird in Abbildung 6 verdeutlicht.

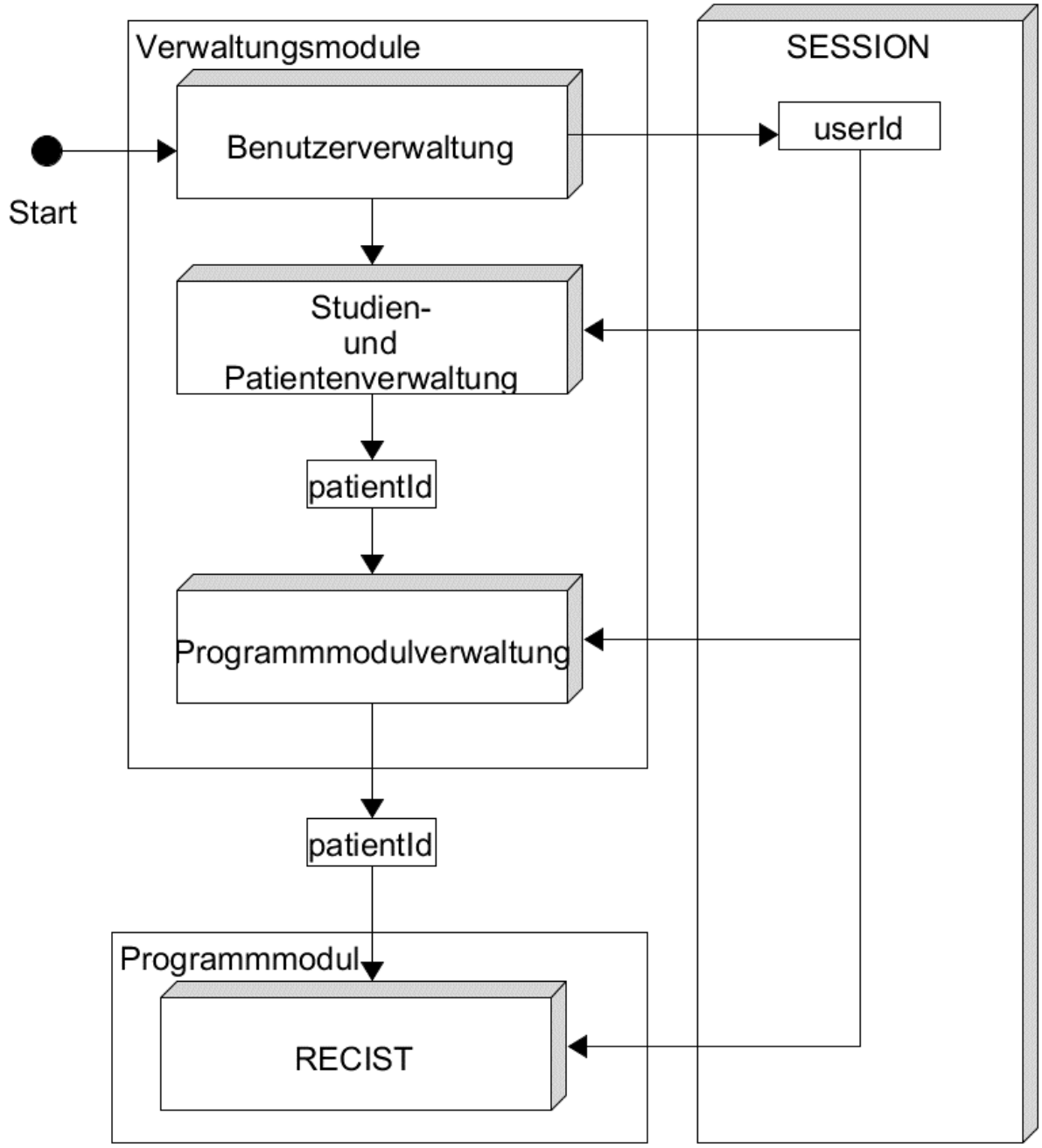

Abbildung 6: Datenfluss zwischen Modulen. Die userId wird im Session-Array abgelegt. Die patientId wird über die Adresszeile per GET-Methode von Modul zu Modul weitergegeben.

4.5 Seitenstruktur

Die Software ist auf eine Vielzahl von Webseiten unterteilt. Grundsätzlich soll darauf geachtet werden, dass mehrere Module nicht innerhalb einer Datei gespeichert werden. Ein Modul soll immer aus einer oder mehreren abgekoppelten Webseiten bestehen. Eine Übersicht zeigt der folgende Webseitenplan.

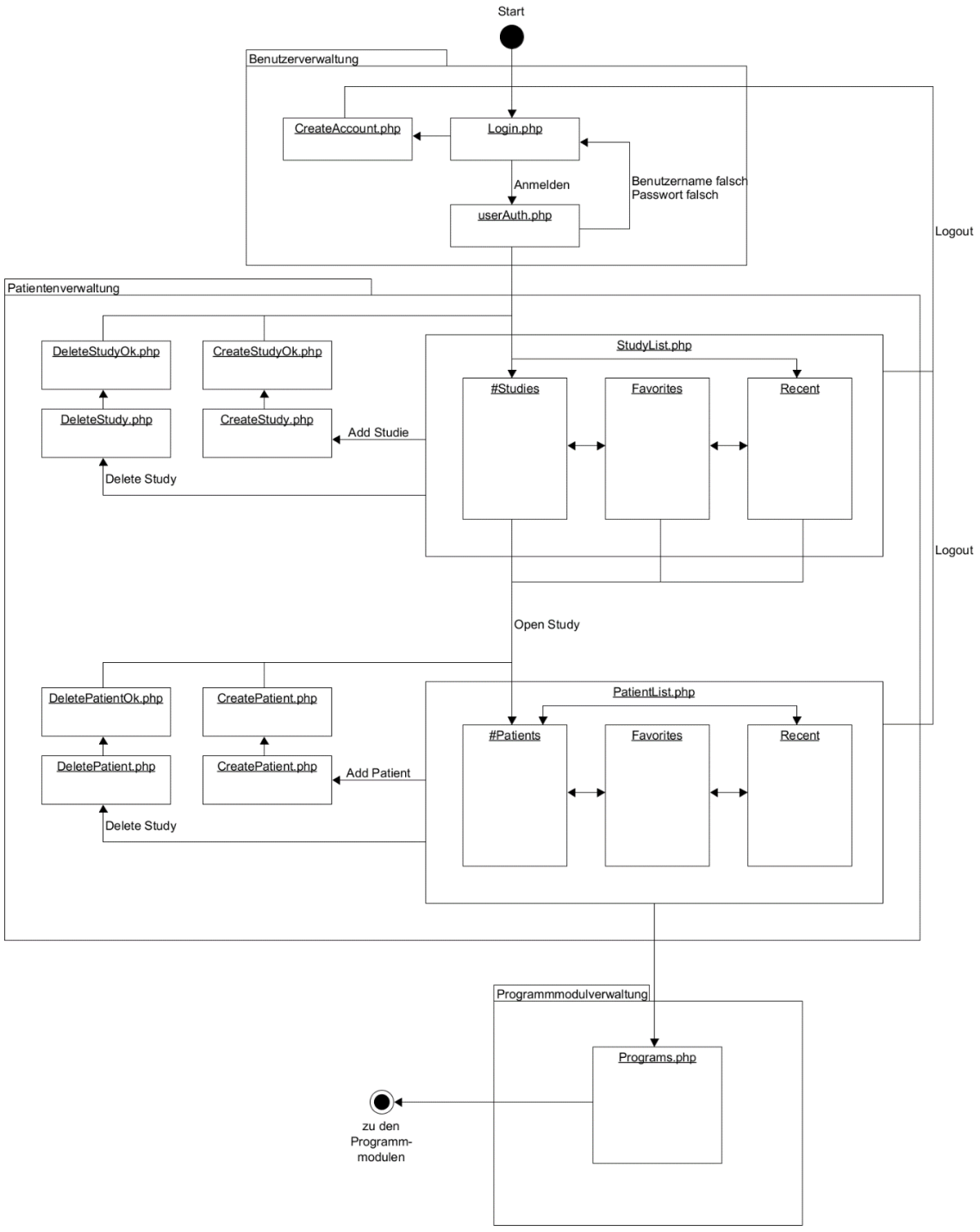

Abbildung 7: Dateien der Verwaltungsmodule und Verknüpfung zwischen den Modulen

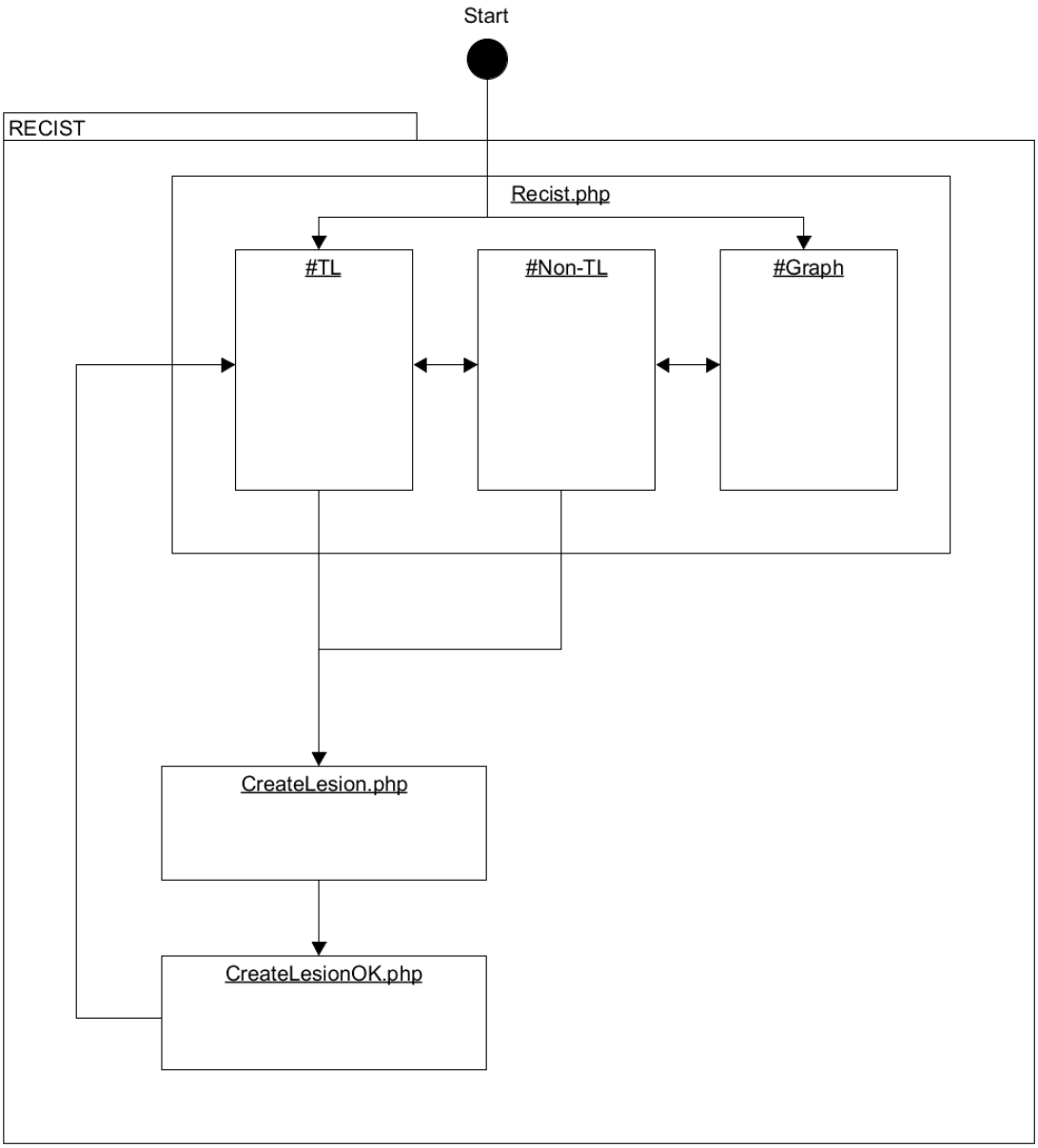

Abbildung 8: Dateien des Programmmoduls RECIST. Dem Benutzer werden gespeicherte Läsionen und eine Auswertung, sowie eine graphische Darstellung des Durchmesserverlaufs der Läsionen angezeigt. Der Benutzer kann Läsionen über CreateLesion.php in die Datenbank einspeichern.

4.6 Präsentation

Die Präsentation des Inhalts erfolgt soweit möglich standardisiert über Komponenten, die jQuery Mobile zur Verfügung stellt. Für Diagramme wird die JavaScript-Bibliothek Chart.js verwendet.

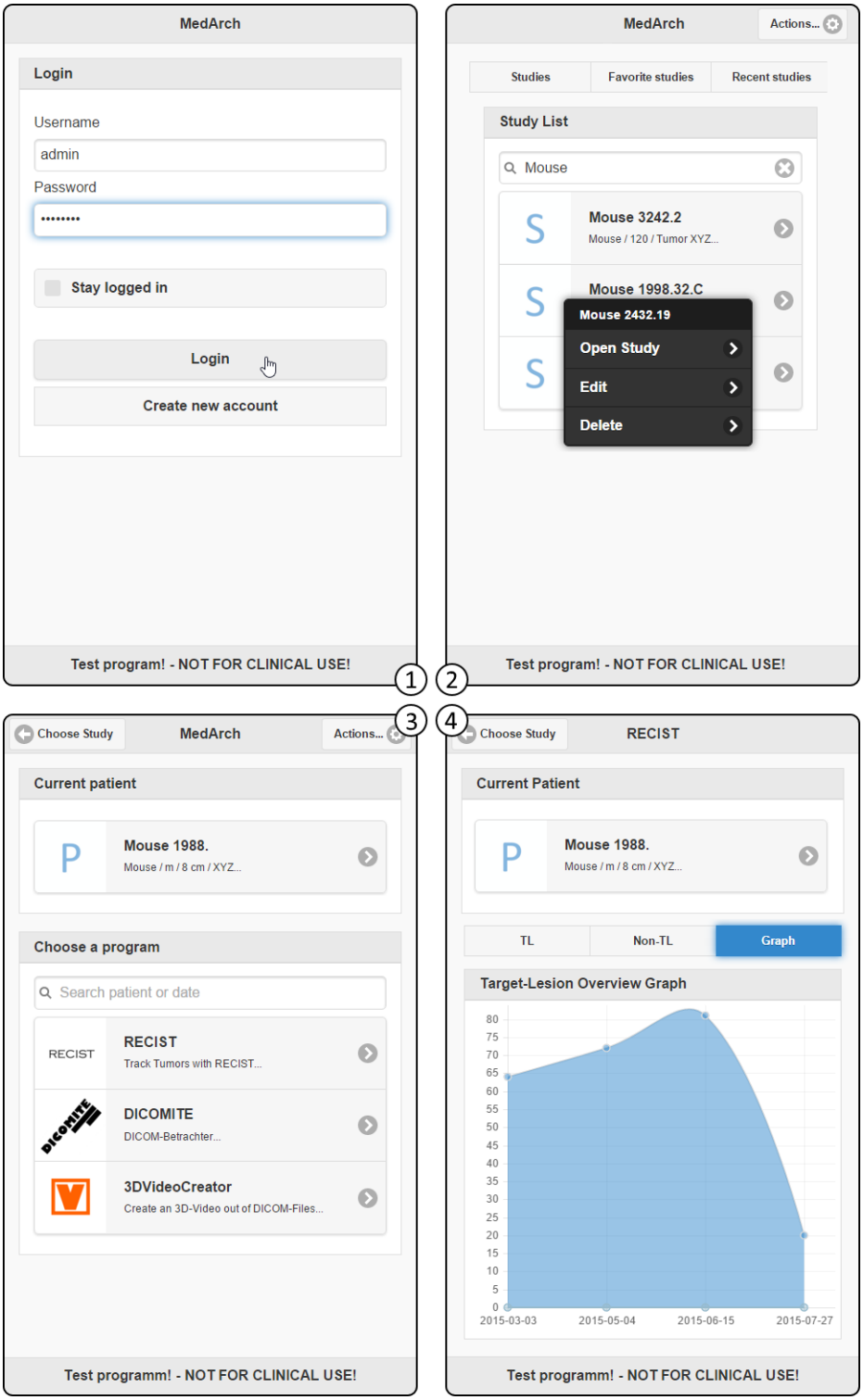

Abbildung 9: Aktuelle Implementierung der Module und User-Interface. 1: Anmeldung am System über das Modul Benutzerverwaltung; 2: Auswahl einer Studie im Modul Patientenverwaltung mit Auswahlmenü und Suchfunktion; 3: Anzeige Programmmodulverwaltung nach Auswahl des Patienten „Mouse 1988“; 4: Darstellung des Verlaufs der Durchmessersummen aller Läsionen über die Zeit im Programmmodul RECIST.

5 Diskussion

5.1 Diskussion der Anforderungen

Die Komplexität der Architektur ist durch die Verwendung lediglich zweier Server und häufig verwendeter Programmiersprachen gering. Dennoch wird durch die Nutzung der JavaScript-Bibliotheken jQuery und jQuery Mobile zukünftigen Entwicklern eine hohe Flexibilität bei der Programmierung bzw. Programmerweiterung erlaubt. Es kann auf einen großen Pool an JavaScript-Funktionen zurückgegriffen werden, um Web-Applikationen zu entwerfen, die dem aktuellen Stand der Technik entsprechen. Die Verwendung von PHP auf der Serverseite macht kurze Einarbeitungszeiten möglich und ist außerdem mit keinen Kosten verbunden. Für alle verwendeten Programmiersprachen ist umfangreiche Literatur erhältlich. Auch im Internet sind auf den Seiten der jeweiligen Hersteller gute Dokumentationen sowie Beispielcode vorhanden [10, 11].

Aus der Analyse der speziellen Anforderung an RECIST ergibt sich weiterhin, dass eine Aufteilung der Module in Verwaltungsmodule und Programmmodule sinnvoll ist. Das RECIST-Modul soll lediglich die Kernaufgabe, also die Speicherung, Auswertung und Anzeige von Läsionsdatensätzen übernehmen. Um die Datensätze Benutzern und Patienten zuzuordnen, sollen standardisierte Verwaltungsmodule in die Architektur implementiert werden, die später ebenso von anderen Programmmodulen genutzt werden können.

Dadurch wird viel Zeit bei der Entwicklung spezieller Programmmodule gespart, da Funktionen nicht mehrmals implementiert werden müssen. Weiterhin wird eine konsistente Bedienbarkeit sichergestellt, die Nutzern der Software eine kurze Einarbeitungszeit ermöglicht. Die Weiterentwicklung der Verwaltungsmodule kann ohne Rücksicht auf die Programmmodule vorangetrieben werden, solange die Übergabe der Daten an die Programmodule standardisiert bleibt. Hierbei soll der Output der Verwaltungsmodule zwar erweitert werden dürfen, bestehende Outputs müssen jedoch erhalten bleiben, um eine fortdauernde Funktion der Programmmodule sicherzustellen.

Auch, wenn dies bei der jetzigen Umsetzung noch nicht notwendig war, ist es vorstellbar, dass auch eine Kommunikation direkt zwischen Programmmodulen stattfinden könnte. So ist es z.B. denkbar, dass ein DICOM-Bildbetrachtungsmodul Zugriff auf die Segmentierungsfunktionen eines anderen Moduls benötigt. Es muss daher bei der Entwicklung weiterer Programmmodule geprüft werden, ob eine Aufspaltung des zu entwickelnden Moduls in mehrere Module sinnvoll ist, wenn Teile des Projekts auch für eine Verwendung in anderen Projekten als geeignet erscheinen. Hierbei ist unbedingt festzulegen, mit welche Out- und Inputvariablen ein Modul umgehen kann, um Inkompatibilitäten zu vermeiden.

5.1.1 Konzeption als Web-App

Eine grundlegende Anforderung an das System ist die Nutzbarkeit auf unterschiedlichen Endgeräten. Das System soll sowohl auf Smartphones, als auch auf einem Computer betrieben werden können. Darum wurde die Software als Web-Applikation konzipiert.

Web-Applikationen werden im Browser ausgeführt. Eine Installation der Software auf einem entsprechenden Endgerät ist nicht erforderlich. Die Technologie erlaubt eine Nutzung der Software auf verschiedenen Plattformen unter Verwendung eines gemeinsamen Programmcodes. Dazu muss der verwendete Browser neben HTML lediglich die Ausführung von JavaScript ermöglichen. Der Betrieb der App ist daher auf allen aktuellen Smartphone-Betriebssystemen, sowie Windows, Linux und iOS möglich. Lediglich die Darstellung des Inhalts muss an das jeweilige Endgerät angepasst werden. Hierbei ist insbesondere auf eine abweichende Bildschirmauflösung und Bildschirmgröße zu achten. Cascading Style Sheets bieten mit den CSS Media Queries eine Möglichkeit an, die Darstellung im Code für Displays von verschiedenen Endgeräten zu konfigurieren. In jQuery Mobile ist bereits eine vorgefertigte Konfiguration vorhanden, die eine reibungslose Darstellung auf unterschiedlichen Endgeräten erlaubt. Diese kann auch nachträglich auf besondere Bedürfnisse angepasst werden. [7, S. 61 ff.]

Außerdem scheint es im Rahmen der Forschung- und Entwicklungsarbeit an der HAW Landshut sinnvoll, dass die Hochschule zunächst volle Kontrolle über die entwickelten Softwarelösungen behält. Die Konzeption als Web-App erlaubt es auf einfache Weise, Partnern den aktuellen Entwicklungsstand der Software aufzuzeigen. Zudem ist es möglich, die Software zu aktualisieren und zu warten, ohne dass Arbeiten auf den Clients durchgeführt werden müssen. Softwareteile können bei Bedarf wieder aus dem Internet entfernt werden. Eine unkontrollierte Weiterverbreitung von Software kann nahezu ausgeschlossen werden.

Die Rechenleistung von Smartphones hat in den letzten Jahren zwar enorm zugenommen, besonders Bildverarbeitungsaufgaben stellen aber immer noch hohe Anforderungen an die Verarbeitungsgeschwindigkeit. Daher ist es notwendig, rechenaufwendige Prozesse, wie z.B. die Segmentierung eines Tumors, auf einem externen Server durchzuführen und lediglich das Ergebnis einer Berechnung an den Client zu übertragen. Durch die in Web-Anwendungen verwendete Server-Client-Architektur wird dies möglich.

5.1.2 Modularität der Benutzeroberfläche

Durch die Verwendung des Web-Frameworks jQueryMobile (jQuery Foundation) wird ein hoher Grad an Modularität bezüglich der Benutzeroberfläche erreicht. Entwickelte Applikationen verfügen durch die Verwendung des Frameworks über ein konsistentes Design. Der Entwickler kann aus verschiedenen Objekten auswählen und ohne Designkenntnisse eine auf verschiedenen Plattformen ausführbare Applikation erstellen. Auf diese Weise kann sich die Entwicklungsarbeit vorwiegend auf die Programmlogik konzentrieren.

5.2 Ausblick

Aufgrund des großen Umfangs dieses Softwareprojektes konnten nur grundliegende Strukturen implementiert werden. Es steht eine rudimentäre Benutzerverwaltung zur Verfügung, die die Erstellung von Benutzern mit Benutzername und Passwort erlaubt. Eine Studien- und Patientendatenbank wurde angelegt. Der Benutzer kann Studien und Patienten jeweils mit ID und Beschreibung im System abspeichern. Die Studien- und Patientendatenbank kann über eine Instant-Suche durch den Nutzer nach Daten abgesucht werden. Es ist weiterhin durch den Benutzer möglich, die Daten aus dem System zu löschen.

Eine Moduldatenbank wurde angelegt. Ein Administrator kann neue Programme mit Name, URL und einer passenden Abbildung über phpMyAdmin in das System integrieren. Der Nutzer erhält nach Auswahl eines Studien/Patientendatensatzes nutzbare Programme angezeigt und kann diese im Anschluss öffnen.

Im der RECIST-Modul können für einen ausgewählten Patienten Target-Läsionen angelegt werden. Die Berechnung der Summe, sowie Abweichung von der Baseline und von NADIR wurden implementiert. Läsionen werden einem Patienten über die Patienten-ID zugeordnet. Die Beschreibung von Non-Target-Läsionen ist in der Tabellenstruktur der Datenbank vorgesehen, ist in der vorliegenden Version aber noch nicht implementiert worden.

Aufgrund des erheblichen Softwareumfangs konnte die Implementierung noch nicht abgeschlossen werden. Die Verwaltungsmodule müssen um eine Vielzahl von Funktionen erweitert werden. Dem RECIST-Modul fehlt insbesondere noch die Möglichkeit der Auswertung der eingespeicherten Läsionen nach RECIST 1.1.

Im Folgenden soll ein zentraler Ausblick auf notwendige Folgearbeiten gegeben werden.

<u>Vereinfachung der Anzeige von Patienten und Studien</u>

Die Webseiten zum Anzeigen von Patienten und Studien gleichen sich in ihrer Struktur. Es wird lediglich auf verschiedene Tabellen in der Datenbank zugegriffen. Auch die Möglichkeiten zum Anlegen, Verändern und Löschen von Studien- und Patientendaten befinden sich in der aktuellen Version des Programms auf verschiedenen Webseiten. Um die Komplexität der Webseitenstruktur zu verringern, sollte die Patientenliste mit der Studienliste vereinigt werden. Diese gemeinsame Webseite soll abhängig von einer übergebenen Variablen die Patientenliste oder die Studienliste anzeigen.

Die Webseiten zum Anlegen, Verändern und Löschen von Datensätzen sollten ebenfalls in einer Webseite vereinigt werden. In Abhängigkeit einer übergebenen Variablen befindet sich die Seite im Modus zum Anlegen, Verändern oder Löschen eines Datensatzes. Eine zweite Variable ermöglicht eine Unterscheidung der Datenbank, in welcher die entsprechende Aktion ausgeführt werden sollte. Der hierzu notwendige Datenbankzugriff kann in einer externen Datei konfiguriert werden.

Überprüfung von Eingaben

Jede Eingabe des Nutzers in die Software ist mit einem Sicherheitsrisiko verbunden. Eine SQL-Injection erlaubt es einem Angreifer, durch geschickte Ausnutzung der Eingabefelder Datenbankabfragen zu generieren und so Zugriff auf Daten zu erhalten oder gar Daten zu verändern. Daher muss jede Eingabe überprüft werden, um einen ausreichenden Datenschutz zu gewährleisten. In der Regel soll der Benutzer ausschließlich Zeichen eingeben können, die für eine Eingabe der entsprechenden Information notwendig sind. Insbesondere muss die Eingabe von Programmcode ausgeschlossen werden. PHP stellt Funktionen zur Verfügung, die eine serverseitige Überprüfung der Eingaben erlauben [4].

Zusätzlich kann eine Überprüfung der Eingabefelder durch Java-Skript sinnvoll sein. Hierdurch kann erreicht werden, dass der Benutzer schon bei der Eingabe von Daten über fehlerhafte Eingaben informiert wird. Allerdings muss beachtet werden, dass grundsätzlich die Möglichkeit der Deaktivierung von Java-Skript browserintern oder über die Installation von Browser-Add-Ons möglich ist. Eine serverseitige Überprüfung durch PHP ist daher in jedem Fall sinnvoll [4].

Übergabe von Variablen zwischen Webseiten

Daten, die über das Array &_GET() an die nächste Webseite übergeben werden, sind grundsätzlich für den angemeldeten Nutzer in der Adresszeile sichtbar. Dies ermöglicht es dem Nutzer, die übergebenen Informationen einzusehen und zu verändern. Bei der Übergabe muss also überlegt werden, ob der Nutzer die übergeben Information einsehen darf.

Eine weitere Möglichkeit ist die Speicherung von Variablen bzw. Informationen im Array $_SESSION(). Meldet sich ein Nutzer an, wird eine Session-ID erzeugt, die eine eindeutige Zuordnung des Clients durch den Server erlaubt. Daten im Session-Array bleiben so lange bestehen, bis der Benutzer die Session beendet. In der Regel geschieht dies durch schließen der Webseite bzw. des Internetbrowsers. Das Session-Array erlaubt also einen Transport von Information über Webseiten hinweg. Es muss sichergestellt werden, dass durch die bereits beschriebene Überprüfung von Eingaben die Übernahme einer Session durch fremde Personen ausgeschlossen werden kann. Aufgrund der Komplexität der Session-ID ist ein erraten der ID kaum vorstellbar [15, S. 413 ff.].

Kommunikation zwischen Server und Client

Damit Daten zwischen Client und Server sicher ausgetauscht werden können bzw. die Übertragung nicht abgefangen werden kann, müssen übertragenen Daten verschlüsselt werden. Auch eine Authentifizierung zwischen Client und Server ist wünschenswert, um sicherzustellen, dass der Datenaustausch wirklich zwischen den gewünschten Parteien stattfindet. Das HTTPS-Übertragungsprotokoll ermöglicht eine Verschlüsselung der Daten über TSL und die Authentifizierung über ein signiertes Server-Zertifikat [9].

Verbesserung der User-Expierence durch Ajax

Um Veränderungen an der Datenbank vorzunehmen muss der Nutzer die Daten in ein Formularfeld eintragen und das Abschicken der Daten durch einen Button initiieren. Jede Anlage, Veränderung oder Löschung von Daten, also jeder Datenbankzugriff, geht daher mit einem erneuten Laden einer

Webseite einher. Der Benutzer hat das Gefühl, eine gewöhnliche Internetseite zu bedienen. Um das Verhalten einer nativen Software zu imitieren, muss eine dynamische Veränderung der Webseite auch unter Berücksichtigung von Datenbankinhalten sichergestellt werden. Diese Dynamik kann in vorliegender Architektur durch JavaScript erzeugt werden. Ajax erlaubt z.B. die Durchführung von Datenbankabfragen, ohne das ein erneutes Laden der Webseite notwendig ist. Der Nutzer kann Inhalte also unmittelbar verändern. Die Nutzung von Ajax führt allerdings zu einer erhöhten Komplexität der Programmierung und wird daher im beschriebenen ersten Prototypen einer solchen Anwendung noch nicht verwendet. Das Konzept ist jedoch in jQuery enthalten. Entsprechende JavaScript-Funktionen können bei eingebundener Bibliothek direkt verwendet werden. [19, S. 391 ff.]

Auswahl weiterer Bewertungskriterien

Neben RECIST in der Version 1.1 gibt es eine Vielzahl weiterer Bewertungskriterien für onkologische Erkrankungen. Genannt seien an dieser Stelle beispielsweise die irRC (Immune-related Response Criteria), deren Verwendung bei einer Behandlung durch eine Immuntherapie sinnvoll sein kann [20].

Eine Implementierung weiterer auswählbarer Kriterien würde eine umfangreiche Befundung verschiedener onkologischer Erkrankungen ermöglichen. Im einfachsten Fall kann dies erreicht werden, indem für verschiedene Bewertungskriterien jeweils eine eigene Programmlogik zur Bewertung implementiert wird. Dadurch müssen keine weiteren Anpassungen in der Software durchgeführt werden.

Läsionen von der Bewertung ausschließen

Um dem Arzt eine möglichst hohe Flexibilität bei der Befundung zu ermöglichen, sollte es möglich sein, abgespeicherte Läsionen vorübergehend von der Befundung auszuschließen. Dies kann erreicht werden, indem die Auswertelogik lediglich Läsionen in die Bewertung einschließt, bei denen ein Flag anzeigt, dass die Läsion bewertet werden soll. Wird das Flag auf 0 gesetzt, z.B. über einen Schalter in der Applikation, behandelt das Programm die Läsion, als wäre sie nicht vorhanden. Ein solcher Ausschluss von der Bewertung muss für den Benutzer jederzeit sichtbar sein, z.B. durch eine farbliche Kennzeichnung der entsprechenden Läsion.

Das letzte Wort hat der Arzt.

Eine Befundung erfolgt letztendlich immer durch den Arzt selbst. Software soll lediglich unterstützend auf den Prozess der Befundung einwirken. Nur der Arzt kennt den Patienten und die individuell stets unterschiedlichen Erscheinungsformen einer Erkrankung. Es muss für einen Arzt daher möglich sein, sich über die Bewertungslogik der Software hinwegzusetzen. Dennoch soll die Software den Arzt darüber informieren, dass seine Eingaben nicht mit den ausgewählten Kriterien vereinbar sind und ihn über die Gründe der inkorrekten Eingabe informieren. Der Arzt sollte in diesem Fall eine Meldung bestätigen müssen, die darauf hinweist, dass die Bewertung nicht mehr konform zu den gewählten Kriterien sein wird. Dies muss dann für den Nutzer deutlich sichtbar im Programm gekennzeichnet werden.

Automatische Datenakquisition aus Bilddaten

In der vorliegenden Version werden Messdaten direkt in die RECIST-Anwendung eingetragen. Es ist vorstellbar, dass der Messvorgang direkt in die Software integriert wird. Die aufgenommenen Bilddaten müssten dazu zunächst in das Programm geladen werden.

Die Bilddaten werden automatisiert an ein weiteres Programmmodul zusammen mit den Bildinformationen übergeben. Hier können Läsionen sowohl automatisiert, also auch durch Eingreifen des Nutzers segmentiert werden. Nach der Messung der Durchmesser der segmentierten Läsionen werden diese an die RECIST-Anwendung übergeben. Die übermittelten Daten werden dem Arzt in der RECIST-App angezeigt. Der Arzt kann die Daten anpassen oder direkt in die Datenbank übernehmen. Auch eine Archivierung der Bilddaten wäre hier sinnvoll.

Teile des Softwaremoduls zur Segmentierung und Vermessung von Läsionen werden in der Bachelorarbeit von Herrn Andreas Hintermaier beschrieben [8]. Eine Kombination der Module ist derzeit nicht implementiert. Eine Implementierung ist in Folgearbeiten vorstellbar.

Alternativ wäre die Anbindung an ein bestehendes PACS-System denkbar. Hier müsste allerdings zunächst eine Prüfung rechtlicher Aspekte durchgeführt werden.

5.3 Fazit

Mit dieser Arbeit wurde eine Grundlage für zukünftige Softwareprojekte im Studiengang Biomedizinische Technik an der Hochschule Landshut geschaffen. Die Funktion des beschriebenen Softwarekonzepts wurde anhand der teilweisen Implementation einer RECIST-Anwendung gezeigt. Die bestehenden Module können in zukünftigen Arbeiten weiter verbessert und ausgearbeitet werden. Neue Module können dem bestehenden System hinzugefügt werden, um ein umfangreiches Softwarepaket für zahlreiche medizinische Anwendungen zu generieren.

6 Anhang

ID	Name der Anforderung	Beschreibung der Anforderung	Priorität	Systemteil
1	**Mobile App**	Das System muss auf Smartphones und Tablets ausführbar sein.	1	Architektur
2	**Modularität**	Das System muss die Wiederverwendbarkeit von Programmteilen für weitere Softwareprojekte ermöglichen.	1	Architektur
3	**Erweiterbarkeit**	Neue Softwaremodule müssen dem System später einfach über einen Datenbankeintrag hinzugefügt werden können.	1	Architektur
4	**geringe Komplexität**	Es sollen häufig verwendete und schnell erlernbare Programmiersprachen eingesetzt werden.	1	Architektur
5	**Flexibilität**	Die verwendeten Technologien müssen für eine möglichst große Zahl unterschiedlicher medizinischer Softwareprojekte geeignet sein.	1	Architektur

Tabelle 7: Grundsätzliche Anforderungen an das Softwarekonzept.

ID	Name der Anforderung	Beschreibung der Anforderung	Priorität	Systemteil	Status
1	**Nutzerprofil erstellen**	Das System muss dem Nutzer die Möglichkeit bieten, ein Benutzerkonto mit E-Mail und eigenem Passwort anzulegen.	1	Architektur	✓
2	**Kennwort anfordern**	Das System wird dem Nutzer die Möglichkeit bieten, ein neues Kennwort anzulegen, falls das alte Kennwort vergessen wurde.	3	Architektur	
3	**Rechteverwaltung**	Das System wird fähig sein, den Nutzerzugriff auf verschiedene Datenbanken und Programmfunktionen zu beschränken.	3	Architektur	
4	**Patientendaten eintragen**	Das System muss dem Nutzer die Möglichkeit bieten, die Patientendaten "ID" und "Beschreibung" in ein Formular einzutragen.	1	Architektur	✓
5	**Patientendaten ändern**	Das System muss dem Nutzer die Möglichkeit bieten, die Patientendaten "ID" und "Beschreibung" über ein Formular zu verändern.	1	Architektur	
6	**Patienten speichern**	Das System muss dem Nutzer die Möglichkeit bieten, Patientendaten in einer externen Datenbank abzuspeichern.	1	Architektur	✓
7	**Patientenliste**	Das System muss dem Nutzer die Möglichkeit bieten, eine Liste gespeicherter Patienten anzeigen zu lassen.	1	Architektur	✓
8	**Patienten suchen**	Das System wird dem Nutzer die Möglichkeit bieten, die Patientendatenbank zu durchsuchen.	3	Architektur	✓
9	**Favoriten anlegen**	Das System soll dem Nutzer die Möglichkeit bieten, besonders häufig benötigte Patienten als "Favoriten" zu markieren.	2	Architektur	✓

Tabelle 8: Anforderungen (1-9) an die Module. Der Status beschreibt, ob die Anforderung bereits erfüllt ist.

10	**Favoriten anzeigen**	Das System soll dem Nutzer die Möglichkeit bieten, als "Favoriten" markierte Personen in einer Liste anzeigen zu lassen.	2	Architektur	✓
11	**Drucken**	Das System soll dem Nutzer die Möglichkeit bieten, den aktuell angezeigten Bildschirminhalt auf einem beliebigen Drucker ausdrucken zu können.	2	Architektur, RECIST	✓ 50%
12	**Zuletzt verwendete Patienten anzeigen**	Das System soll dem Nutzer die Möglichkeit bieten, zuletzt verwendete Personen in einer Liste anzeigen zu lassen.	2	Architektur	✓ 20%
13	**Modulübersicht**	Das System muss dem Nutzer die Möglichkeit bieten, verfügbare Module in einer Liste anzeigen zu lassen.	1	Architektur	✓
14	**Modul hinzufügen**	Das System muss dem Administrator die Möglichkeit bieten, neue Module in das System zu integrieren.	1	Architektur	✓
15	**Target-Läsionen anlegen und speichern**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, unter einem zuvor ausgewählten Patienten Target-Läsionen in Konformität mit RECIST 1.1 anzulegen und in einer Datenbank abzuspeichern.	1	RECIST	✓ 70%
16	**Non-Target-Läsion anlegen uns speichern**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, unter einem zuvor ausgewählten Patienten Non-Target-Läsionen in Konformität mit RECIST 1.1 anzulegen und in einer Datenbank zu abzuspeichern.	1	RECIST	✓ 20%

Tabelle 9: Anforderungen (10-16) an die Module. Der Status beschreibt, ob die Anforderung bereits erfüllt ist. Eine Prozentangabe macht deutlich, dass die Implementierung nicht abgeschlossen ist.

17	**Neue Läsionen anlegen und speichern**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, unter einem zuvor ausgewählten Patienten neue Läsionen in Konformität mit RECIST 1.1 anzulegen und in einer Datenbank zu abzuspeichern.	1	RECIST	✓
18	**Läsion als Lymphknoten markieren**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, Läsionen als Lymphknoten zu markieren.	1	RECIST	✓
19	**Summe der Durchmesser berechnen**	Die RECIST-App muss aus zuvor gespeicherten Target-Läsionen die Summe aller Durchmesser zu einem bestimmten Messzeitpunkt berechnen können.	1	RECIST	✓
20	**Änderung im Vergleich zur "Baseline" berechnen**	Die RECIST-App muss die Änderung der aktuellen Summer der Durchmesser im Vergleich zur Summe der Baselinedurchmesser berechnen können.	1	RECIST	✓
21	**Änderung im Vergleich zur kleinsten Durchmessersumme berechnen**	Die RECIST-App muss die Änderung der aktuellen Summer der Durchmesser im Vergleich zur niedrigsten gemessenen Durchmessersumme berechnen können.	1	RECIST	✓
22	**Änderung zwischen gemessenen Durchmessersummen**	Die RECIST-App muss die Änderung der Durchmessersumme von einer zur nächsten Messung berechnen können.	1	RECIST	
23	**Tumorantwort berechnen**	Die RECIST-App muss die TL-Response berechnen können (CR, PR, SD, PD)	1	RECIST	
24	**Graph anzeigen**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, sich den zeitlichen Verlauf der Tumordurchmesser und Schwellwerte für PR und PD in einem Diagramm anzeigen zu lassen.	1	RECIST	✓ 80%

Tabelle 10: Anforderungen (17-24) an die Module. Der Status beschreibt, ob die Anforderung bereits erfüllt ist. Eine Prozentangabe macht deutlich, dass die Implementierung nicht abgeschlossen ist.

25	**RECIST-Kriterien Übersicht**	Die RECIST-App muss dem Nutzer die Möglichkeit bieten, eine Übersicht der RECIST 1.1 - Kriterien anzeigen zu lassen.	1	RECIST	
26	**Ausfüllhilfe**	Die RECIST-App soll den Nutzer die Möglichkeit bieten, sich Hilfetexte bei der Eingabe von Informationen anzeigen zu lassen.	2	RECIST	
27	**Überprüfung von Eingaben**	Die RECIST-App muss alle Eingaben des Nutzers auf Konformität zu RECIST 1.1 überprüfen und bei Falscheingabe eine Fehlermeldung und einen Korrekturvorschlag ausgeben.	1	RECIST	
28	**Daten nach Excel exportieren (*.xlsx)**	Die RECIST-App wird dem Nutzer die Möglichkeit bieten, RECIST-Daten zur Weiterverarbeitung in das Excel-Format (*.xlsx) zu konvertieren.	3	RECIST	
29	**Offline-Rechner**	Die RECIST-App soll dem Nutzer die Möglichkeit bieten, eine vereinfachte RECIST-Berechnung, bei welcher ausschließlich zwei Durchmessersummen miteinander verglichen werden können, ohne Internetanbindung durchführen zu können.	2	RECIST	
30	**Lokaler Cache**	Die RECIST-App wird eingegebene Daten automatisch offline zwischenspeichern, wenn keine Internetverbindung besteht und bei erneuter Internetverbindung die zwischengespeicherten Daten in eine Datenbank übertragen.	3	RECIST	

Tabelle 11: Anforderungen (25-30) an die Module. Der Status beschreibt, ob die Anforderung bereits erfüllt ist.

7 Danksagung

Für die Unterstützung bei meiner Bachelorarbeit möchte ich mich bei folgenden Personen bedanken:

Frau Prof. Dr. Stefanie Remmele, meiner Betreuerin, die mir zu jeder Zeit mit hilfreichen Anregungen und konstruktiver Kritik zur Seite stand

Herrn Dr. Thorsten Persigehl von der Uniklink Köln für die Idee, eine RECIST-Anwendung zu erstellen und die freundliche Einladung zu einem Gespräch an die Uniklinik Köln

Herrn Prof. Dr. Holger Timinger, der mir alle Fragen zur Entwicklung des Softwarekonzepts beantwortete

8 Quellenverzeichnis

[1] Böckmann, R.-D. and Frankenberger, H. 2015. *MPG & Co*. *Eine Vorschriftensammlung zum Medizinprodukterecht mit Fachwörterbuch*. Praxiswissen Medizintechnik. TÜV Media, Köln.

[2] Bongers, F. and Vollendorf, M. 2011. *jQuery*. *Das Praxisbuch ; [Grundlagen, Einsatz, Praxisbeispiele ; Plug-ins nutzen und erstellen, jQuery UI ; Navigationen, Tooltips, Sprites, Bildergalerien, Formulare, flexible Tabellen, jQTouch]*. Galileo Computing. Galileo Press, Bonn.

[3] Bramer, M. 2015. *Web Programming with PHP and MySQL*. *A Practical Guide*. Springer International Publishing, Cham, s.l.

[4] Clarke, J. 2012. *SQL injection attacks and defense*. Syngress, Burlington, Mass.

[5] DRG. *Kittletaschenkarte RECIST 1.0 und RECIST 1.1*. http://www.onkologische-bildgebung.drg.de /media/document/5580/kitteltaschenkarte-recist.pdf.

[6] Eisenhauer, E. A., Therasse, P., Bogaerts, J., Schwartz, L. H., Sargent, D., Ford, R., Dancey, J., Arbuck, S., Gwyther, S., Mooney, M., Rubinstein, L., Shankar, L., Dodd, L., Kaplan, R., Lacombe, D., and Verweij, J. 2009. New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1). *European journal of cancer (Oxford, England : 1990)* 45, 2, 228–247.

[7] Franke, F. and Ippen, J. 2013. *Apps mit HTML5 und CSS3*. *Für iPhone, iPad und Android*. Galileo Press, Bonn.

[8] Hintermaier, A. 2016. *Entwicklung einer Architektur für modulare medizinische Apps und deren Validierung durch eine 3d-Volume-Rendering App (in Arbeit)*. Bachelorarbeit.

[9] Institute of Electrical and Electronics Engineers, IEEE Symposium on Security and Privacy, SP, and S&P. 2013. *2013 IEEE Symposium on Security and Privacy (SP)*. *19 - 22 May 2013, San Francisco, California, USA*. IEEE, Piscataway, NJ.

[10] 2015. *jQuery Mobile Demos*. http://demos.jquerymobile.com/1.4.5/. Accessed 25 February 2016.

[11] *PHP: Hypertext Preprocessor*. http://php.net/. Accessed 25 February 2016.

[12] *Radiology Tutor on the App Store*. https://itunes.apple.com/us/app/radiology-tutor/ id646539919?mt=8. Accessed 25 February 2016.

[13] *RECIST 1.1 Calculator – Android-Apps auf Google Play*. https://play.google.com/store/apps/ details?id=appinventor.ai_louis_lassalle.RECIST_1_1_Calc&hl=de. Accessed 25 February 2016.

[14] Rubin, D. L., Willrett, D., O'Connor, M. J., Hage, C., Kurtz, C., and Moreira, D. A. 2014. Automated Tracking of Quantitative Assessments of Tumor Burden in Clinical Trials1. *Translational Oncology* 7, 1, 23–35.

[15] Theis, T. 2015. *Einstieg in PHP 5.6 und MySQL 5.6*. *[ideal für Programmieranfänger geeignet ; schnell und einfach dynamische Webseiten entwickeln ; mit vielen Beispielprojekten und Übungsaufgaben ; Formulardaten auswerten und speichern, Sessions, Sicherheit, Grafiken, Ajax u.v.m. ; CD-ROM XAMPP, Notepad++, HTML-Crashkurs, Musterlösungen und alle Code-Beispiele]*. Rheinwerk computing. Rheinwerk Verl., Bonn.

[16] Throll, M. and Bartosch, O. 2008. *Einstieg in SQL*. *Verstehen, einsetzen, nachschlagen*. Galileo Computing. Galileo Press, Bonn.

[17] *Turning On PHP Errors • bavotasan.com*. http://bavotasan.com/2010/turning-on-php-errors/. Accessed 25 February 2016.

[18] *Usage Statistics and Market Share of Server-side Programming Languages for Websites, February 2016*. http://w3techs.com/technologies/overview/programming_language/all. Accessed 25 February 2016.

[19] Wenz, C. 2008. *JavaScript und Ajax. Das umfassende Handbuch ; [Einführung, Praxis, Referenz ; browserübergreifende Lösungen ; Web 2.0: DOM, CSS, XML, Web Services ; neu in der 8. Auflage: Microsoft Silverlight, ASP.NET AJAX 1.0, Ausblick auf Firefox 3 und JavaScript 1.8]*. Galileo Computing. Galileo Press, Bonn.

[20] Wolchok, J. D., Hoos, A., O'Day, S., Weber, J. S., Hamid, O., Lebbe, C., Maio, M., Binder, M., Bohnsack, O., Nichol, G., Humphrey, R., and Hodi, F. S. 2009. Guidelines for the Evaluation of Immune Therapy Activity in Solid Tumors. Immune-Related Response Criteria. *Clinical Cancer Research* 15, 23, 7412–7420.