



TECHNISCHE  
UNIVERSITÄT  
WIEN  
  
VIENNA  
UNIVERSITY OF  
TECHNOLOGY



MEDIZINISCHE  
UNIVERSITÄT  
WIEN

## MASTERARBEIT

# PLANUNG UND REALISIERUNG EINES WEBBASIERTEN, MEDIZINISCHEN INFORMATIONSSYSTEMS ZUR UNTERSTÜTZUNG KLINISCHER MULTICENTERSTUDIEN

Ausgeführt am

Institut für Medizinische Informations- und Auswertesysteme  
der Medizinischen Universität Wien

unter der Anleitung von

Ao. Univ.-Prof. Dipl.-Ing. Dr. Georg Duftschmid

durch

Lukas Gerhold, Bakk. techn.

Leitermayergasse 35

1180 Wien

Wien, 2. Mai 2007

## Kurzfassung

Das System Archimed ist ein Informationssystem zur Unterstützung klinischer Studien. Dieses System ist seit 1997 am AKH Wien unter der Betreuung des Institutes für Medizinische Informations- und Auswertesysteme zur Durchführung von klinischen Studien im Einsatz. Zuletzt erwuchs die Anforderung am AKH mit dem bestehenden System Archimed klinische Multicenterstudien durchzuführen. Aufgrund von Sicherheitsbestimmungen seitens des Rechenzentrums der Medizinischen Universität Wien war eine Erweiterung der Client Server Architektur der bestehenden Applikation nicht umsetzbar. Die graphische Benutzerschnittstelle der Client Server Applikation wäre auf Grund von Benutzbarkeitsüberlegungen ebenfalls zu überarbeiten gewesen. Um dem vorherrschenden Trend zu folgen und Patienteninformationen, statt papierbasiert, direkt über den PC einzugeben und webbasiert zu erfassen, fiel die Entscheidung auf eine Neuentwicklung einer Webapplikation, basierend auf der Systemdatenbank und dem Datenbestand. Der Zugriff der Webapplikation auf den Datenbestand ermöglicht den webbasierten Zugang zu allen klinischen Studien im System. Bei Ankauf einer kommerziellen Standardsoftware wäre ein Zugriff auf den Datenbestand nur durch aufwendige Schnittstellengprogrammierung möglich gewesen. Die Anforderungen an die zu entwickelnde Webapplikation lagen in der Erfassung, dem Monitoring und Datenmanagement von Patienteninformationen sowie der Patientenverwaltung bei Multicenterstudien. Ein Hauptaugenmerk bei der Entwicklung der Webapplikation lag weiters auf der Optimierung der Ergonomie und Benutzbarkeit der graphischen Benutzerschnittstelle.

## Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Typische Aufgaben die im Rahmen klinischer Studien zu realisieren sind...	2
1.2	Software für klinische Studien .....	6
1.3	Motivation.....	7
2	Stand der Forschung.....	9
2.1	Vom CRF zum eCRF .....	9
2.2	Architekturansätze.....	9
2.3	FDA Anforderungen an ein Studiensystem .....	11
3	Methoden .....	22
3.1	Methoden der Systemanalyse.....	22
3.2	Methoden der Systemspezifikation.....	23
3.3	Methoden der Systemauswahl und Eigenentwicklung .....	25
4	Ergebnisse .....	26
4.1	Ergebnisse der Systemanalyse.....	26
4.1.1	Einleitung .....	26
4.1.2	Untersuchungsplan .....	27
4.1.3	Analyse und Bewertung .....	29
4.1.4	Analysebericht.....	37
4.2	Ergebnisse der Systemspezifikation.....	39
4.2.1	Ausgangssituation.....	39
4.2.2	Zieldefinition und Abgrenzung.....	56
4.2.3	Produkteinsatz und Produktumgebung .....	56
4.2.4	Anforderungen .....	57
4.2.5	Weitere Vorgaben .....	60
4.3	Ergebnisse der Eigenentwicklung .....	61
4.3.1	Auswahl der Programmiersprache und der Entwicklungsumgebung ..	61
4.3.2	Namenskonventionen und Codegestaltung .....	63
4.3.3	Integrationsplanung - Bottom Up – Integration.....	64
4.3.4	Implementierung der Klassen.....	65
4.3.5	Implementierung der grafischen Benutzerschnittstelle.....	74
5	Diskussion.....	93
6	Zusammenfassung und Ausblick.....	97
7	Literatur .....	99

# 1 Einleitung

---

*Evidence Based Medicine* (EBM) bedeutet, dass jede am Patienten angewandte diagnostische oder therapeutische Maßnahme praktisch-wissenschaftlich fundiert sein muss. Für diesen Weg zu einer EBM sind *klinische Studien* unerlässlich. Aber auch die Ergebnisse klinischer Studien dürfen durchaus diskutiert werden und nicht jede publizierte Studie liefert überzeugende Ergebnisse (1). Bei einer klinischen Therapie-Studie wird der Einfluss einer medizinischen Behandlung auf eine Krankheit in einem kontrollierten experimentellen Umfeld geprüft. Bei einer Diagnosestudie wird ein neues Diagnoseverfahren im Vergleich zu einem anderen untersucht. Unter Beobachtungsstudien versteht man eine patientenbezogene Datenerhebung, im Gegensatz zu Therapie- und Diagnosestudien erfolgt dabei jedoch keine Intervention am Patienten. Beobachtungsstudien werden auch nicht in einem experimentellen Umfeld durchgeführt. Um möglichst aussagekräftige Resultate zu erzielen benötigt man eine statistisch errechenbare, ausreichend große Zahl an teilnehmenden Patienten (2). Hierfür bietet sich eine Studie an die einmal geplant und vielerorts, also *multizentrisch*, durchgeführt wird. Dieses Vorgehen gewährleistet eine homogene Sammlung an Patientinformationen, die durch ihre Homogenität miteinander vergleichbar sind, und die dann an einer Stelle, der *Studiengentrale*, auswertbar sind. Das *Studiendesign* entscheidet wesentlich darüber, wie aussagekräftig die erhaltenen Daten sind. Folgende Begriffe werden zum Beschreiben einer Studie verwendet.

1. *Aktiv kontrolliert*: Eine Patientengruppe erhält den zu testenden neuen Wirkstoff, die Kontrollgruppe ein etabliertes Medikament, wobei Wirkung und Verträglichkeit miteinander verglichen werden.
2. *Placebokontrolliert*: Zum Vergleich erhält eine Patientengruppe den Wirkstoff, die Kontrollgruppe ein Scheinmedikament, welches keinen Wirkstoff enthält. Ethisch ist diese Studienart nur dann vertretbar wenn noch kein Gold-Standard existiert. Sonst muss eine aktiv kontrollierte Studie als Studiendesign gewählt werden.

3. *Doppelblind*: Weder der behandelnde Arzt noch der Patient wissen, ob es sich bei der angewandten Therapie um Placebo/Gold-Standard oder Verum handelt.
4. *Randomisiert*: Die Entscheidung, welcher Patient welcher Gruppe zugeordnet wird, erfolgt nach dem Zufallsprinzip.
5. *Cross-over*: Die Teilnehmer an der Studie erhalten für einen bestimmten Zeitraum die Kontrollbehandlung und die zu testende Behandlung. Der Einfluss der neuen Behandlung kann somit für jeden einzelnen Patienten ermittelt werden.
6. *Single dose*: Ein Patient erhält die Behandlung innerhalb der Studie nur einmal.
7. *Multiple dose*: Die Behandlung erfolgt pro Patient mehrmals.
8. *Monozentrisch*: Die Studie wird an einer einzigen Institutionen (z.B. Krankenhausabteilung) durchgeführt. Monozentrische Studien sind weniger aufwändig als multizentrische, ihre Ergebnisse müssen aber vor dem Hintergrund evtl. Besonderheiten der jeweiligen Institution bewertet werden.
9. *Multizentrisch*: Die Studie wird (national oder international) in einer Vielzahl von Institutionen (typischerweise Kliniken) - im Folgenden *Zentren* genannt - durchgeführt. Multizentrität ist geeignet, den Einfluss von lokalen Faktoren (wie z. B. Anzahl und Qualifikation der behandelnden Ärzte) auf das Studienergebnis zu bewerten und gering zu halten.

Da die Komplexität solcher Studien mitunter sehr hoch sein kann ist ein computerunterstütztes Studiensystem ein geeignetes Mittel, das Zusammenspiel von Experten wie Medizinern, MTAs (med. techn. Assistenten), Informatikern sowie Statistikern zu erleichtern (3).

## **1.1 Typische Aufgaben die im Rahmen klinischer Studien zu realisieren sind**

Studienzentralen sollen klinischen Forschern eine Unterstützung bei der Methodologie in der klinischen Forschung bieten, und dadurch eine optimale Hilfe bei der Durchführung von kontrollierten klinischen Studien bereitstellen (4).

Die Aufgaben von Studienzentralen sind:

### **Studiendesign**

Eine klinische Studie, hat präzise Fragestellungen die auf eine oder mehrere spezielle Behandlungen abzielen. Eine Studienzentrale soll durch Auswahl des Typus einer Studie (Phase 1, 2, 3, 4 oder Pilotstudie zur Prüfung der Tauglichkeit der These) sowie durch Errechnung der für die Beantwortung der Fragestellung optimalen Anzahl der Patienten (das Patientenkollektiv soll eine hinreichende Größe für das zu erreichende Konfidenzintervall aufweisen damit die nötige Signifikanz der Studie erreicht wird und somit gezeigt wird dass es sich bei dem Resultat der Studie um keinen Zufall handelt), die Forscher bei der Definition der Kriterien nach denen die Behandlungsergebnisse beurteilt werden, unterstützen (siehe Tabelle 1). Je nach Entwicklungsfortschritt werden die Studien in sog. klinische Phasen unterteilt. Die Genehmigung zur nächsthöheren Phase wird von der entsprechenden Aufsichtsbehörde nur dann erteilt, wenn die vorangegangene Studienphase mit Erfolg abgeschlossen wurde.

Phase	Personen	Dauer	Hauptziel
I	Ca. 6-32	Wochen	Pharmakokinetik, Verträglichkeit und Sicherheit des Medikaments (auch als "First in Man" / FIM bezeichnet)
II	Ca 50-200	Wochen bis Monate	Überprüfung des Therapiekonzepts (Proof of Concept, Phase IIa), Findung der geeigneten Therapiedosis (Dose Finding, Phase IIb), positive Effekte der Therapie sollten zu beobachten sein
III	Ca 200 – 10.000	Monate bis Jahre	Signifikanter Wirkungsnachweis (Pivotal Study) und Marktzulassung der Therapie; nach Marktzulassung werden laufende Studien dann zu IIIb-Studien
IV	Ab ca. 1000	Jahre	erfolgen mit bereits zugelassenen Medikamenten und dienen hauptsächlich dem Marketing, der Gewährleistung der Sicherheit des Arzneimittels, z.B. der Feststellung sehr seltener Nebenwirkungen

**Tabelle 1: Typen klinischer Studien (adaptiert von (5)).**

Eine Studienzentrale sollte einem *Studienkoordinator* dabei helfen, die grundsätzlichen Fragenstellungen der Wissenschaftler zu formulieren, und diese mit ihren Antworten im Studienprotokoll festzuhalten:

1. wann und wie der Patient für die Studie rekrutiert wird und wie die Registrierung durchgeführt werden muss, wenn die Behandlung randomisiert (dem Zufallsprinzip folgend) zugewiesen wird
2. welche Informationen man zum Zeitpunkt der Registrierung benötigt,
3. wann und zu wem die Daten geschickt werden müssen,
4. wo die Medikamente zu finden sind,
5. wer zu benachrichtigen ist falls unerwartete Nebeneffekte auftreten,
6. wer das Review durchführt,
7. wer die ausgeschickten Daten bestätigt,
8. wer spezielle Fragen beantwortet betreffend der Eignung eines Patienten, und die Anpassung der Behandlung an den Patienten,
9. sonstige wichtige Fragen die mit der Studie in Zusammenhang stehen.

Der Studienkoordinator ist der Verantwortliche für die Studie.

### **Formulardesign**

Welche Daten im Laufe einer Studie erfasst werden sollen gehört zu den wichtigsten und heikelsten Überlegungen, die vorab geklärt werden müssen.

Kliniker sind sich oft nicht einig, welche *Patienteninformationen* für eine klinische Studie relevant sind und welche nicht. Patienteninformationen sind Daten, die mit einem Patienten in Verbindung stehen und innerhalb dieses Kontexts eine bestimmte Bedeutung haben (z.B. ein Laborparameter eines Patienten). In der Praxis können die meisten Kliniker zwar eine gute Übersicht der relevanten Informationen einer individuellen Sammlung sämtlicher Patienteninformationen liefern, haben aber meist wenig Erfahrung dabei eine gründliche Auflistung sachdienlicher *Patientendatenelemente* zu erstellen. Patientendatenelemente sind Metainformationen die festlegen welche Daten zu einem Patient gespeichert werden. Eine derartige Auflistung wird anhand des Protokolls in Zusammenarbeit zwischen der Studienzentrale und dem Studienkoordinator erstellt. Die Formulare müssen für Statistiker, Datenmanager und dem Studienkoordinator alle relevanten Daten enthalten, die zur Auswertung der Studienresultate, zur Prüfung der Datenkonsistenz, und zur Evaluierung von Fällen notwendig sind. Datenmanager

sind verantwortlich für die Speicherung der Daten und sie sind für die inhaltliche Betreuung der Datenbank zuständig. Daten, die nicht zu dem Zweck der Auswertung, der Prüfung der Datenkonsistenz oder zur Evaluierung von Fällen erhoben werden, würden eine Erhöhung des Arbeitsaufwandes für die Forscher (die die Daten zu erheben hätten), wie auch für die Datenmanager bei der Studienzentrale mit sich bringen.

Das Layout der Formulare ist sehr wichtig, da es die Forscher dabei unterstützen sollte die Formulare so einfach und auch so schnell wie möglich auszufüllen, damit die Information rechtzeitig an die Studienzentrale übermittelt werden kann (6).

### **Patientenregistrierung und Randomisierung**

Damit zu einem Patienten Daten erhoben werden können, die für die statistische Auswertung relevant sind, muss ein Patient bevor die Behandlung anfängt für die Studie registriert werden. Die Studienzentrale kann durch eine Registrierung die Eignung eines Patienten an einer Studie teilzunehmen feststellen, wie auch den täglichen Patientenzuwachs überwachen. Eine der Aufgaben einer Studienzentrale besteht darin den Forschern ein interaktives Registrierungssystem zur Verfügung zu stellen.

Randomisierung bedeutet, dass per Zufall zwei alternative Behandlungsmethoden den Probanden zugeteilt werden. Damit werden Fehler vermieden, die sich bei nicht zufälliger Verteilung ergeben können. Dieses Vorgehen erhöht die Glaubwürdigkeit und Seriosität der Studie (7).

### **Datenerfassung und Qualitätskontrolle**

Weitere essentielle Aufgaben einer Studienzentrale liegen in den Bereichen Datenerfassung und Qualitätskontrolle.

Zu allen an der Studie teilnehmenden Patienten müssen die Daten bei der Studienzentrale zentralisiert und computerisiert gesammelt werden, um sie auszuwerten. Datenmanager sorgen durch die Erstellung von Datenbanktabellen für die notwendigen Datenstrukturen und kümmern sich um die Kommunikation zwischen den Studienzentren und der Studienzentrale, sowie den Transfer der Daten von den Studienzentren zu der zentralisierten Datenbank. Das Ziel der Arbeit von Datenmanagern ist, dass die zu einem Patienten erhobenen Patienteninformationen, exakt den Daten entsprechen, die in der zentralisierten Datenbank zum Zweck der

statistischen Auswertung zusammengeführt worden sind. Es darf nicht passieren, dass Patienteninformationen zu ein und demselben Patienten mehrfach unter einer anderen Patientenidentifikation übermittelt werden, genauso wenig wie es nicht vorkommen darf, dass Patientenidentifikationen durcheinander kommen und nicht mehr klar feststellbar ist welche Daten zu welchem Patienten gehören (8).

Für die Qualität der Daten ist die Kooperation zwischen den an der Studie teilnehmenden Forschern sowie ihren Mitarbeitern, dem Studienkoordinator, den Reviewern (Pathologen, Radiologen, etc.) und den Statistikern verantwortlich. Schließlich haben die Datenmanager die Bemühungen aller zu koordinieren, für die Kommunikation zu sorgen, und sich um die Speicherung der Resultate der Reviews zu kümmern (9).

### **Überwachung der Studie, Monitoring**

Datenmanager können durch eine strenge Überwachung der in der Eingangsphase zu einem Patienten gesammelten Daten rasch entscheiden, ob eine Behandlung sowie administrative Prozesse für den Patienten geeignet sind oder nicht (7).

Beim Monitoring überwachen Personen die Einhaltung der „guten klinischen Praxis“ (*Good Clinical Practice*) während der Studie, und kontrollieren die Durchführung des Studienprotokolls sowie die Dokumentation der entsprechenden Datenerhebungsbögen (10). Unter *Good Clinical Practice* versteht man nach ethischen und praktischen Gesichtspunkten aufgestellte, vom aktuellen Stand der wissenschaftlichen Erkenntnis abhängige Regeln für die Durchführung von medizinischen Behandlungen oder klinischen Tests.

## **1.2 Software für klinische Studien**

Keine klinische (insbesondere multizentrische) Studie läuft heutzutage ohne EDV Unterstützung. Die im vorigen Abschnitt erwähnten Aufgaben eines Studienzentrums wie Studiendesign, Formulardesign, Patientenregistrierung, Datenerfassung, Monitoring sowie die statistische Auswertung sollten großteils mit Hilfe der dafür entwickelten Studiensystemen realisiert werden. Die große Komplexität dieser Aufgaben bedarf aufwendiger Implementierungen. Ideal ist es, wenn eine Software die gesamte Planung, Durchführung und Auswertung einer Studie umfasst.

Für die Erstellung der Datenerhebungsbögen (CRFs) werden heute meist Textverarbeitungssysteme wie z.B. MS-Word verwendet (1).

Für die Studiendatenbank wird eines der heute gängigen Datenbanksysteme verwendet. Für kleinere Studien mag es häufig MS Access sein, für größere Studien Oracle. Die Erstellung der Sicherheitskopien und die Archivierung der abgeschlossenen Studiendatenbanken erfolgt mit Komponenten des Datenbanksystems. Im *Trial Master File*, welches alle in einer Studie entstehenden Dokumente speichert, ist darauf zu achten, dass die Datenträger auch noch nach Jahrzehnten lesbar sind (1).

Für die Datenerfassung sind entweder spezielle Datenerfassungsprogramme notwendig oder es werden Komponenten zur Datenerfassung des Datenbanksystems verwendet. Zunehmend gewinnt auch *Remote Data Entry* an Bedeutung, d.h. die Direkteingabe der Daten in den Computer in der Klinik oder in der Praxis (1). Bisher wurden die Daten zunächst auf Papier dokumentiert und anschließend in das Computersystem übertragen.

Software für Plausibilitätskontrollen wird meist in die Programme zur manuellen Datenerfassung eingebaut. In den Datenerfassungsmodulen und Bildschirmmaskengeneratoren der Datenbanksysteme sind einfache Plausibilitätskontrollen zunehmend enthalten (1).

Die bisher genannten Softwareprodukte (MS Access, MS Word, etc.) wurden für verschiedenste Zwecke entwickelt, nicht speziell für klinische Studien. Zunehmend werden aber Softwaresysteme auf den Markt gebracht, die das gesamte Data Management einer klinischen Studie integrieren. Das reicht vom Entwurf der Erhebungsbögen und der Datenerfassungsmasken, von der Struktur der Studiendatenbank über Queries bis hin zur Datenkorrektur. Die Idee solcher integrierter Systeme ist, dass die Struktur einer Studie nur einmal für das System erfasst werden muss und dann Erhebungsbögen, Datenerfassungsmasken, Plausibilitätskontrollen, Queries und Datenkorrekturen diese Strukturinformation benutzen und damit wesentlich einfacher werden. Der Vorteil der integrierten Systeme ist die weitgehende automatisierte Dokumentation aller Studienvorgänge z.B. wann ein Dokument angelegt, verändert oder gelöscht wird und damit eine gute Nachvollziehbarkeit des Monitorings und des Data Managements (1).

### **1.3 Motivation**

Das Ziel dieser Arbeit ist die Planung und Realisierung eines Informationssystems zur Unterstützung klinischer Multicenterstudien (MCS). Unter einem solchen

Informationssystem versteht man ein integriertes Softwaresystem, welches Klinikern und Datenmanagern bei der Erstellung von Studienformularen, Formulierung von Plausibilitätsprüfungen und bei der Datenkorrektur Unterstützung leisten soll. Weiters soll eine Dokumentation der Studiendaten über Webtechnologien ermöglicht werden. Durch die webbasierte Dokumentation soll Klinikern die Möglichkeit geboten werden eine Studie national oder international in einer Vielzahl von Institutionen, also multizentrisch, durchzuführen.

Das am AKH Wien verwendete Studiensystem Archimed ist eine Eigenentwicklung des Instituts für Medizinische Informations- und Auswertesysteme. Es erfüllt die an ein Studiensystem gestellten Anforderungen, erlaubt jedoch bisher nicht, MCS durchzuführen (siehe Kapitel Systemanalyse).

Auf Wunsch der klinischen Nutzer sollen nun mit dem System Archimed zukünftig auch MCS abdeckbar sein. Weiters sollen bereits vorhandene Studien über das Web zugänglich gemacht werden können. Die im System Archimed gespeicherten Daten sollten dadurch für neue Studien retroaktiv, also im Nachhinein, zugreifbar gemacht werden.

## 2 Stand der Forschung

---

### 2.1 Vom CRF zum eCRF

In den vergangenen Jahren konnte bei klinischen Studien ein starker Trend weg von den papierbasierten Studienformularen, den *CRFs* (Case Report Forms), hin zu der elektronischen Direkteingabe der Patienteninformationen am PC dem *Direct Data Capturing* beobachtet werden. Die für diesen Zweck entworfenen *eCRFs* (electronic Case Report Forms) oder elektronische Studienformulare verbessern die Datenqualität und erleichtern die Eingabe. Die Verwendung solcher *eCRFs* im Internet ermöglicht die Nutzung von *eCRFs* für MCS. Die in *HTML* Hypertext Markup Language entwickelten *eCRFs* werden im Internet publiziert. Das mit Daten befüllte *eCRF* (Dokument) kann über Email oder *CGI* (Common Gateway Interface) Script direkt in die Studiendatenbank abgespeichert werden. Ebenso können die erhobenen Daten erneut zum Zweck des Editierens der Daten in ein *eCRF* geladen werden. Dieses Vorgehen erhöht die Effizienz bei MCS enorm, da falls entdeckt wird, dass Fehler bei der Eingabe passiert sind, das papierbasierte Dokument nicht über den Postweg zurückgeschickt werden muss sondern das elektronische Dokument über das Internet editiert werden kann (11-13).

Um den Klinikern am AKH Wien die best mögliche IT-Infrastruktur zu bieten, ist es dem Institut für Medizinische Informations- und Auswertesysteme ein Anliegen dem Trend der Durchführung von MCS über das Internet zu folgen. Es soll demzufolge ein Studiensystem geschaffen werden das den Klinikern ermöglicht *eCRFs* zu erstellen und diese im Internet zu publizieren.

### 2.2 Architekturansätze

Zur Realisierung von *eCRFs* und zur Bereitstellung dieser über das Internet bietet sich *HTML* in Kombination mit *JavaScript* an. *JavaScript* übernimmt die clientseitigen also im Browser ablaufenden Konsistenz- und Plausibilitätsprüfungen der Dateneingabe, und *HTML* dient zur Datenübermittlung und zur Darstellung im Browser. Bei einer *HTML* basierten Datenübertragung bedarf es eines *CGI* Scripts,

welches die im Dokument (ausgefülltes eCRF) erfassten Daten serverseitig verarbeitet und in die Studiendatenbank speichert. Eine HTML JavaScript/CGI basierte eCRF Entwicklung muss also mit unterschiedlichen Komponenten gebildet werden. Zum einen müssen die eCRFs in HTML und JavaScript geschrieben werden und zum anderen muss ein CGI entwickelt werden, welches das HTML Dokument verarbeitet und in die Studiendatenbank speichert. Daten aus der Studiendatenbank müssen in HTML und JavaScript eingebettet werden, um sie an den Browser zu übermitteln. Änderungen am HTML basierten eCRFs bedeuten Programmieraufwand an den serverseitigen CGI Scripts, da sie eine Änderung des Datenmodells zur Folge haben. Auch ist zu beachten, dass HTML limitierte *GUI* (Graphical User Interface) (grafische Benutzerschnittstelle) Eigenschaften aufweist (14, 15). Ein Ansatz, die Problematiken die sich durch die HTML/CGI Variante ergeben zu umgehen, besteht darin eine Client/Server Applikation zu entwickeln (15). Hier wird eine in Java2 geschriebene Applikation am Client installiert, die mit der zentralen Studiendatenbank die Daten austauscht. Hier ist auf der Client Applikation das eCRF implementiert. Zugleich wurde in (15) noch eine Variante unter der Verwendung von Java2 Applets implementiert, die gegenüber der Client Applikation einige Vorteile besitzt: Das Java2 Applet kann unter der Voraussetzung, dass eine Java2 Virtual Maschine installiert ist, in einem Web Browser geöffnet werden. Es muss nicht umständlich eine Applikation installiert werden über die man das eCRF mit Patienteninformationen befüllen kann, sondern das Applet wird zentral zur Verfügung gestellt. Bei Änderungen am eCRF muss somit keine Neuinstallation am Client erfolgen. Der Ansatz der Java Applets wurde auch in (14) vorgestellt.

Sowohl der Ansatz der Clientapplikation, als auch der Ansatz der Nutzung der Applet Technologie basieren auf der Java2 Technologie. Beide Ansätze profitieren von der in der Java2 Technologie zur Anwendung kommenden objektorientierten Programmierung. Die Entwicklung eines *Multicenterstudiensystems* (MCSS) am AKH Wien sollte ebenfalls auf einer objektorientierten Modellierung sowie der daraus folgenden objektorientierten Programmierung basieren. Durch eine objektorientierte Modellierung lassen sich komplexe Sachverhalte gut abstrahieren, und eine objektorientierte Programmierung ermöglicht eine leichtere Wartung des Systems. Mehr zu Java2 und zum objektorientierter Paradigma folgt in Kapitel 4.

## **2.3 FDA Anforderungen an ein Studiensystem**

Die FDA (Food and Drug Administration) hat eine Richtlinie herausgegeben, anhand derer elektronische Studien-Aufzeichnungen als äquivalent zu traditionellen handgeschriebenen Dokumenten betrachtet werden dürfen (16). Basierend auf diesen Anforderungen kann man ein Studiensystem zertifizieren lassen. Durch eine Zertifizierung kann die Einhaltung dieser Standards sichergestellt werden, wodurch die Qualität und die Glaubwürdigkeit eines Produktes erhöht werden. Ich möchte nun in diesem Abschnitt die Richtlinie vorstellen.

### **FDA Guidance for Industry, Computerized Systems used in Clinical Trials**

#### **1. Allgemeine Vorgehensweisen**

- 1.1. Wenn ein Computersystem benutzt wird um Daten zu erzeugen, zu editieren, zu speichern, zu archivieren, auszuwerten oder zu übermitteln sollte das im Studienprotokoll festgehalten werden.
- 1.2. In jeder Studie sollte dokumentiert werden welche Software, und wenn bekannt welche Hardware verwendet wird, falls Daten computerunterstützt erzeugt, editiert, gespeichert, archiviert, ausgewertet oder übermittelt werden.
- 1.3. Quelldokumente sollten aufbewahrt werden, um eine Rekonstruktion zu ermöglichen.
- 1.4. Falls die Originalbeobachtungen direkt in das Computersystem eingegeben werden, so entspricht diese Aufzeichnung dem Quelldokument.
- 1.5. Das Design eines Computersystems sollte sicherstellen, dass in einer Studie alle notwendigen regulatorischen Anforderungen bezüglich der Datensatzspeicherung im gleichen Ausmaß berücksichtigt werden wie bei einer Papierdokumentation.
- 1.6. Kliniker sollten entweder die Original-Dokumentation oder eine zertifizierte Kopie aller Quelldokumente aufbewahren, die an den Sponsor oder die Vertragsforschungsorganisation übermittelt worden ist. Weiters sollte ein Abkommen bestehen, welches dem Kliniker ermöglicht, Anfragen bezüglich der übermittelten Dokumente zu stellen.
- 1.7. Jede persistente Änderung an einem Datensatz darf die originale Information nicht verschleiern oder löschen. Der Datensatz sollte deutlich

- anzeigen, dass eine Veränderung stattgefunden hat und ein deutliches Mittel zur Verfügung stellen die vorherige Information zu lesen und zu lokalisieren.
- 1.8. Änderungen an elektronisch gespeicherten Daten verlangen immer einen Audit Trail, wie in 21 CFR 11.10(e) (17) gefordert. Diese Dokumentation sollte beinhalten, wer die Änderungen vorgenommen hat, wann und warum sie gemacht worden sind.
  - 1.9. Die FDA kann jeden Datensatz inspizieren, unabhängig davon, wie diese gespeichert aufbewahrt wird.
  - 1.10. Daten sollten so auswertbar/abrufbar sein, dass alle Informationen eines individuellen Subjekts diesem zurechenbar sind. (bzw. über dieses zugänglich sind.)
  - 1.11. Computersysteme sollten so entwickelt werden, dass 1. alle im Studienprotokoll an das System gestellten Anforderungen erfüllt sind und 2. alle Fehler in der Datenerzeugung, -änderung, -archivierung, -abfrage, oder -Übermittlung ausgeschlossen sind.
  - 1.12. Sicherheitsmechanismen sollten eingesetzt werden, um unauthorisierten Zugang zu den Daten und dem Computersystem zu vermeiden.
2. Standardarbeitsanweisungen (Standard Operating Procedures) sollten errichtet werden für:
- 2.1. System Setup und Installation
  - 2.2. Datenerfassung und Bedienung
  - 2.3. Systemwartung
  - 2.4. Daten Backup, Recovery, und Ausfallpläne
  - 2.5. Sicherheit
  - 2.6. Änderungskontrolle
3. Dateneingabe
- 3.1. Elektronische Signaturen
    - 3.1.1. Um sicherzustellen, dass Personen eine ausreichende Berechtigung besitzen, um Daten zu erfassen, sollte das Dokumentationssystem zu Beginn der Sitzung, eine Authentifizierung in Form einer elektronischen Signatur verlangen. Beispiele hierfür wären etwa Benutzername/Passwort, Codes, Biometrische Daten.

- 3.1.2. Das Datenerfassungssystem sollte so entwickelt sein, dass die einzelnen Einträge attributierbar sind. Jeder Eintrag bzw. Änderungen in/an einen elektronischen Datensatz sollte ausschließlich unter der Signatur jener Person gemacht werden, die gerade den Eintrag oder die Änderung vornimmt. Das bedeutet natürlich nicht, dass für jeden Eintrag bzw. jede Änderung eine Authentifizierung notwendig ist. Eine einzige Authentifizierung (z.B. Login) ermöglicht mehrfaches Eintragen oder Ändern von Datensätzen.
    - 3.1.2.1. Der Name derjenigen Person, die die Daten eingibt, sollte während der gesamten Sitzung am Eingabeschirm angezeigt werden. Damit soll vermieden werden, dass unbeabsichtigter Weise eine Person unter falschem Namen Daten eingibt.
    - 3.1.2.2. Falls der Name am Eingabeschirm nicht dem der Person, die gerade davor sitzt, entspricht, sollte diese bevor sie mit der Datenerfassung fortfährt sich unter ihrem eigenem Namen einloggen.
  - 3.1.3. Jede Person sollte ihren eigenen Zugang (Passwort, etc.) besitzen und diesen nicht mit anderen gemeinsam benutzen. Eine Person sollte sich nicht für eine andere Person (die keinen Zugang hat) einloggen.
  - 3.1.4. Passwörter bzw. andere Zugangsdaten sollten nach regelmäßigen Intervallen geändert werden.
  - 3.1.5. Wenn eine Person ihren Arbeitsplatz verlässt, sollte sie sich vom System abmelden. Vergisst eine Person sich abzumelden, sollte nach einer angemessenen Inaktiv-Zeit (Idle Time), ein automatisches Logout durch das System vorgenommen werden. Für kurze Inaktiv-Zeiten sollte ein Mechanismus vorgesehen werden, der vor unauthorisierter Dateneingabe schützt, wie z.B. Bildschirmschoner mit Passwortabfrage
- 3.2. Audit Trails
- 3.2.1. Absatz 21 CFR 11.10(e) (17) verlangt von Personen, die ein elektronisches Datenerfassungssystem benutzen, einen Audit Trail zu führen, also ein Instrument um Authentizität, Integrität und die Vertraulichkeit der elektronischen Aufzeichnung zu gewährleisten.
    - 3.2.1.1. Beim Ändern, Erzeugen, Löschen oder sonstigen operativen Einträgen müssen Personen eine unabhängige Aufzeichnung mit

einem sicheren, computergenerierten Zeitstempel führen. Ein Eintrag gilt als erzeugt, wenn dieser auf ein dauerhaftes Medium gespeichert wurde.

3.2.1.2. Audit Trails müssen mindestens gleichlange aufbewahrt werden, wie das für den Datensatz notwendig ist zu dem diese gehören.

3.2.2. Personen die Datensätze erzeugen, verändern oder modifizieren können, sollten Audit Trails nicht verändern dürfen.

3.2.3. Kliniker sollten entweder die originalen Audit Trails oder zertifizierte Kopien aufbewahren.

3.2.4. FDA Personal sollten in der Lage sein Audit Trails, sowohl auf der Studienseite, als auch dort, wo die Daten gespeichert werden einsehen zu können.

3.2.5. Audit Trails sollten inkrementell und in chronologischer Reihenfolge erzeugt werden. Weiters sollte sichergestellt sein, dass neu erzeugte Audit Trail Information existierende nicht überschreibt §11.10(e).

### 3.3. Date/Time Stamps

3.3.1. Es sollten Kontrollen vorhanden sein um sicherzustellen, dass die Systemzeit / -datum korrekt sind.

3.3.2. Es sollte einem beschränkten und autorisierten Personal vorbehalten sein Änderungen an Datum und Zeit vorzunehmen, und dieses Personal sollte benachrichtigt werden, falls Abweichungen bei Datums bzw. Zeitangaben auftreten. Änderungen sollten dokumentiert werden.

3.3.3. Datum- und Zeitangaben sollten lokal sein zu dem Ort, an dem dokumentiert wird und sollten Jahr, Monat, Tag, Stunde, und Minute umfassen.

3.3.4. Bei klinischen Studiensystemen für MCS erfolgt die Berechnung des lokalen Zeitstempels über einen Remoteserver, der möglicherweise in einer anderen Zeitzone steht.

## 4. Systemeigenschaften

4.1. Systeme, die für eine direkte Dateneingabe benutzt werden, sollten Eigenschaften bereitstellen, die die Datenerfassung erleichtern und die Datenqualität erhöhen.

4.1.1. Warnungen, Flags oder andere Hilfsmittel sollen dem Benutzer dabei helfen klinische Terminologien konsistent zu benutzen, und den

Benutzer warnen, wenn die eingegebenen Daten außerhalb des Wertebereichs liegen. Automatismen, die automatisch Felder befüllen, wenn diese übergangen werden, sollte man nicht benutzen.

- 4.1.2. e-CRFs sollten so designt sein dass sie Benutzern ermöglichen Kommentare einzutragen. Kommentare erhöhen durch ad hoc Information die Datenqualität. Diese Informationen könnten hilfreich sein im Zusammenhang mit Abstoßreaktionen oder unerwarteten Resultaten. Der Datensatz sollte deutlich anzeigen, wer den Kommentar hinzugefügt hat und wann (Datum, Uhrzeit).
- 4.2. Systeme für direkte Dateneingabe sollten so entwickelt sein, dass Funktionalitäten zur Verfügung stehen, die das Kontrollieren und Nachprüfen der Eingabe unterstützen. Daten-Tags (wie unterschiedliche Farbe, -Font, Flags) sollten benutzt werden, um zu verdeutlichen welche Daten geändert oder gelöscht wurden wie im Audit Trail dokumentiert.
- 4.3. Abfrage / Auswertung von Daten
  - 4.3.1. Dem Sponsor muss die Möglichkeit geboten werden, selbst wenn das Computersystem durch ein neueres ersetzt wird oder nicht mehr weiter betreut wird, die Daten, die im alten System erfasst worden sind, zu begutachten, zu kontrollieren und abzufragen. Diese Forderung kann dadurch erfüllt werden, dass man entweder das alte System weiter betreut oder die Daten in das neue System überführt.
  - 4.3.2. Bei der Migration der Daten in das neue System muss man darauf achten, dass genaue und vollständige Kopien der Studiendaten sowie alle Zusatzinformationen, die notwendig sind, um die Datenintegrität zu gewährleisten in das neue System übergeführt werden. Diese Zusatzinformationen umfassen z.B. Audit Trails. Jede Datenabfragesoftware, jedes Script, oder Abfragelogik die dazu benutzt wird um Daten für eine Auswertung zu manipulieren, abzufragen oder Daten zu extrahieren, sollte dokumentiert und für die Lebensdauer der Auswertung aufbewahrt werden.
- 4.4. Rekonstruktion von Studien: Die FDA erwartet sich in der Lage zu sein eine Studie rekonstruieren zu können. Das bezieht sich nicht nur auf die Daten sondern auch darauf, wie die Daten zustande gekommen sind und wie sie gehandhabt werden/wurden. Alle einsetzbaren Softwareversionen,

Betriebssysteme, und Softwareentwicklungsumgebungen, die an der Verarbeitung von Datensätzen beteiligt sind/waren, sollten solange erreichbar sein solange diese Datensätze aufbewahrt werden müssen. Sponsoren behalten sich möglicherweise diese Versionen selbst auf oder erwerben die Möglichkeit diese selbst auszuführen. Obwohl die FDA von Herstellern und Sponsoren erwartet ältere Versionen ausführen zu können, erkennt die FDA, dass es in einigen Fällen für Hersteller und Sponsoren nicht immer möglich sein wird ältere Versionen auszuführen.

## 5. Sicherheit

### 5.1. Physische Sicherheit

- 5.1.1. Zusätzlich zu internen Sicherheitsmechanismen, die in das System integriert sind, sollten externe Sicherheitsmechanismen umgesetzt werden, die sicherstellen, dass der Zugang zu computerisierten Systemen und den Daten auf autorisierte Personen beschränkt ist.
- 5.1.2. Der sorgsame Umgang mit Systemsicherheitsmaßnahmen und mit den limitierten Zugängen für autorisiertes Personal sollte den Mitarbeitern bewusst sein.
- 5.1.3. Standardarbeitsanweisungen (SOPs) für den Umgang und die Speicherung des Systems sollten etabliert werden, um den unautorisierten Zugang zu vermeiden.

### 5.2. Logische Sicherheit

- 5.2.1. Der Zugang zu klinischen Daten sollte eingeschränkt und durch Systemsoftware mit erforderlichen Log-on, Sicherheitsmechanismen, und Audit Trails überwacht werden. Daten sollten keines falls durch externe Software, die nicht den Sicherheitsmechanismen unterliegt, änderbar, ansehbar, abfragbar oder auswertbar werden.
- 5.2.2. Es sollte ein Datensatz existieren, der zu jedem Zeitpunkt die gesamten Namen der autorisierten Personen, deren Titel und deren Zugriffsrechte beinhaltet. Dieser Datensatz sollte über die Studiendokumentation erreichbar sein.
- 5.2.3. Falls ein Sponsor ein Computersystem ausschließlich für klinische Studien anschafft, muss darauf geachtet werden, dass das System dem Zweck für den es bestimmt und validiert worden ist, zugeteilt bleibt.

- 5.2.4. Falls ein Computersystem, welches normalerweise für andere Zwecke in Verwendung ist, für eine klinische Studie benutzt wird, muss dafür Sorge getragen werden, dass die Teile des Computersystems, die für die Studie benutzt werden, logisch und physisch isoliert sind, damit unbeabsichtigte Interaktion mit Nicht-Studiensoftware vorgebeugt wird. Falls eines der Softwareprogramme geändert wird, sollte das System dorthingehend evaluiert werden ob diese Änderungen Einfluss auf die logische Sicherheit haben.
- 5.2.5. Es sollten Kontrollmechanismen eingesetzt werden deren Ziel die Vorbeugung, Auffindung, sowie Linderung der Effekte von Computerviren auf Studiendaten und Software sind.
6. Systemstabilität und –zuverlässigkeit. Der Sponsor sollte sicherstellen und dokumentieren, dass das verwendete Computersystem den vom Sponsor gestellten Anforderungen wie Vollständigkeit, Genauigkeit, Zuverlässigkeit und Konsistenz entspricht.
  - 6.1. Systemdokumentation sollte an der Stelle zugänglich sein, von der aus die klinische Studie geleitet wird. Diese Dokumentation sollte eine allumfassende Beschreibung des Computersystems, der Beziehungen mit der Hardware, mit der Software und mit der physischen Umgebung darstellen.
  - 6.2. Die FDA kann diese Dokumentation um die Validierung der Software zu beweisen, durch ein reguliertes Unternehmen inspizieren lassen. Die Verantwortung solche Dokumentation zum Zeitpunkt der Inspektion an der Stelle wo die Software verwendet wird zugänglich zu machen, liegt beim Studiensponsor. Kliniker sind solange sie die Software nicht entworfen oder verändert haben für die Validierung nicht verantwortlich.
    - 6.2.1. Der Großteil der Validierung für Standardanwendungen sollte bereits von dem Unternehmen, das die Software geschrieben hat, vorgenommen worden sein. Der Sponsor oder die Vertragsforschungsorganisation sollte über Dokumentation der Validierung durch den Hersteller verfügen und sollte selber Funktionstest durchgeführt haben und Softwarelimitierungen, Probleme, und Fehlerkorrekturen erforscht haben. Der Sponsor oder die Vertragsforschungsorganisation hat

möglicherweise in Spezialfällen in denen Datenbanksoftware oder Tabellensoftware (1) als Standardsoftware, (2) als Software entwickelt für großteils allgemeinen Nutzen, (3) unverändert und (4) nicht für direkte Dateneingabe verwendet wird, keine Validierung. In diesen Fällen sollte der Sponsor oder die Vertragsforschungsorganisation selber Funktionstests durchgeführt haben, und Softwarelimitierungen, Probleme, und Fehlerkorrekturen erforscht haben.

6.2.2. Dokumentation, die Softwarevalidierung belegt, sollte folgendes beinhalten:

6.2.2.1. Dokumentierte Designspezifikation die beschreibt was und wie die Software arbeitet.

6.2.2.2. Ein schriftlich festgehaltener Testplan basierend auf der Designspezifikation, der sowohl strukturelle als auch funktionelle Analyse beinhaltet.

6.2.2.3. Test-Ergebnisse und eine Evaluierung, die belegt, dass diese Ergebnisse mit der Spezifikation übereinstimmen.

6.3. Änderungsüberwachung

6.3.1. Es sollten Anweisungen dokumentiert werden, die sicherstellen, dass bei Änderungen wie Softwareupdates, oder der Ersatz von Komponenten oder Instrumenten am Computersystem die Integrität der Daten und die Integrität des Protokolls erhalten bleiben.

6.3.2. Die Auswirkung einer Änderung am Computersystem sollte evaluiert werden und eine Entscheidung gefällt werden, ob eine Revalidierung notwendig ist. Eine Revalidierung sollte dann durchgeführt werden, wenn Änderungen vorgenommen wurden, die das Einsatzgebiet oder die Designspezifikation überschreiten.

6.3.3. Alle Änderungen an einem Computersystem sollten dokumentiert werden.

7. Systemüberwachung

7.1. Softwareversionskontrolle

Maßnahmen um sicherzustellen, dass die Softwareversionen zu Erzeugung, Sammlung, Speicherung und Übermittlung von Daten mit den Versionen in der Dokumentation übereinstimmen, sollten etabliert werden.

## 7.2. Ausweichpläne

Im Falle eines Ausfalls oder Versagens des Computersystems sollten schriftlich festgehaltene Vorgehensweisen bereitstehen, die die Ausweichpläne beschreiben, um ein alternatives Fortsetzen der Studie zu ermöglichen.

## 7.3. Backup und Wiederherstellung elektronischer Datensätze

7.3.1. Um Datenverlust zu vermeiden, sollten Backup- und Wiederherstellungsanweisungen klar in den Standardanweisungen SOPs formuliert sein. Daten sollten regelmäßig gebackupt werden, so dass ein katastrophaler Datenverlust vermieden wird und die Datenqualität und –Integrität gewährleistet ist.

7.3.2. Backup Datensätze sollten an einer sicheren Stelle gespeichert werden, die in den SOPs spezifiziert ist. Diese Speicherung ist typischerweise an einer anderen Stelle als die Originale.

7.3.3. Backup- und Wiederherstellungs-Logs sollten aufbewahrt werden, um das Ausmaß eines Datenverlustes auf Grund eines Systemversagens feststellen zu können.

## 8. Personalschulung

### 8.1. Qualifikationen

8.1.1. Jede Person, die Daten eingibt oder verarbeitet, sollte Wissen, Schulung und Erfahrung oder jede andere dafür notwendige Voraussetzung haben, um die ihr zugewiesene Funktion auszuüben.

8.1.2. Jede Person, die die Studie kontrolliert und überwacht, sollte Wissen, Schulung und Erfahrung haben die notwendig ist, um eine Studie adäquat zu kontrollieren und zu überwachen.

### 8.2. Schulungen

8.2.1. Die Personen, die spezielle Operationen ausführen werden, sollen dementsprechend geschult sein.

8.2.2. Schulungen sollten wenn nötig regelmäßig von qualifizierten Personen geleitet werden, um sicherzustellen, dass die Benutzer mit dem System und Erneuerung am System während der Studiendauer umgehen können.

### 8.3. Dokumentation

Mitarbeiterwissen, -schulungen und -erfahrungen sollten dokumentiert werden.

## 9. Datenbegutachtung

9.1. Die FDA kann alle für eine Begutachtung notwendigen Datensätze einsehen, gleichgültig wie sie erzeugt oder aufbewahrt wurden. Deshalb sollten Systeme in der Lage sein sowohl menschenlesbare als auch maschinenlesbare, präzise und vollständige Kopien der Datensätze anzulegen, die für eine Inspektion, eine Begutachtung und eine Kopie für die Agency notwendig sind. Die Personen sollten die Agency kontaktieren falls es Zweifel gibt welche Datenformate und Medien die Agency lesen und kopieren kann.

9.2. Der Sponsor sollte, falls notwendig in der Lage sein, an der Stelle an der die Inspektion stattfindet, dem FDA Personal, welches die elektronische Dokumentation sowie die Audit Trails inspiziert, Software und Hardware zur Verfügung zu stellen.

10. Zertifizierung elektronischer Signaturen: Personen die elektronische Signaturen verwenden, die nach dem 20. August 1997 benutzt werden, müssen gemäß 21 CFR 11.100(c) (17) der FDA vorher oder zum Zeitpunkt der Verwendung zertifizieren, dass diese elektronischen Signaturen mit einer traditionellen handgeschriebenen Unterschrift gleichzusetzen sind. Des Weiteren, wie in 21 CFR 11.100(c) (17) verlangt soll diese Zertifizierung in Papierform unterschrieben mit einer traditionellen handgeschriebenen Unterschrift im Office of Regional Operations(HFC-100), 5600 Fishers Lane, Rockville Maryland 20857, eingereicht werden. Diese Zertifizierung muss bevor oder zum Zeitpunkt an dem die elektronischen Signaturen zum Einsatz kommen, übermittelt werden. Eine einzige Zertifizierung beglaubigt alle elektronischen Signaturen einer dezidierten Organisation. Diese Zertifizierung ist ein amtliches Dokument, welches anerkennt, dass elektronische Signaturen die gleiche rechtliche Bedeutung haben wie traditionell handgeschriebene Unterschriften. Eine akzeptable Zertifizierung kann folgende Form haben (original Text):  
"Pursuant to Section 11.100 of Title 21 of the Code of Federal Regulations, this is to certify that [name of organization] intends that all electronic

signatures executed by our employees, agents, or representatives, located anywhere in the world, are the legally binding equivalent of traditional handwritten signatures."

## **3 Methoden**

---

Die in diesem Kapitel verwendeten Methoden sind aus dem Buch „IT-Projektmanagement im Krankenhaus und Gesundheitswesen“ (18) abgeleitet. Aus den generisch gehaltenen Methoden aus den Kapiteln 8 und 9 im Buch werde ich die Methoden zur Systemanalyse und zur Systemspezifikation einsetzen.

Die hier vorgestellten Methoden dienen der Analyse des IST-Zustandes sowie der Gewinnung des SOLL-Zustandes und der Systemauswahl.

### **3.1 Methoden der Systemanalyse**

Gliederung des Analyse Plan wie im Buch „IT-Projektmanagement im Krankenhaus und Gesundheitswesen“ von Ammenwerth und Haux (18), Kapitel 8.2.

#### **1. Einleitung**

##### **1.1. Hintergrund und Zielsetzung der Systemanalyse**

In diesem Abschnitt werden der Hintergrund und die Zielsetzung unter Bezugnahme auf das übergeordnete Projektziel konkretisiert und der zu analysierende Problembereich genauer definiert.

##### **1.2. Genaue Fragestellung der Systemanalyse**

In diesem Abschnitt wird die Fragestellung der Systemanalyse vorgestellt. Anhand der Beantwortung dieser Fragestellung wird nach der Analyse die Bewertung durchgeführt.

#### **2. Untersuchungsplan**

Der Untersuchungsplan beschreibt auf welche Weise, wann, wo und was untersucht wird.

##### **2.1. Überblick: Allgemeines Studiendesign**

Dieser Abschnitt beinhaltet das allgemeine Vorgehen während der Untersuchung.

##### **2.2. Ort und Zeitraum der Studie**

In diesem Abschnitt werden der Ort und die Dauer der Analyse definiert.

##### **2.3. Untersucher Bereich**

Hier wird definiert welcher Bereich zu untersuchen ist. Zu untersuchende Bereiche sind z. B. Organisationseinheiten, Anwendungssysteme und Dokumente.

#### 2.4. Abgrenzung der Studie

An dieser Stelle soll eine Eingrenzung des zu analysierenden Bereiches stattfinden. Darunter versteht man z. B. Anwendungssysteme die nicht Inhalt der Analyse und Bewertung sind.

#### 2.5. Planung der Informationsbeschaffung

In diesem Abschnitt findet die Planung der Informationsbeschaffung statt. Darunter versteht man z.B. Expertenbefragungen, Literaturanalysen.

#### 2.6. Planung der Modellierung bzw. Datenauswertung

An dieser Stelle werden die Formen der Darstellung der Informationen, die während der Analyse erfasst wurden und zur Datenauswertung verwendet werden erläutert.

#### 2.7. Planung der Systembewertung

In diesem Abschnitt wird dargestellt auf welche Weise die Systembewertung durchgeführt wird.

### 3. Analyse und Bewertung

#### 3.1. Analyse

Dieser Abschnitt beinhaltet die Systemanalyse.

#### 3.2. Bewertung

Dieser Abschnitt beinhaltet die Systembewertung.

### 4. Analysebericht

Der Analysebericht fasst nocheinmal die Ergebnisse der Systemanalyse zusammen.

## **3.2 Methoden der Systemspezifikation**

Erstellung des Pflichtenheftes wie im Buch „IT-Projektmanagement im Krankenhaus und Gesundheitswesen“ von Ammenwerth und Haux (18), Kapitel 9.3.

#### 1. Ausgangssituation

Dieser Abschnitt soll einen Überblick über den IST-Zustand des Problembereichs geben.

- 1.1. Übersicht über das Informationssystem  
In diesem Abschnitt wird das neu zu spezifizierende Informationssystem dargestellt.
- 1.2. Ergebnisse der Systemanalyse  
Hier werden die Ergebnisse der Systemanalyse, also des IST-Zustandes dargestellt.
- 1.3. Ergebnisse der Systembewertung  
In diesem Abschnitt werden die Ergebnisse der Systembewertung dargestellt.
- 1.4. Gewinnung von Anforderungen durch Betrachtung kommerzieller Systeme  
In diesem Abschnitt werden kommerzielle Systeme betrachtet und analysiert. Aus der Analyse werden Anforderungen an das zu spezifizierende System extrahiert.
- 1.5. Weitere relevante Randbedingungen  
In diesem Abschnitt befinden sich Randbedingungen die Einfluss auf die Systemspezifikation haben.
2. Zieldefinition und Abgrenzung  
In diesem Abschnitt werden das Ziel und die Abgrenzung der Systemspezifikation formuliert.
3. Produkteinsatz und Produktumgebung  
Dieser Abschnitt soll einen Überblick über das Produkt, die Einbettung in Geschäftsprozesse und in die technische Produktumgebung geben.
4. Anforderungen  
Dieser Abschnitt und seine Unterkapitel beschreiben die Anforderungen an das System. Unter funktionalen Anforderungen versteht man jene Anforderungen die aus Sicht des Benutzers die Funktionalität des Systems beschreiben. Unter nichtfunktionalen Anforderungen versteht man jene Anforderungen die das System zusätzlich noch erfüllen muss.
  - 4.1. Nichtfunktionale Anforderungen
  - 4.2. Funktionale Anforderungen
5. Weitere Vorgaben  
Dieser Abschnitt behandelt Vorgaben die sich aufgrund der Spezifikation für einen erfolgreichen Betrieb des Systems notwendig sind.
  - 5.1. Einführung, Betrieb, Wartung

In diesem Abschnitt wird beschrieben worauf man bei der Einführung, beim Betrieb und bei der Wartung des Systems beachten muss.

### **3.3 Methoden der Systemauswahl und Eigenentwicklung**

Typische Aktivitäten des Abschnitts Systemauswahl wie im Buch „IT-Projektmanagement im Krankenhaus und Gesundheitswesen“ von Ammenwerth und Haux (18), Kapitel 10, vorgestellt sind:

#### 1. Abwägen zwischen Eigenentwicklung und einer kommerziellen Standardsoftware

Dieser Abschnitt beinhaltet die Überlegungen die zu der Entscheidung für oder wider eine kommerzielle Standardsoftware führen.

#### 2. Umsetzung (19)

Dieser Abschnitt beschreibt das Vorgehen und die Durchführung der Implementierung des Systems.

##### 2.1. Auswahl der Programmiersprache und der Entwicklungsumgebung

Hier wird zwischen den zur Verfügung stehenden Programmiersprachen und Entwicklungsumgebungen verglichen und eine Auswahl getroffen.

##### 2.2. Namenskonventionen und Codegestaltung

In diesem Abschnitt werden für die Entwicklung notwendige Namenskonventionen und die Codegestaltung definiert.

##### 2.3. Integrationsplanung

Hier wird definiert wie bei der Entwicklung der einzelnen Komponenten vorgegangen wird.

##### 2.4. Implementierung der Klassen

In diesem Abschnitt werden die implementierten Klassen vorgestellt.

##### 2.5. Implementierung der grafischen Benutzerschnittstelle

In diesem Abschnitt werden die implementierten Webseiten vorgestellt.

## 4 Ergebnisse

---

### 4.1 Ergebnisse der Systemanalyse

Ziel dieses Abschnitts ist es die in Kapitel 3.1 vorgestellte Gliederung abzuarbeiten um als Resultat die Systembewertung und den Analysebericht zu erhalten.

#### 4.1.1 Einleitung

##### Hintergrund und Zielsetzung der Systemanalyse

Seit vielen Jahren ist für Kliniken am AKH das System Archimed, unter der inhaltlichen Betreuung der Mitarbeiter des Instituts für Medizinische Informations- und Auswertesysteme, als Studiendokumentationssystem im Einsatz. Das im System verwendete Datenmodell basiert auf einem Entity Attribute Value (EAV) Datenmodell (20). Mehr zum EAV Datenmodell im siehe Abschnitt 4.1.3.1.3.

Das Ziel der Systemanalyse besteht darin herauszufinden, ob es sinnvoll ist, das bestehende Studiendokumentationssystem zu einem webbasierten MCSS zu erweitern oder dadurch zu ersetzen.

Die Mitarbeiter des Instituts für Medizinische Informations- und Auswertesysteme haben das System Archimed entworfen und implementiert. Diese stehen für den Analyseprozess zur Verfügung und ermöglichen tiefe Einblicke in die Software Systemarchitektur, sowie in die realisierten Funktionen des Systems.

##### Genauere Fragestellung der Systemanalyse

1. Kann man mit dem bestehenden Studiensystem MCS durchführen?
2. Kann man MCS webbasiert durchführen?
3. Ist das Datenmodell flexibel genug, um eine Webapplikation für MCS hinzuzufügen?
4. Welche Möglichkeiten zur Integration von Daten bietet das System Archimed?
5. Wie sieht die Hardwarekonfiguration des Servers aus?
6. Ist die Hardware leistungsfähig genug um eine Webapplikation zu betreiben?

7. Soll für die Realisierung eines webbasierten Informationssystems zur Unterstützung medizinischer MCS das bestehende Studiensystem erweitert oder durch ein neues ersetzt werden?

## **4.1.2 Untersuchungsplan**

### **Überblick: Allgemeines Studiendesign**

Das Studiendesign richtet sich danach, das System Archimed auf Systemarchitekturebene zu analysieren. Das Ziel ist es festzustellen ob es für MCS geeignet ist, und welche Technologie zur Erweiterung des Systems um eine Webkomponente zur Durchführung für MCS am besten scheint. Die Analyse wird ausschließlich am Institut für Medizinische Informations- und Auswertesysteme durchgeführt. Weiters ist eine mündliche Befragung der dort angestellten Experten geplant, die individuell je nach Problem- und Aufgabenbereich abgehalten wird. Anforderungen an ein webbasiertes MCSS werden von diesen Experten zusammengefasst und kommuniziert.

Die zu betrachtenden Bereiche sind:

1. Studiendokumentation
2. Systemarchitektur
3. Datenbankstruktur und Datenbank
4. Webtechnologien
5. Datenbankanbindung

Die Studie wird folgendermaßen durchgeführt:

1. Befragung der Systemexperten und Klinikbetreuer
2. Literaturanalyse

### **Ort und Zeitraum der Studie**

Die Analyse wird am Institut für Medizinische Informations- und Auswertesysteme an der Besonderen Einrichtung für Medizinische Statistik und Informatik durchgeführt und zwar an der Medizinischen Universität Wien.

Für die Analyse ist ein Zeitraum von 2 Wochen veranschlagt.

### **Untersuchter Bereich**

Das Institut für Medizinische Informations- und Auswertesysteme besteht aus 3 wissenschaftlichen Mitarbeitern (inkl. Leiter), 7 Klinikbetreuern und Systemexperten und 4 externen Klinikbetreuern und Systemexperten.

Die untersuchten Komponenten sind:

1. Studiendokumentation
2. Systemarchitektur
3. Datenbank und Datenbankstruktur
4. Datenbankanbindung

Die untersuchten Dokumente und befragten Personen sind:

1. Systemdokumentation Archimed im System
2. Literatur (20)
3. Literatur (21)
4. Personen des Instituts für Med. Informations- und Auswertesysteme

### **Abgrenzung der Studie**

Es ist nicht Teil der Studie eine umfassende Befragung von Klinikern durchzuführen, da diese bereits im Vorfeld durch individuelle Kontakte des Archimed-Entwicklerteams mit klinischen Nutzern erfolgte und die entsprechenden Informationen von diesen erhoben wurden.

Ebenso wenig ist es Teil der Studie, andere Applikationszweige außer der Studiendokumentation und dem Formulardesigner näher zu betrachten.

### **Planung der Informationsbeschaffung**

Erhoben werden die Antworten auf die Fragestellungen der Analyse in Einzelgesprächen mit den Systemexperten und Klinikbetreuern.

Weiters werden vorhandene Literaturreferenzen zur Durchführung einer Datenbestandsanalyse herangezogen.

### **Planung der Modellierung bzw. Datenauswertung**

Die Resultate der Systemanalyse werden in einem Analysebericht zusammengefasst.

### **Planung der Systembewertung**

Die Stark- und Schwachstellenanalyse ist ein Verfahren, um Stark- und Schwachstellen einer Komponente oder Situation systematisch darzustellen. Sie fasst dabei die Erkenntnisse einer Systembewertung in zwei Bereiche zusammen:

1. Starkstellen: Vorhandene Stärken der betrachteten Situation.
2. Schwachstellen: Vorhandene Schwächen der betrachteten Situation.

### **4.1.3 Analyse und Bewertung**

Dieser Abschnitt beschäftigt sich mit der Analyse und Bewertung des Systems Archimed. Die Analyse hat als Ziel die Fragestellung aus Abschnitt 4.1.1 zu beantworten, die Bewertung basiert auf dieser Beantwortung.

#### **4.1.3.1 Analyse**

Die Analyse wird zielgerichtet nach den Fragestellungen von Abschnitt 4.1.1 abgewickelt. Die zur Beantwortung der Frage notwendigen Ressourcen werden herangezogen und aufgelistet.

##### *4.1.3.1.1 Kann man mit dem bestehenden Studiensystem MCS durchführen?*

Befragte Personen: Archimed-Entwicklerteam

Die Durchführung von MCS ist prinzipiell möglich. Allerdings ergeben sich folgende organisatorische Schwierigkeiten:

1. Das Problem der Softwareverteilung muss gelöst werden. Der erste Ansatz besteht darin, die *Archimed Clientapplikationen* zu verteilen. Bei der Installation der Archimed Clientapplikation muss darauf geachtet werden, dass die notwendigen Registryeinträge im Betriebssystem vorgenommen werden. Diese Einträge sind notwendig, damit die Verbindung zur Datenbank hergestellt werden kann. Ein zweiter Ansatz besteht darin, dass der *Archimed Virtual Client* als Batch Datei verteilt wird. Dabei würden die Registryeinträge im Betriebssystem bei jedem Start der Applikation automatisch und temporär gesetzt werden.
2. Zugang zum Datennetz der Medizinischen Universität Wien muss auf sicherem Weg möglich sein. Eine Verschlüsselung beispielsweise über einen verschlüsselten VPN Tunnel oder *Oracle Security* wäre notwendig. Bei der Oracle Security würden die Kommunikationspakete zwischen der Clientapplikation und/oder dem Archimed Virtual Client und der

Studiendatenbank verschlüsselt werden. Oracle Security bedeutet eine Neuprogrammierung der Clientapplikation und/oder des Archimed Virtual Clients. Das Problem der Softwareverteilung könnte z. B. über CDs oder einen Softwareserver gelöst werden.

3. Der Zugang zum MUW Datennetz ist deshalb notwendig, da der Datenbankserver innerhalb dieses MUW Datennetzes (= Intranet) steht und dieses Intranet über eine Firewall nach außen hin abgesichert ist. Prinzipiell ist der Zugang von außerhalb in das MUW Datennetz gesperrt. Ein temporäres Öffnen der Ports in der Firewall für jeden PC, der einen Archimed (Virtual) Client ausführt mit ausschließlichem Zugang zum Datenbankserver, wäre denkbar aber organisatorisch enorm aufwändig, da ein temporäres Öffnen der Ports in der Firewall vom zentralen Informatikdienst (ITSC) streng überwacht werden müsste, um Missbrauch zu vermeiden.
4. Ein weiteres Hindernis stellt die Zuteilung eines Zugangs zum MUW Datennetz dar. Ein Zugang für Personen die nicht an der MUW angestellt sind, was bei MCS sehr wahrscheinlich der Fall sein wird, bedeutet eine Sonderstellung in den Sicherheitspolicies des ITSC. Demnach erhalten ausschließlich Personen, die eine MUW User ID haben einen Zugang zum Datennetz. Externer Zugang also von außerhalb der Firewall über VPN wird nur jenen Personen ermöglicht, die eine MUW User ID besitzen. MUW User IDs werden Personen nur dann zugewiesen, wenn diese entweder Bedienstete der MUW sind, Forschungsstipendiaten, Privatdozenten, oder assoziierte Mitarbeiter der MUW. Dabei handelt es sich um Personen, die mit der MUW oder einer ihrer Organisationseinheiten zusammenarbeiten, kein Dienstverhältnis mit der MUW aufweisen und kein Stipendium beziehen(22). Jeder Arzt oder Assistent, der an einer MCS teilnimmt, müsste also in den Personenkreis der assoziierten MitarbeiterInnen aufgenommen werden, um eine MUW User ID zu bekommen und hätte somit Anrecht auf einen verschlüsselten VPN Zugang. Das Erstellen von Anträgen zur Zuweisung von MUW User IDs hätte einen großen organisatorischen Aufwand für die Studienzentrale zur Folge.
5. Eine weitere Methode zur sicheren Datenübertragung bietet Oracle Security an, in dem es die über das Netz kommunizierten Datenbankabfragen verschlüsselt. Dieser Ansatz bedeutet aber ebenso wie schon oben diskutiert eine Freigabe der Ports in der Firewall für jeden PC, der einen Archimed Client ausführt, sowie

eine Umprogrammierung des Archimed Virtual Clients, um die notwendigen Funktionalitäten zur Nutzung der Oracle Security zur Verfügung zu stellen.

Folgende Voraussetzungen sind bereits im System realisiert die eine Durchführung von MCS ermöglichen:

1. Das Berechtigungskonzept ermöglicht zentrumsspezifische Dateneingabe ohne gegenseitige Sichten auf die erhobenen Daten. Für die Studienzentrale sind für den Zweck des Monitorings, der Auswertung von Daten, sowie anderer Datenmanagementaufgaben alle Sichten über die in den Zentren erfassten Daten im Berechtigungskonzept vorgesehen. Weiters sind die Berechtigungen auf Personenebene nicht nur abhängig von der Gruppe in der sich die Personen befinden sondern ebenfalls von der Rolle, die sie für die Studie zugewiesen bekommen, was ebenfalls im System abbildbar ist.
2. Es besteht die Möglichkeit pseudonymisiert zu dokumentieren. Der Patient erhält eine fortlaufende Nummer als Identifikation. Über diese Identifikationsnummer können bei Follow Ups zum Beispiel weitere Datensätze zu dem Patienten erhoben werden. Die Assoziation zwischen der Identifikationsnummer und dem Patienten muss im jeweiligen Zentrum erfolgen in dem der Patient zur Studie aufgenommen worden ist.
3. Die grafische Benutzerschnittstelle GUI der Clientapplikation ist zwar gut gewartet und wird von qualifizierten Mitarbeitern gut betreut (Telefonhotline, etc.), jedoch auf Grund des großen Funktionsumfangs des Systems Archimed und der damit verbundenen Komplexität bleibt die Verwendung ausschließlich gut geschulten Benutzern vorbehalten. Der Schulungsaufwand für das GUI eines Benutzers der ausschließlich zu MCS dokumentiert wäre gering. Das Look and Feel der GUI jedoch wirkt veraltet und nicht immer intuitiv.
4. Da die Clientapplikation hauptsächlich auf Windows 2000 und Windows XP-Plattformen ausgeführt wird, wurde bei den neuen Versionen der Clientapplikation wenig Augenmerk auf die Plattformunabhängigkeit gelegt. Ein Ausführen der Clientapplikation auf einem Apple Macintosh zum Beispiel ist nur durch eine aufwändige Installation möglich. Die Ausführbarkeit auf PCs mit Linux Plattformen wurde nicht getestet.

#### 4.1.3.1.2 *Kann man MCS webbasiert durchführen?*

Befragte Personen: Archimed-Entwicklerteam

Nein. Das bestehende System basiert auf einer Oracle Datenbank und einer Client Applikation, die in Cincom Smalltalk (Visual Works) realisiert worden ist. Es hat in der Vergangenheit Bestrebungen gegeben, die Applikation mit Hilfe des von Cincom Smalltalk mitgelieferten Webserver Visual Wave für das World Wide Web zu öffnen, die aber aufgrund der hohen Komplexität des Systems und des damalig geringen Ausgereiftheitsgrades des Produktes gescheitert und nicht weiter verfolgt worden sind. Das Problem lag an der Funktionsweise des Visual Wave Webservers. Eine in Smalltalk (Visual Works) geschriebene Applikation wird über die Logik des Webservers in eine HTML Seite übersetzt. Diese Übersetzung scheiterte an dem komplexen GUI der Visual Works Applikation und ließ keinen Spielraum für nachträgliche programmatische Eingriffe offen.

#### *4.1.3.1.3 Ist das Datenmodell flexibel genug, um eine Webapplikation für MCS hinzuzufügen?*

Befragte Personen: Archimed-Entwicklerteam

Aus dem Punkt 4.1.3.1.1 dieses Kapitels geht hervor, dass das Berechtigungskonzept mächtig genug ist, um damit die Rechte der Personengruppen sowie einzelner Benutzer innerhalb MCS abbilden zu können.

Die Authentifizierung des Benutzers am System durch dessen Benutzername und Kennwort erfolgt an der Datenbank, es existiert also kein Applikationsbenutzer. Auch hier gilt, dass die Verbindung zur Datenbank nicht über eine verschlüsselte Leitung läuft, also im Klartext kommuniziert wird, was, wie schon erwähnt, innerhalb des MUW Datennetzes keinen Verstoß gegen die momentane Sicherheitspolicies darstellt, da nicht angenommen wird, dass böswillige Lauschangriffe im Intranet vorgenommen werden.

Die Archimed Systemdatenbank läuft auf einem Oracle Datenbankserver. Das Ziel der Entwicklung war es, dass Kliniker als Endbenutzer nicht für jede Studie eine Datenbank neu aufsetzen müssen. Anstatt dessen gibt es eine generische Datenbank, die als gemeinsamer Speicher für alle klinischen Studien dient. Diese Anforderung verlangt ein generisches Datenmodell, passend für Studiendaten unabhängig von deren Herkunft (unterschiedliche Formulare innerhalb des Systems, oder importierte Daten).

Beim Entwurf des Datenmodells war die wesentliche Frage wie kann man die Daten unterschiedlicher medizinischer Variablen auf einem Formular, in der Datenbank

abbilden. Wir haben uns gegen ein Modell entschieden, in dem eine Tabelle alle Variablen als Spalten hält, da die meisten der gebräuchlichsten Datenbanken ein oberes Limit für die Anzahl der Spalten vorschreiben. Außerdem hätte so ein Design wann immer eine Variable hinzugefügt wird, eine Änderung der Tabellenstruktur zur Folge gehabt. Weiters hätte eine Tabelle dieser Art zum größten Teil aus Nullwerten bestanden, was meist nur in einer Verschwendung von Platz resultiert, da ein Bruchteil der gesamten Variablen von einer Einfügung (Insertion) betroffen ist.

Das Datenmodell, welches wir benutzen ermöglicht dynamische Vergrößerung der Anzahl medizinischer Variablen und Formulare und erlaubt formularunabhängige vorgefertigte Anfragen. Es basiert auf dem *Entity Attribute Value (EAV)* Design, welches eine generische und einheitliche Speicherung der Daten und Metadaten von allen klinischen Studien innerhalb einer Institution ermöglicht. EAV Modelle werden von einigen gut bekannten Systemen benutzt wie z.B. dem HELP System (23), dem Columbia-Presbyterian Medical Centre's clinical repository (24), Yale's Trial/DB system (25), and Oracle Clinical (26). Das EAV Design zielt darauf ab, die Daten einer Entität reihenweise in eine einzige Tabelle mit wenigen Spalten abzulegen, anstatt diese in eine einzige Zeile über mehrere Tabellen und Spalten zu speichern. Der Begriff EAV verweist auf eine Datenmodellstrategie, bei der eine Reihe die Informationen der Entität (Patienten ID, Dokumenten ID, etc.), des Attributes (ID der zu speichernden Variablen), und den Wert (Value) der Variablen beinhaltet.

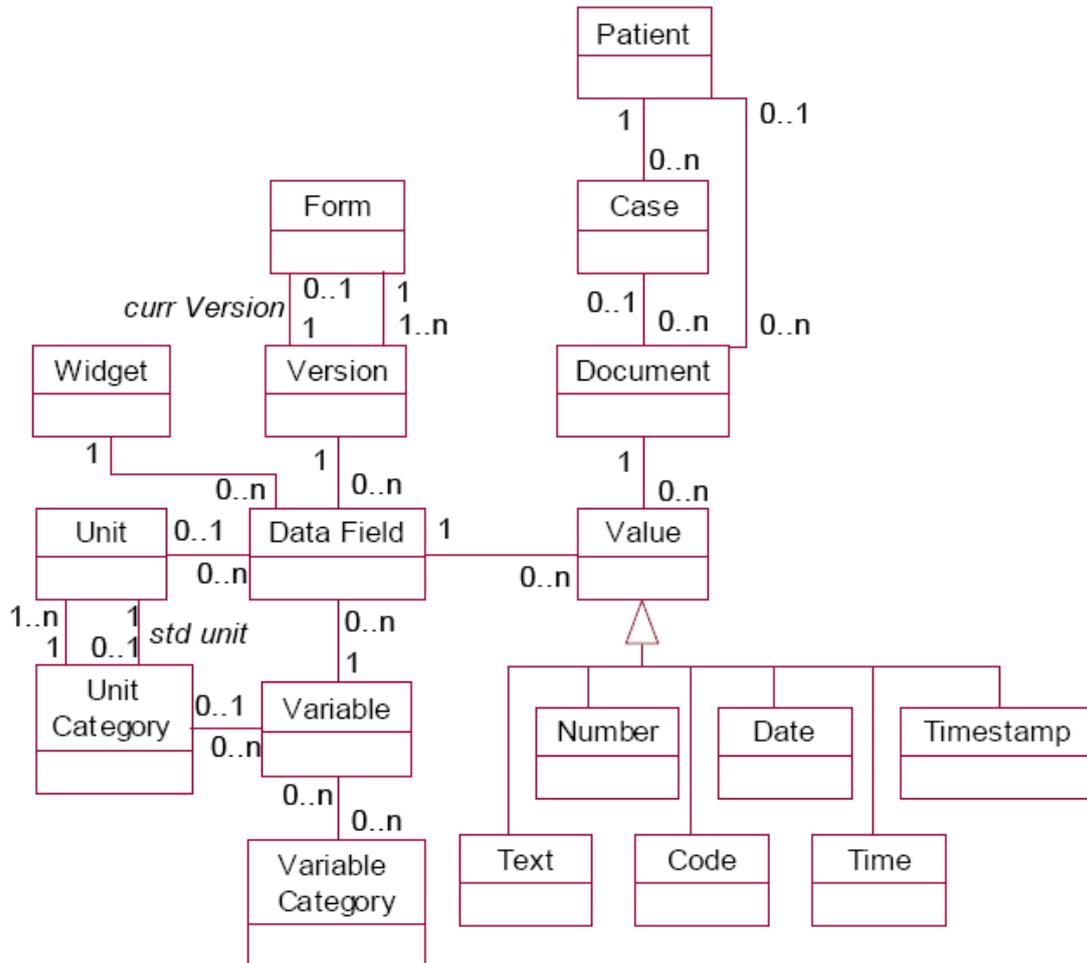


Abbildung 1: Kern Datenmodell des Systems Archimed (20)

Der Kern des Datenmodells, siehe Abbildung 1, besteht aus den folgenden Elementen (20):

1. Der *Variable*-Katalog repräsentiert den Satz der medizinischen Variablen für die Patientendaten erhoben werden können. Über die *Variable Category* können sie semantisch gruppiert werden.
2. Ein Formular (*Form*) beinhaltet die Daten der zugewiesenen Variablen die erhoben werden.
3. Es existieren verschiedene Formularversionen (*Version*), die die Änderungshistorie beinhalten. Eine Version ist vorgesehen als des Formulars aktuelle Version.
4. Wenn eine *Variable* auf ein Formular (*Version*) platziert wird, entsteht ein Datenfeld (*Data Field*) in welches Daten eingetragen werden können. Ein Datenfeld hat einige Attribute die die Repräsentation der zugehörigen Variable

betreffen (z.B. Position auf dem Formular, Font, ...). Des Weiteren ist es mit einem bestimmten *Widget* (z.B. Eingabefeld, Combobox, ...) assoziiert welches das Verhalten des Datenfeldes definiert.

5. Numerische Variablen sind assoziiert mit einer Einheiten-Kategorie (*Unit Category*) (z.B. length, weight), bestehend aus Einheiten, die innerhalb einer Kategorie konvertierbar sind. Innerhalb einer Kategorie ist eine Einheit als Standardeinheit definiert, um Daten einheitlich abzuspeichern. Für die Anzeige der Daten eines numerischen Datenfeldes wird eine Einheit zugewiesen.
6. Wird ein Formular (Version) mit Patientendaten befüllt, dann wird es beim Speichern zu einem Dokument (*Document*).
7. Ein Dokument besteht aus Werten (*Values*) (z.B. Text, Zahl, Code Datum, etc.) und jeder dieser Werte verweist auf ein Datenfeld und dadurch ebenso auf eine Variable.
8. Fälle (*Case*) zu einem Patienten (*Patient*) können ebenfalls erfasst werden, jeder assoziiert mit einem Satz von Dokumenten. Dokumente können direkt zu einem Patienten hinzugefügt werden oder können unzugewiesen bleiben.

#### 4.1.3.1.4 *Welche Möglichkeiten zur Integration von Daten bietet das System Archimed?*

Befragte Personen: Archimed-Entwicklerteam

Die Integration von Daten kann auf mehreren Wegen passieren.

Der Datenbankserver ist über eine Schnittstelle mit dem am AKH Wien verwendeten *KIS* (Krankenhausinformationssystem) verbunden. Über diese Schnittstelle werden die Daten aus dem *KIS* in das System Archimed zeitgetriggert übertragen. Will man in einem Studiendokument *KIS* Daten wie z.B. Laborparameter angezeigt bekommen, so muss man die betreffende Variable auf dem Studienformular definieren und diese mit dem gewünschten Wert vorbelegen.

Weiters besteht die Möglichkeit Daten, die elektronisch in Excel Tabellen, Access Datenbanken etc. erfasst worden sind in das System zu importieren. Als Vorbereitung für den Import müssen zuvor die passenden Formulare mit deren Datenfeldern angelegt werden. Über ein Mapping werden anschließend die Daten in den rudimentären Formaten (Excel, Access, SAS, etc.) zur Deckung gebracht und mittels SQL Statement eingespielt.

4.1.3.1.5 *Ist die Hardware leistungsfähig genug, um eine Webapplikation zu betreiben?*

Befragte Personen: Archimed-Entwicklerteam und Mitarbeiter des MUW Rechenzentrums

Die Hardware des Datenbankservers setzt sich aus folgenden Komponenten zusammen:

1. IBM pSeries 660 6M1
2. 4 Prozessoren RS64IV, 750MHz, 8MB L2
3. 32 GB RAM
4. 2 Fibre-Channel-Adapter
5. 1 Gigabit-Ethernet-Adapter
6. 1 10/100 Mbit-Ethernet-Adapter
7. 1 DVD-RAM Drive
8. 1 Power GXT 135P Grafikadapter
9. 2 interne 18,2GB Harddisks (restl. benötigter Platz wird auf dem ESS zur Verfügung gestellt)
10. Betriebssystem: AIX 5.1

Die laufenden Dienste auf dem Server sind:

1. Archimed DB: die Archimed Studiendatenbank
2. Enterprise DB Backup Service: dient der Überwachung und dem Backup der Studiendatenbank
3. KIS Server (PSI n Prozesse bei n User): dient der Onlineverbindung mit dem KIS
4. Datenübernahme Prozesse betreffend Patienten-, Dokumentdaten, Dienst zur Datenüberleitung vorselektierter Routinedaten in die Archimed Datenbank

Laut Auskunft der Hardwarebetreuung operiert die Hardware bei einer Auslastung von 30%, was einen eventuellen Betrieb eines Web- und Applikationsservers ermöglicht.

4.1.3.1.6 *Soll für die Realisierung eines webbasierten Informationssystems zur Unterstützung medizinischer MCS das bestehende Studiensystem erweitert oder durch ein neues ersetzt werden?*

Befragte Personen: Archimed-Entwicklerteam

Das bestehende Studiensystem sollte erweitert werden:

1. Der große Vorteil der Erweiterung des bestehenden Systems liegt darin, dass die Daten in der Datenbank ohne weiteren Aufwand über die Webapplikation zugänglich wären. Man könnte aktuelle Studien sehr leicht über das Web zugänglich machen.
2. Der bestehende Benutzerkreis müsste nicht auf ein neues System umlernen, sondern könnte die Funktionalität der Client Applikation im Web nutzen.
3. Bestehende Schnittstellen müssten nicht neu implementiert werden.

#### **4.1.3.2 Bewertung**

Die Bewertung gliedert sich in Starkstellen und Schwachstellen und bietet eine übersichtliche Darstellung welches die positiven und negativen Eigenschaften des bestehenden Systems sind.

##### *4.1.3.2.1 Starkstellen*

1. Hohe Flexibilität des Datenmodells.
2. Schnelle Erstellung von Formularen.
3. Möglichkeit zur Durchführung von MCS.
4. Umfassendes Berechtigungskonzept.
5. Leistungsstarke Hardware mit ausreichend Ressourcen.
6. Integration von Routinedaten leicht möglich.

##### *4.1.3.2.2 Schwachstellen*

1. Veraltetes GUI.
2. Mangelnde Plattformunabhängigkeit .
3. Sicherheitslücken in der Kommunikation zwischen Client Applikation und Datenbankserver.
4. Kein webbasierter Zugriff möglich.

#### **4.1.4 Analysebericht**

Wie aus der Systemanalyse und der Bewertung hervorgeht, ist die Architektur der Datenbank mit ihrem Berechtigungskonzept für MCS geeignet.

Eine Verteilung der Clientapplikation außerhalb des Intranets und ein Zugriff auf den Datenbankserver durch Öffnen der Firewallports für die IP Adressen eines beschränkten Benutzerkreises ist aus mehreren Gründen als Lösung unvorstellbar. Erstens bedeutet das eine Abänderung der Sicherheitspolicy des zentralen Informatikdienstes (ITSC). Die Restriktionen der Firewalldurchlässigkeit würde untergraben werden. Zweitens besteht auf Grund der unverschlüsselten Übertragung zwischen Clientapplikation und Server ein erhöhtes Sicherheitsrisiko, welches bei personenbezogener Datenerfassung zu einer Datenschutzrechtsverletzung führen kann. Eine verschlüsselte Übertragung der Daten ist nur über eine Neuprogrammierung der Clientapplikation möglich. Ein Zugang über VPN käme als Lösung für die verschlüsselte Kommunikation nur bedingt in Frage, da die Studienzentrale sich darum kümmern müsste, dass alle Personen, die in den externen Zentren Daten erfassen, eine Benutzerkennung an dem MUW Datennetz (Intranet) zugewiesen bekommen. Diese Zuweisung erteilt das ITSC und nicht die für die Betreuung des Systems Archimed zuständigen Systemexperten. Die Antragstellung für eine Zuweisung eines Zugangs zum MUW Datennetz müsste die Studienzentrale übernehmen und würde eines großen organisatorischen Aufwandes bedürfen.

Abgesehen davon ist eine Verteilung der Clientsoftware auch aus Gründen der grafischen Benutzerschnittstelle (GUI) nicht zu empfehlen. Die grafische Benutzerschnittstelle ist zwar gut gewartet und wird von qualifizierten Mitarbeitern gut betreut (Telefonhotline, etc.), bleibt jedoch auf Grund des großen Funktionsumfangs des Systems Archimed und der damit verbundenen Komplexität die Verwendung ausschließlich gut geschulten Benutzern vorbehalten. Des Weiteren wirkt sich die Teils veraltete Technologie negativ auf die Darstellung und die Bedienung (Look and Feel) aus.

Eine Webapplikation würde viele der oben angerissenen Probleme lösen. Die Plattformunabhängigkeit wäre damit erreicht, da es nahezu für jedes Betriebssystem mittlerweile eine Browser der HTML interpretiert gibt. Die Schnittstelle zwischen dem Datenbankserver und der Webapplikation ist zwar komplizierter als jene zwischen Datenbankserver und Clientapplikation, wird aber von den meisten Anbietern von Applikationsservern auf denen eine Webapplikation ausgeführt wird mitgeliefert. Die Benutzerkennung würde wie auch im Fall der Client Applikation die Datenbank übernehmen, was als grundlegende Voraussetzung für das vorliegende

Berechtigungskonzept gilt. Eine Erweiterung des Berechtigungskonzeptes ist nicht notwendig.

Änderungen am Datenmodell sind ebenfalls nicht notwendig abgesehen von zusätzlichen Tabellen zur Darstellung von Web spezifischen Strukturen, wie zum Beispiel Menüs.

Ein enormer Vorteil der Nutzung des Kerndatenmodells besteht in der Möglichkeit zur webbasierten Darstellung der im System gespeicherten Formulare und Dokumentendaten, was aus rein technischer Sicht dazu führt, dass selbst bisherige Studien die rein über die Client Applikation durchgeführt wurden, zukünftig auch über das Web, also über die Grenzen der Firewall hinaus dokumentierbar sind.

## **4.2 Ergebnisse der Systemspezifikation**

### **4.2.1 Ausgangssituation**

Die Ausgangssituation stellt sich so wie in Kapitel der Systemanalyse beschrieben dar. Der Vollständigkeit halber wird in diesem Abschnitt noch einmal eine kurze Zusammenfassung davon gegeben.

#### **4.2.1.1 Übersicht über das Informationssystem**

Das System Archimed ist ein klinisches Informations- und Dokumentationssystem, das auf einem Entity Attribute Value Datenmodell beruht. Das Modell und die darauf aufbauende Client Applikation ermöglichen eine schnelle Erstellung von Formularen, sowie die Hinterlegung von Constraints zur Überprüfung der Dateneingabe. Das System wird am AKH Wien zur klinischen Dokumentation verwendet. (20, 21)

##### *4.2.1.1.1 Ergebnisse der Systemanalyse*

Siehe Kapitel 4.1.3.1.

##### *4.2.1.1.2 Ergebnisse der Systembewertung*

Siehe Kapitel 4.1.3.2.

##### *4.2.1.1.3 Gewinnung von Anforderungen durch Betrachtung kommerzieller Systeme*

Zur Gewinnung der Anforderung wurden 3 Systeme eResNet (27), MACRO (28), und secuTrial (29) zur näheren Betrachtung herangezogen. Die Auswahl beruht darauf, dass diese Systeme an 12 *KKS (Koordinierungszentren für klinische Studien)* an Universitätskliniken in Deutschland etabliert wurden. Eine der Aufgaben der KKS liegt in der informationstechnischen Unterstützung der Durchführung klinischer Studien, insbesondere der papierlosen Erfassung von Patientendaten durch Electronic Data Capturing (30).

Um die oben genannten Systeme strukturiert zu analysieren, verwende ich die folgende Checkliste:

1. Benutzer Login zur Authentifizierung des Benutzers
2. Systeminformation zur Darstellung aktueller Informationen über das System
3. Studiauswahl: Anzeige der verfügbaren Studien
4. Formularauswahl: Anzeige der verfügbaren eCRFs
5. Monitoring: Review der ausgefüllten eCRFs
6. Studiendokumentation: Erfassen der Patienteninformationen in ein eCRF
7. Benutzeradministration: Kennwortwechsel, Rechtevergabe, etc.
8. Webhilfe
9. Sonstiges: Sonstige Anforderungen

Die auf Grund der Analyse der Systeme gewonnenen Anforderungen fließen in den Abschnitt Anforderungen ein.

### **eResNet – eResearchNetwork**

Die Abarbeitung der Checkliste erfolgt anhand der Online Ressourcen der Firma Research Technology, Inc. (27).

Das eResNet besteht aus 3 Komponenten, dem *eData Management*, dem *eStudy Conduct* und dem *eSafety Net*.

eData Management ist laut Online Ressource ein Internet basiertes System zur Datenerfassung und für Datenmanagementaufgaben. Datenmanagementaufgaben sind das Analysieren der erfassten Daten und das Auffinden von fehlenden oder falschen Dateneingaben. eData Management ermöglicht die Aufbereitung der gesammelten Daten zur Durchführung von statistischen Analysen.

eStudy Conduct ist eine Applikation zur Erstellung, Administration und Überwachung klinischer Studien.

eSafety Net ist eine Applikation die die Einhaltung von internen wie externen Anforderungen zur Berichterstattung von regelwidrigem Vorgehen innerhalb der Studie ermöglicht. Auftraggeber und klinische Forscher können ihre eigenen Prozesse zur Erstellung und Validierung von Formularen (eCRFs) definieren.

Checkliste:

1. Benutzer Login zur Authentifizierung des Benutzers

Aus den Unterlagen geht hervor, dass eine Authentifizierung des Benutzers am System vorgesehen ist.

2. Systeminformation zur Darstellung aktueller Informationen über das System.

Keine Information vorhanden.

3. Studiauswahl: Anzeige der verfügbaren Studien.

Zur Auswahl der Studie steht die eData Management Applikation zur Verfügung.

4. Formularauswahl: Anzeige der verfügbaren eCRFs.

Die Funktionalität der Dateneingabe übernimmt die eData Management Applikation, die Funktionalität der Überwachung der eCRFs übernimmt die eStudy Conduct Applikation.

5. Monitoring: Review der ausgefüllten eCRFs

Die Funktionalität des Monitorings der Daten der eCRFs übernimmt die eData Management Applikation, während das Monitoring der Studie in den Funktionsumfang der eStudy Conduct Applikation fällt.

6. Studiendokumentation: Erfassen der Patienteninformationen in ein eCRF

Die Funktionalität der Datenerfassung wird von der eData Management Applikation bereitgestellt. Zur Erhöhung der Datenqualität können Plausibilitätschecks im eCRF definiert werden.

7. Benutzeradministration: Kennwortwechsel, Rechtenvergabe, etc.

Die Benutzeradministration erfolgt über die eStudy Conduct Applikation. Diese soll die Sicherheit der Daten gewährleisten, in dem die Applikation nur einem eindeutigen Benutzerkreis den Zugang zum System ermöglicht.

## 8. Webhilfe

Keine Information vorhanden

## 9. Sonstiges: Sonstige Anforderungen

Keine Information vorhanden

### **MACRO (InferMed)**

Die Abarbeitung der Checkliste erfolgt anhand von Systemdokumentation der Firma InferMed sowie durch Analyse der Webapplikation des Systems MACRO.

MACRO besteht aus mehreren Applikationen: MACRO Study Definition, MACRO Library Management, WebRDE und WinRDE (Remote Data Entry), MACRO Data Review, MACRO Reports, MACRO Create Data Views, MACRO Query Module, MACRO Archive und MACRO System Management.

MACRO Study Definition ist eine Windows Applikation zum Zweck des Erstellens einer klinischen Studie. Mit dieser Applikation werden auch die eCRFs entwickelt.

MACRO Library Management ist eine Windows Applikation zur Erstellung von Standard-Patientendatenelementen. Es können ebenfalls Standard-eCRFs entwickelt werden, die in sämtlichen Studien zum Einsatz kommen können.

WebRDE und WinRDE sind Applikationen, die eine webbasiert, die andere für Windows Betriebssysteme mit denen man Daten in einem eCRF dokumentieren kann. Die folgende Checkliste wird sich ausschließlich mit dem WebRDE befassen.

MACRO Data Review ist ein Applikationszweig der WebRDE und WinRDE der es Benutzern ermöglicht, den Fortschritt der Dateneingabe zu überwachen.

MACRO Reporting ist ebenfalls ein Applikationszweig der WebRDE und WinRDE der es Benutzern ermöglicht, sämtliche in der Datenbank abgebildeten Informationen darzustellen und auszudrucken.

MACRO Create Data Views ist eine Windows Applikation die dem Benutzer das Erstellen vereinfachter Tabellen ermöglicht, basierend auf dem Datenbestand der Studiendatenbank z.B. zum Zweck des Exportes für statistische Auswerteprogramme.

MACRO Query Module ist eine Windows Applikation, die es dem Benutzer erlaubt Queries oder Filter zu formulieren und diese an die Studiendatenbank zu schicken.

Der Zweck besteht ebenfalls darin, Datenexporte für andere Softwarehersteller zu erzeugen.

MACRO Archive ist eine Windows Applikation und ermöglicht dem Benutzer die Erstellung von Datenarchiven in HTML oder PDF Format. Es wird eine Archivdatei für jede Studie erstellt. Diese Archivdatei enthält in menschenlesbarer Form, Patientendaten und Audit Daten.

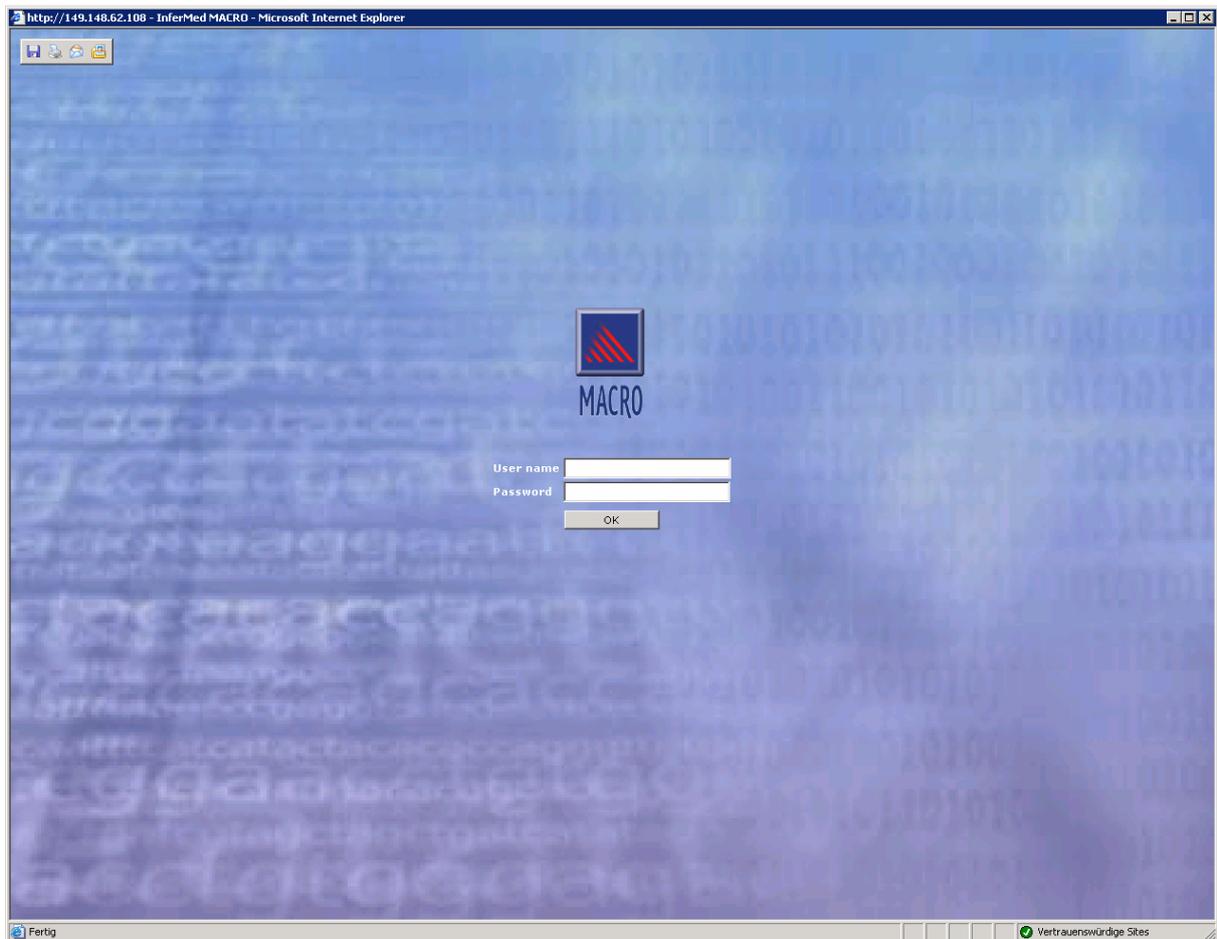
MACRO System Management ist ebenfalls eine Windows Applikation und erlaubt die Administration der gesamten MACRO Applikationspalette. Diese Administrationsaufgaben sind:

1. Anlegen, Registrieren und Administrieren der Datenbank
2. Neue Benutzer dem System hinzuzufügen
3. Benutzerrollen definieren
4. Zuweisen von Benutzerrollen zu Studien
5. Zuweisen und Ändern von Passwörtern
6. Sichten der Logdateien
7. Timeout setzen

Checkliste:

1. Benutzer Login zur Authentifizierung des Benutzers

Ein Benutzer Login zur Authentifizierung des Benutzers ist vorhanden, siehe Abbildung 2, und geschieht mit der Eingabe von Benutzername und Passwort. Bei erfolgreichem Abgleich mit dem im System zum Benutzernamen gespeicherten Passwort wird der Benutzer auf die Home Seite weitergeleitet.



**Abbildung 2: Loginseite der Webapplikation MACRO**

2. Systeminformation zur Darstellung aktueller Informationen über das System  
Die Home Seite bietet sämtliche Funktionen zur Erstellung von Berichten aus dem System (siehe Abbildung 3). Die Darstellung von Systeminformationen ist möglich. Die Auswahl der Berichte, die man erzeugen kann, umfasst Sicherheitsberichte, Metadatenberichte und Datenberichte.

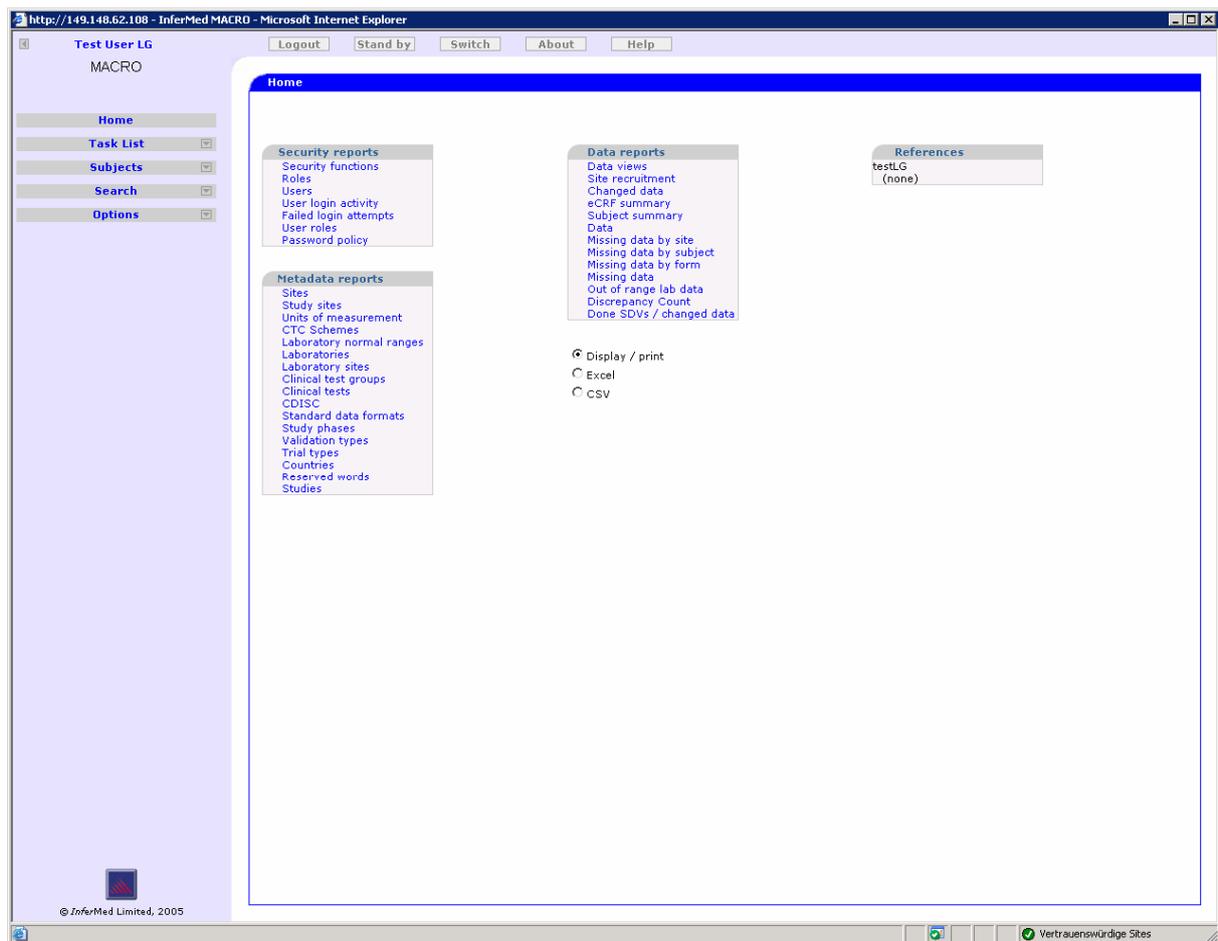


Abbildung 3: MACRO Home Seite, die nach erfolgreichem Login angezeigt wird

### 3. Studienauswahl: Anzeige der verfügbaren Studien

Über das linke Menü navigiert man sich durch die Hauptfunktionen der Webapplikation. Das Menü ist in einer Baumstruktur organisiert. Durch Anklicken der Knoten öffnet sich die *Task List* mit den Aufgaben, die das System beim Dokumentieren unterstützten.

„*Create new subject*“ erzeugt einen neuen Patienten oder allgemein ein neues Subjekt zu dem Daten erhoben werden können. Beim Anwählen dieses funktionalen Menüpunktes öffnet sich eine Seite *New Subject*, die einen Überblick der Studien bietet, zu denen der Benutzer dokumentieren darf (siehe Abbildung 4).

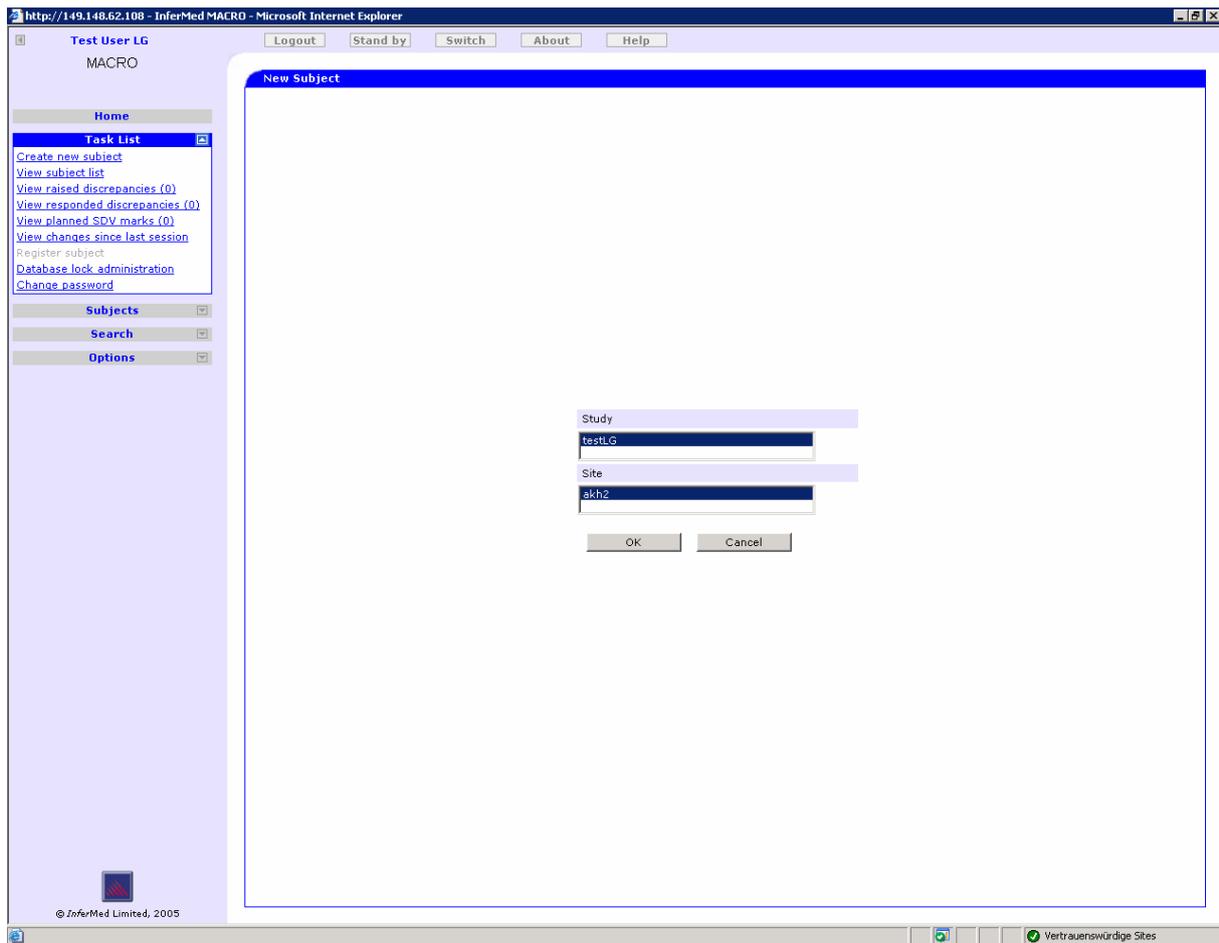
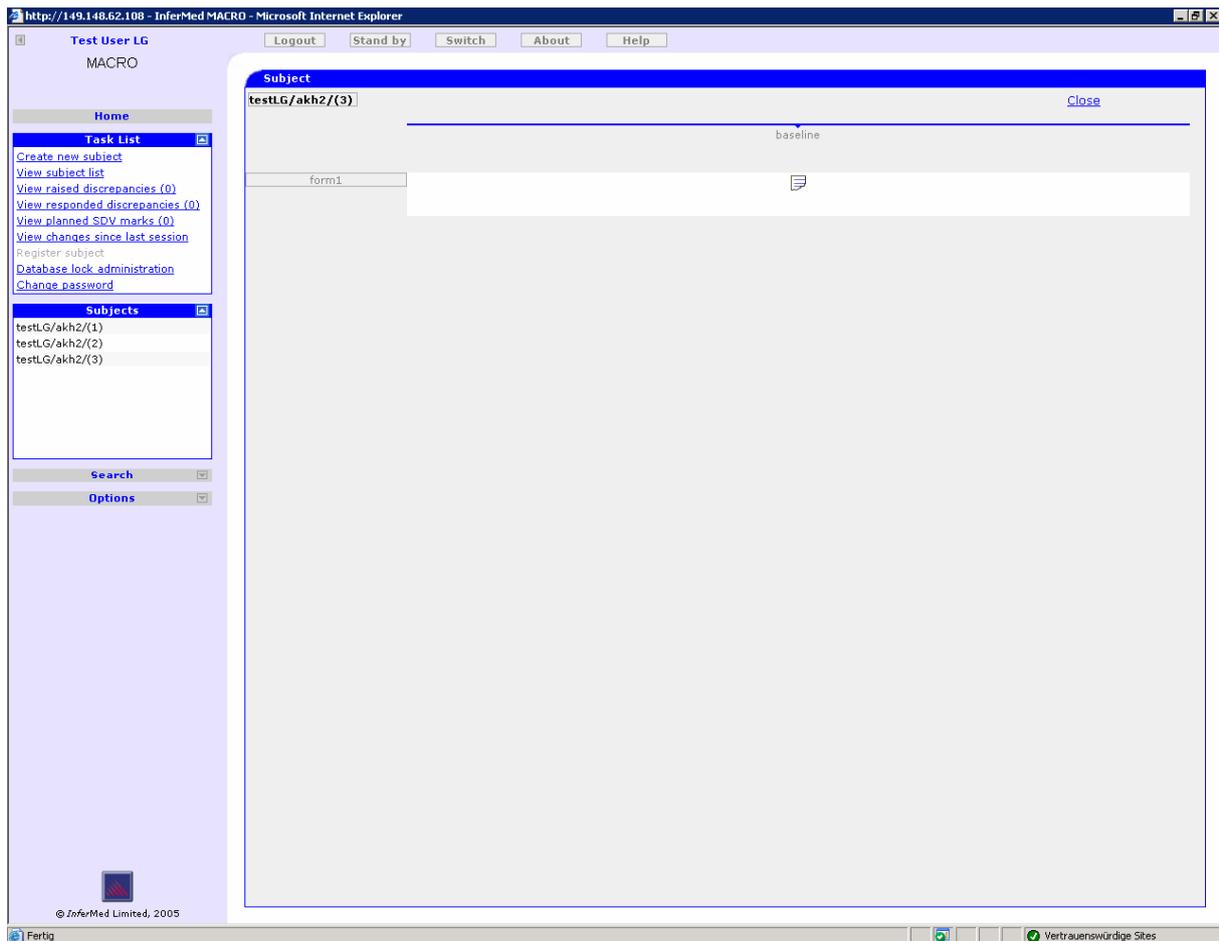


Abbildung 4: New Subject Seite mit der Auswahl der Studie

#### 4. Formularauswahl: Anzeige der verfügbaren eCRFs

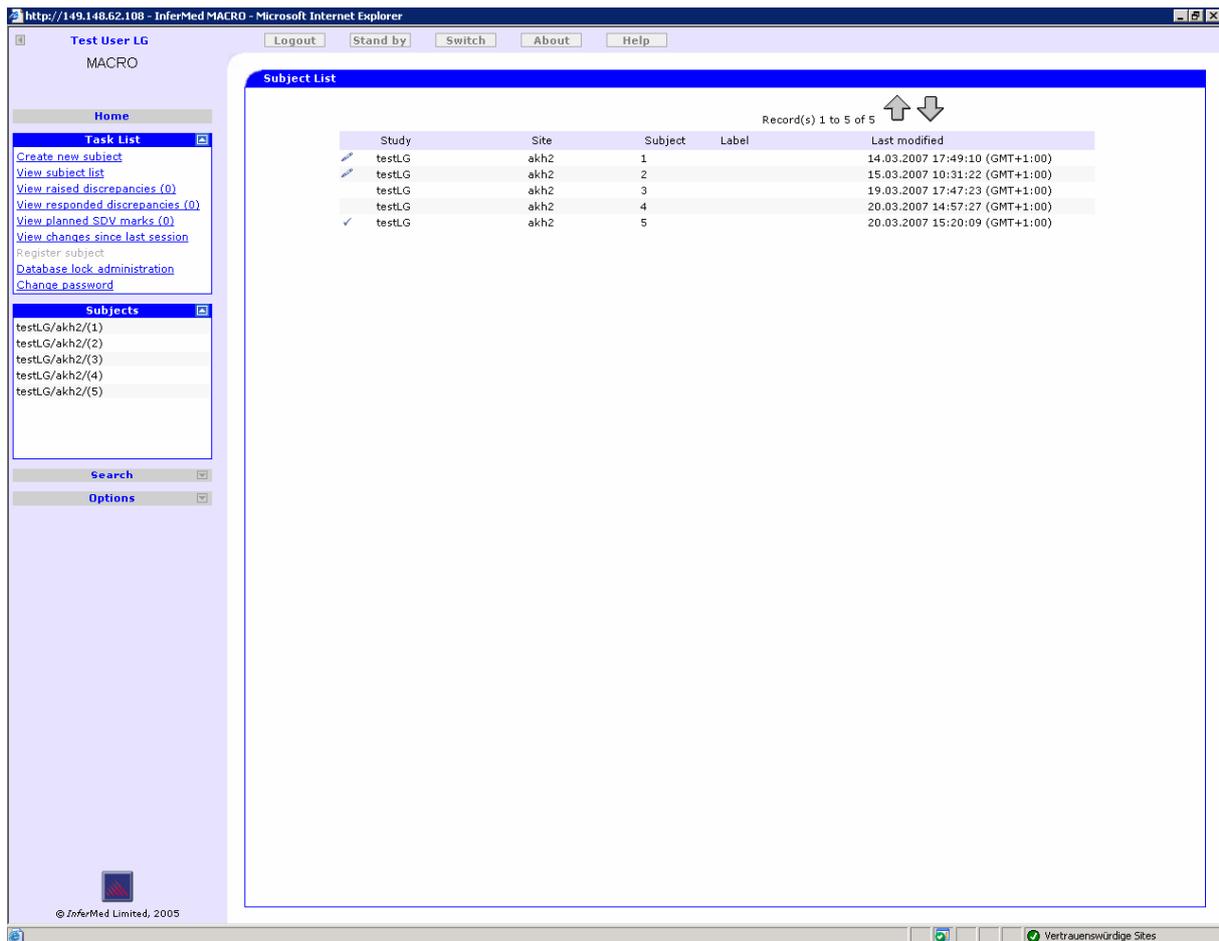
Mit der Auswahl einer Studie und dem Bestätigen durch das Klicken auf OK öffnet sich die *Subject* Seite (siehe Abbildung 5). Diese Seite zeigt die Formulare – eCRFs, die bei dieser Studie und bei diesem Subjekt zu dokumentieren sind. Gleichzeitig mit dem Öffnen der Subject Seite wird im linken Menü unter dem *Knoten Subjects* ein neuer Eintrag für das soeben angelegte Subjekt hinzugefügt.



**Abbildung 5: Formular eCRF Auswahl und die im linken Menü unter dem blauen Balken aufscheinenden Subjekte bzw. Patienten**

## 5. Monitoring: Review der ausgefüllten eCRFs

Ein Review der erfassten Daten ist dadurch möglich, dass man unter dem Menüpunkt Task List auf *View subject list* klickt. Dadurch öffnet sich die *Subject List* mit den bestehenden Subjekten (siehe Abbildung 6). Symbole zeigen an, ob zu einem Dokument also einem ausgefüllten eCRF die Daten erhoben wurden, und ob das Dokument vollständig ausgefüllt ist. Durch Klicken auf eine Zeile der Subject List öffnet sich die *Subject* Seite (siehe Abbildung 5) und man kann dort durch Anklicken der Dokumente die Daten reviewen (siehe Abbildung 7).



**Abbildung 6: Subject List**

In der Subject List (siehe Abbildung 6) erkennt der Benutzer, ob die Dokumente zu einem Subjekt vollständig („Hackerl-Symbol“), teilweise („Bleistift-Symbol“) oder noch nicht erfasst worden sind (kein Symbol am Anfang der Zeile).

## 6. Studiendokumentation: Erfassen der Patienteninformationen in ein eCRF

Das Erfassen der Subjektinformationen also der Patienteninformationen erfolgt durch Anklicken eines eCRFs siehe Punkt 4. Es öffnet sich das Dokument und es kann mit der Erfassung der Daten begonnen werden. Bereits ausgefüllte Felder werden mit einem „Hackerl Symbol“ gekennzeichnet, Felder die man gerade bearbeitet, sind gelb hervorgehoben (siehe Abbildung 7).

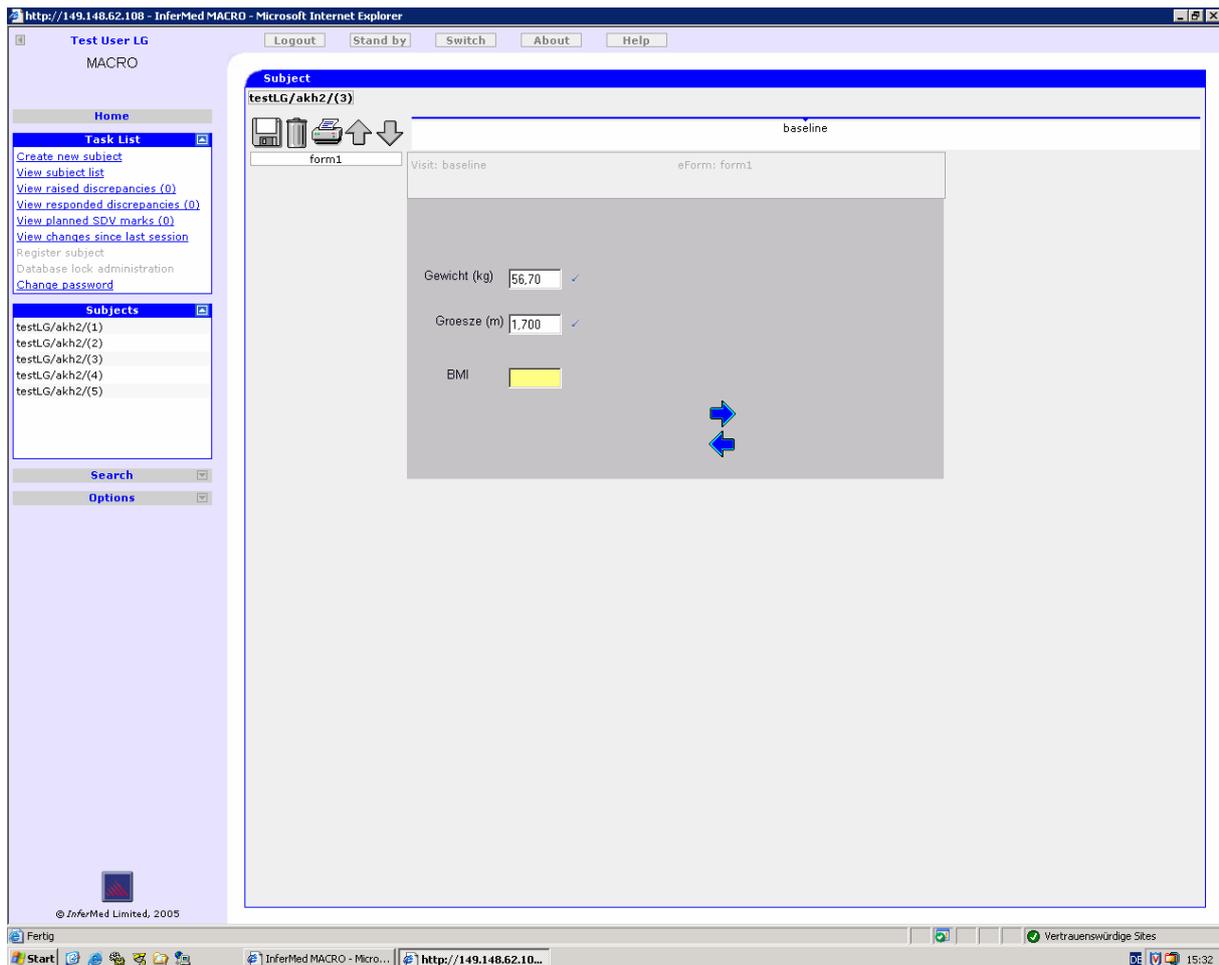


Abbildung 7: ein geöffnetes Dokument

## 7. Benutzeradministration: Kennwortwechsel, Rechtenvergabe, etc.

In der Task Liste im linken Menü befindet sich der *Change password* Menüpunkt. Durch Anklicken dieses öffnet sich ein Change Password Dialog (siehe Abbildung 8), der dem Benutzer das Ändern seines Passwortes ermöglicht. Des Weiteren besteht die Möglichkeit, die Webapplikation über Anklicken des *Logout* Menüpunktes im oberen horizontalen Menü zu verlassen. Ebenfalls im oberen horizontalen Menü befindet sich der Menüpunkte *Stand by* welcher durch Anklicken die Benutzersitzung speichert und beim nächsten Einstieg des Benutzers diese Sitzung erneut lädt. Des Weiteren befindet sich im oberen horizontalen Menü ein *Switch* Eintrag der durch Anwählen den Benutzer abmeldet, um einen anderen Benutzer anzumelden. Weitere Benutzeradministrationsaufgaben sind über die Webapplikation nicht möglich.

The image shows a web browser dialog box titled "Change Password -- Webseitendialog". The user is identified as "Test User LG". The dialog prompts the user to "Please update login password for user Test User LG". It contains three input fields: "Old Password", "New Password", and "Confirm Password". There are "Cancel" and "Close" links in the top right corner. The address bar at the bottom shows the URL "http://149.148.62.108/app/DialogChangePassword.asp" and a "Vertrauenswürdige Sites" (Trusted Sites) icon.

**Abbildung 8: Kennwortwechsel**

#### 8. Webhilfe

Eine Webhilfe ist vorgesehen. Aufgrund der aufwändigen Konfiguration der Webhilfe stand die Testinstallation nur ohne Webhilfe zur Verfügung.

#### 9. Sonstiges: Sonstige Anforderungen

Es besteht die Möglichkeit im Menü unter dem Knoten Search (siehe Abbildung 9) umfassende Suchanfrage auf die im System gespeicherten Studiendaten abzusetzen.

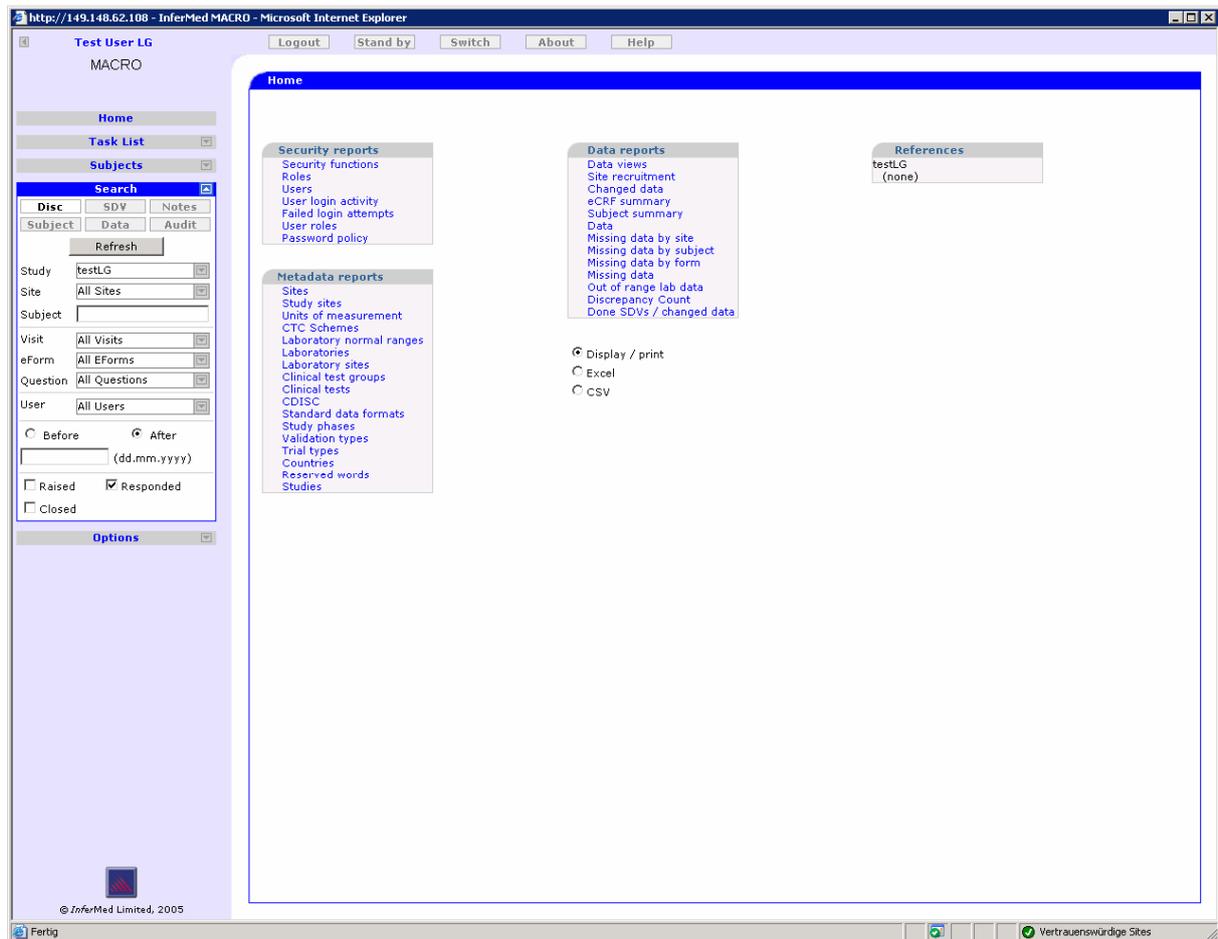


Abbildung 9: Home Seite mit geöffnetem Menüknoten Search

## secuTrial

Die Abarbeitung der Checkliste erfolgt anhand der Online Ressourcen der Firma interActive Systems (31). Da ausschließlich online Ressourcen, wie zum Beispiel eine geführte Vorstellung des Systems in der Art eines Videos zur Verfügung stehen, gestaltet sich eine Abarbeitung der Checkliste als sehr schwierig und keineswegs vollständig.

Checkliste:

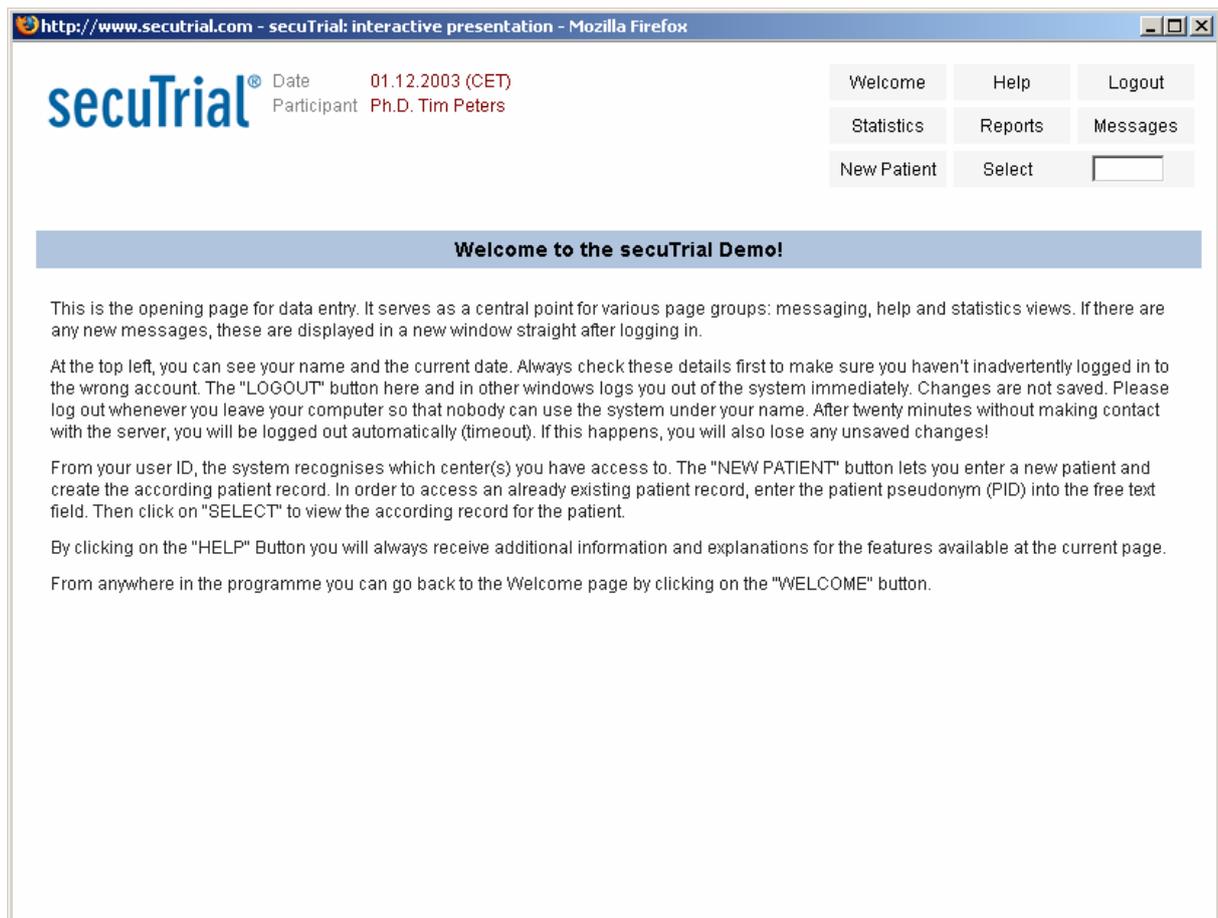
1. Benutzer Login zur Authentifizierung des Benutzers

Ein Benutzer Login zur Authentifizierung des Benutzers ist vorhanden siehe Abbildung 10 und geschieht mit der Eingabe von Benutzernamen und Passwort. Bei erfolgreichem Abgleich mit dem im System zum Benutzernamen gespeicherten Passwort wird der Benutzer auf die Welcome Seite weitergeleitet.



**Abbildung 10: Login Seite des Systems secuTrial**

2. Systeminformation zur Darstellung aktueller Informationen über das System  
Die nach der erfolgreichen Authentifizierung angezeigte Welcome Seite beinhaltet aktuelle Informationen über das System (siehe Abbildung 11).  
Das Navigationsmenü befindet sich in der rechten oberen Ecke. Über dieses lassen sich die Funktionen des Systems aufrufen.



**Abbildung 11: die Welcome Seite siehe ist die erste Seite nach erfolgreichen Anmeldung am System**

### 3. Studienauswahl: Anzeige der verfügbaren Studien

Es sind keine Informationen über eine Auswahl der Studie verfügbar.

Ein Benutzer wird einer Studie automatisch zugewiesen. Dadurch werden alle Patienten, die dieser Benutzer dokumentiert automatisch dieser Studie zugewiesen.

### 4. Formularauswahl: Anzeige der verfügbaren eCRFs

Siehe 5.

### 5. Monitoring: Review der ausgefüllten eCRFs

Durch Eingabe eines Pseudonyms zur Identifizierung eines Subjektes oder Patienten gelangt man zur Patienten Seite (siehe Abbildung 12). Auf dieser Seite sieht man die Dokumente des Patienten sowie den Stand der Datenerfassung. Blau gefärbte Dokumente signalisieren eine vollständig erfolgte Datenerfassung, teilweise blau gefärbte Dokumente zeigen an, dass noch Daten im Dokument fehlen, grau gefärbte

Dokumente zeigen an, dass zu diesem Dokument noch keine Daten erhoben wurden.



Abbildung 12: Patientenseite

## 6. Studiendokumentation: Erfassen der Patienteninformationen in ein eCRF

Durch das Anklicken eines Dokumentes öffnet sich die eCRF Seite. Hier dokumentiert man die Patienteninformationen. In der Abbildung 13 sieht man die ausgefüllten Patientendatenelemente.

http://www.secutrial.com - secuTrial: interactive presentation - Mozilla Firefox

**secuTrial** Project: secuTrial Demo Country: United Kingdom  
 Date: 01.12.2003 (CET) Center: AG London-T  
 Clinical Investigator: Ph.D. Tim Peters Patient: aak938  
 Formfamily: Personal Data Form: Personal Data

Welcome Help Logout  
 Print Back  
 Queries Comments Audit Trail

**A Personal Data** Document-No. 340 - 2

**A1. Date of birth** ? Help  
 [02] - [1920] mm.yyyy Comment  
 Query

**A2. Gender** Comment  
 female  male Query

**A3. Date of examination** Comment  
 [02] - [03] - [2003] dd.mm.yyyy Query

**A4. Date of data entry** Comment  
 [01] - [04] - [2003] - [ ] - [ ] dd.mm.yyyy hh:mi Query

**A5. Diagnosis** Comment  
 Depression Query  
 if OTHERS: specify

**A6. Time of symptom onset** Comment  
 leave month empty if not known  
 [01] - [ ] - [2003] mm.yyyy Query

Cancel Save and Print Save  Check data

Date: 01.12.2003 (CET) Clinical Investigator: Ph.D. Tim Peters Patient: aak938 Visit: 01.01.2003  
 Project: secuTrial Demo Center: AG London-T Formfamily: Personal Data Form: Personal Data Document-No. 340 - 2

Abbildung 13: Dokument (eCRF) zu einem Patienten

7. Benutzeradministration: Kennwortwechsel, Rechtevergabe, etc.

Keine Information vorhanden.

8. Webhilfe

Keine Information vorhanden.

9. Sonstiges: Sonstige Anforderungen

Keine Information vorhanden

#### 4.2.1.1.4 Weitere relevante Randbedingungen

Als wesentliche Randbedingung innerhalb der Organisationseinheit, die das System Archimed entwickelt hat und betreut, tauchte in der Vergangenheit der Wunsch nach einer webbasierten Datenerfassung für MCS auf.

Ebenso relevant (besonders für die Implementierung) sind die Forderungen der Organisationseinheit das System Archimed in Java2 weiterzuentwickeln.

## **4.2.2 Zieldefinition und Abgrenzung**

Das Ziel besteht darin, ein webbasiertes Studiensystem zur Durchführung von MCS zu spezifizieren.

Aus der Systemanalyse und Bewertung geht hervor, dass es sinnvoller ist, das bestehende System um einen webbasierten Applikationszweig zu erweitern. Dadurch soll erreicht werden, dass die Formulare sowie die gespeicherten Daten der Dokumente in der Datenbank über das Webinterface zugänglich gemacht werden.

Es soll versucht werden an das Datenmodell sowie die Benutzerverwaltung in deren bestehenden Formen einen weiteren Applikationszweig hinzuzufügen. Die Anforderungen dieses Applikationszweiges sollen die eines MCS erfüllen mit der Auflage der Integration in das bisherige System. Diese Anforderungen werden im Abschnitt 4.2.4 spezifiziert.

### **Abgrenzung**

Eine Systemspezifikation mit dem Ziel der Neuimplementierung des gesamten Datenbankmodells ist nicht anzustreben.

Ebensowenig ist es geplant, ein Studiensystem komplett neu zu programmieren. Aufbauend auf der Datenbank mit dem bestehenden Datenmodell sowie dem bestehenden Datenbestand soll eine Webapplikation entstehen, die es ermöglicht MCS durchzuführen.

## **4.2.3 Produkteinsatz und Produktumgebung**

Ein webbasiertes, medizinisches Informationssystem zur Unterstützung klinischer MCS soll dazu dienen, Studiendaten von mehreren Orten zentral zusammenzuführen, um sie anschließend einer Auswertung zugänglich zu machen. Das System soll an der Medizinischen Universität Wien unter der inhaltlichen Betreuung des Instituts für Medizinische Informations- und Auswertesysteme vom ITSC gehostet werden. Benutzer des Systems sind Kliniker, Studenten und Assistenten, sowie Systemexperten zur inhaltlichen Betreuung.

Das System soll für eine potentielle weltweite Datenerfassung zum Einsatz kommen.

Die Webapplikation soll der Erfassung von anonymisierten und nicht-anonymisierten Patienteninformationen über das Internet dienen.

## **4.2.4 Anforderungen**

In diesem Abschnitt ist die Rede von nichtfunktionalen und funktionalen Anforderungen. Die funktionalen Anforderungen sind jene Anforderungen, die aus Sicht des Benutzers die Funktionalität des Systems beschreiben. Nichtfunktionale Anforderungen beschreiben, welche Aufgaben das System zusätzlich noch erfüllen muss. Die funktionalen und nichtfunktionalen Anforderungen basieren unter anderem auf der in Kapitel 2.3 dargestellten FDA-Richtlinie.

### **4.2.4.1 Nichtfunktionale Anforderungen**

1. Der Aufbau einer sicheren Verbindung zwischen Webbrowser und Webserver bzw. Applikationsserver mittels https ist notwendig. (https steht für Hypertext Transfer Protocol Secure und dient zur Verschlüsselung und zur Authentifizierung der Kommunikation zwischen Webserver und Browser)
2. Authentifizierung durch Eingabe von Username und Passwort in einem Webformular am Beginn jeder Arbeitssitzung (Session) der Webapplikation → die https Verschlüsselung muss bereits aktiv sein!
3. Authentifizierung über Webapplikation an der Datenbank.
4. Sessionweite https Verschlüsselung.
5. Applikationsweite Identifikation welcher Benutzer angemeldet ist. Man sollte auf jeder Seite erkennen können, welcher Benutzer gerade angemeldet ist.
6. Eindeutige Zuweisung einer Session zu einem Benutzer. Alle Aktionen, die innerhalb dieser Sessionen getätigt werden, stammen von dem authentifizierten Benutzer.
7. Benutzerabhängiger Start der Webapplikation: Abhängig davon, welcher Benutzer sich anmeldet, soll konfiguriert werden können welche Studie aufgeht.
8. Kontextabhängiger Menüaufbau: ein Navigationsmenü für die funktionalen Applikationszweige der Webapplikation (z.B. Studiendokumentation). Ein zweites Navigationsmenü für die nicht funktionalen Rahmenapplikationen der Webapplikation (z.B. Passwortwechsel, Hilfe). Menüs sollen wiedergeben an welcher Stelle man sich innerhalb der Applikation befindet (z.B. durch Baumstruktur der Navigationsmenüs)

9. Seitenaufbau: der Seitenaufbau sollte so gestaltet werden, dass man nach Möglichkeit nicht scrollen muss.
10. Möglichkeit zu bestehenden Studien webbasiert zu dokumentieren, sowie deren Dokument über das Web abrufen zu können.
11. Die Möglichkeit weiterhin über die Archimed Clientapplikation zu dokumentieren, soll bestehen bleiben.
12. Der Benutzer sollte die Möglichkeit haben zwischen deutschen und englischen Bildschirmausgaben zu wechseln.

#### **4.2.4.2 Funktionale Anforderungen**

1. Die Seite, mit der die Applikation gestartet wird, soll die Login Seite sein. Auf dieser Seite findet die Authentifizierung des Benutzers durch seinen Benutzernamen und sein Kennwort statt. Es soll kein funktionales Menü angezeigt werden. Das nichtfunktionale Menü besteht aus einem Link zum Öffnen der Hilfe, und einem zweiten Link zum Wechseln zwischen deutscher und englischer Darstellung der Bildschirmausgaben der Webapplikation.
2. Hauptseite: die Seite, die angezeigt werden soll nachdem der Authentifizierungsvorgang erfolgreich beendet worden ist, ist die Hauptseite. Die Seite kann Informationen über Wartungszeiten, das Betreuungsteam, sonstige Informationen zum System, Links und Hinweise beinhalten. Die Hauptseite sollte Inhalt des funktionalen Navigationsmenüs sein.
3. Studienauswahl: Die Seite sollte eine Liste der Studien anzeigen abhängig von den Berechtigungen des Benutzers. Jeder Benutzer darf auf der Seite nur jene Studien sehen und öffnen zu denen er Berechtigungen hat. Eine Administration also Löschen oder Ändern von Studien soll an dieser Stelle nicht möglich sein. Die Liste der Studien sollte zumindest den Bezeichner der Studie, den Eigentümer, Kommentar zur Studie, sowie einen Informationsfeld pro Zeile und Studie vorsehen. Des Weiteren sollte in der Zeile die Möglichkeit geboten werden, die Studie aufzurufen und damit auf die Seite „Studiendokumentation“ zu wechseln. Die Seite Studienauswahl sollte im funktionalen Navigationsmenü eingetragen sein.
4. Auf der Seite der Studiendokumentation sollen abhängig von den Berechtigungen des Benutzers die Informationen der zuvor ausgewählten Studie angezeigt werden. Informationen einer Studie sind: Liste der Formulare, Liste

der Dokumente (ausgefüllte Formulare), anonyme oder nicht anonyme Liste der Patienten, Liste der Dokumenteneigentümer, Dokumentendatum. Ein Benutzer darf nur jene Formulare und Dokumente angezeigt bekommen, öffnen, ändern, oder löschen, für die er die jeweilige Berechtigung zugewiesen bekommen hat. Auf dieser Seite muss die Möglichkeit bestehen auf Basis eines Formulars ein neues Dokument zu einem Patienten anzulegen. Das Formular muss bereits in dem System bestehen. Weiters sollte man beim Anlegen des Dokuments das Dokumentendatum und den Dokumenteneigentümer (Gruppe) spezifizieren können. Zusätzlich zu der Möglichkeit des Anlegens von Dokumenten sollte auf der Seite eine Liste der bestehenden Dokumente dargestellt werden. Diese Liste sollte für jedes Dokument Informationen wiedergeben wie Dokumentendatum, Patient, Formular, Informationen aus definierten Dokumentenfeldern, Identifikationsnummer und einen Informationsbutton, der detaillierte Dokumenteninformationen liefert. Weiters sollte man ein Dokument aus der Dokumentenliste öffnen bzw. editieren oder löschen können. Die anzuzeigenden Dokumente sowie deren Editierbarkeit und Löscharkeit sind abhängig von den Berechtigungen des Benutzers. Ebenfalls sollte diese Seite die Möglichkeit bieten zu einer Dokumentensuche zu gelangen, bzw. Dokumente über eine Filteransicht schnell aufzufinden. Die Studiendokumentation soll zum Zeitpunkt zu dem die Seite angezeigt wird, im funktionalen Navigationsmenu aufscheinen.

5. Durch Anlegen eines neuen Dokuments oder durch Öffnen eines bestehenden Dokuments gelangt man zur Dokumentenseite. Abhängig davon wird entweder ein neues Dokument angelegt, oder die Daten eines bestehenden Dokumentes geladen. Zusätzlich zu den Informations- und Eingabefeldern des Formulars bzw. des Dokuments sollen folgende Informationen und Eingabefelder angezeigt werden: Identifikation des Dokuments, Eigentümer des Dokuments, assoziierter Patient, änderbares Dokumentendatum. Das Formular bzw. das Dokument soll nach Möglichkeit der Anzeige in der Client Applikation gleichen. Man muss ein Dokument speichern oder verwerfen können. Betreffend letzteres sollte eine Warnung kommen bevor die Änderungen im Dokument gelöscht werden. Nach dem Verwerfen oder Speichern des Dokumentes gelangt man zurück zur Studiendokumentationsseite. Die Seite Dokument soll zum Zeitpunkt des Öffnens eines bestehenden bzw. des Anlegens eines neuen Dokumentes im

funktionalen Navigationsmenu unterhalb der Studiendokumentation als Menüpunkt angezeigt werden.

6. Die Seite Kennwortwechsel sollte die Möglichkeit bieten, nach Eingabe des alten Kennwortes das systemweite Kennwort zu ändern. Die Seite Kennwortwechsel sollte im nichtfunktionalen Menü bei der Anzeige der Hauptseite aufscheinen.
7. Ebenfalls im nichtfunktionalen Menü soll sich ein Link befinden, der die Möglichkeit bietet, zwischen der deutschen Darstellung der gesamten Webapplikation und der englischen zu wechseln.
8. Die Hilfe-Seite ist als Webhilfe zu gestalten und sollte in einem zweiten Fenster oder Browserregistrierkarte (Tab) aufgehen. Die Hilfeseite wird im nichtfunktionalen Menü zu jedem Zustand der Webapplikation angezeigt. Beim Klicken auf den Menüeintrag Hilfe soll eine kontextabhängige Hilfeseite geöffnet werden.
9. Als letzter Eintrag von links nach rechts des nichtfunktionalen Menüs soll der Link Beenden angezeigt werden. Dieser beendet die Applikation. Dabei wird die Benutzersession gelöscht und damit alle benutzerassoziierten Informationen die während der Session gespeichert werden. Danach soll auf die Loginseite verwiesen werden.

## **4.2.5 Weitere Vorgaben**

### **Einführung**

Die Einführung der Webapplikation erfolgt nach einer Testphase. Die Testphase wird auf einer gespiegelten Datenbank und einem für Testzwecke vorgesehen Web- und Applikationsserver durchgeführt. Nach Abschluss der Testphase werden die notwendigen additiven Datenbankstrukturen in die Produktionsdatenbank eingespielt. Nach der Einführung soll aus Sicherheitsgründen der Applikationsserver und der Webserver auf dem gleichen physischen Server laufen auf dem die Datenbank läuft. Der Grund dafür besteht darin, dass eine verschlüsselte Kommunikation zwischen Datenbankserver, Applikationsserver und Webserver notwendig wäre, falls die Webapplikation auf einem anderen Gerät als auf dem, auf dem die Datenbank läuft ausgeführt wird.

### **Betrieb**

Um eine Studie für das Web zu öffnen bedarf es der Kontaktaufnahme des Benutzers mit dem Institut für Medizinische Informations- und Auswertesysteme.

Abhängig davon, ob eine Studie und die dazugehörigen Formulare im System bereits existieren oder nicht, muss in der Client Applikation eine neue Studie angelegt werden. Anschließend werden die Formulare zu der Studie ebenfalls in der Clientapplikation über den Formulardesigner entworfen. (20) Somit existieren die notwendigen Datenstrukturen in der Datenbank, die das System als Basis für die Dokumente benötigt. Ab diesem Zeitpunkt wäre es bereits möglich, über die Clientapplikation zu dokumentieren. Die fachgerechte Gestaltung der Webformulare muss dann von Mitarbeitern des Institutes MIAS vorgenommen werden. Die Gestaltung der Webformulare soll hauptsächlich von *Webdesignern* mit speziellem Wissen und Erfahrungsreichtum im Bereich HTML, JavaScript und anderen Webtechnologien erfolgen.

### **Wartung**

Die Wartung des Servers auf dem die Datenbank und der Applikations- und Webserver läuft geschieht durch das ITSC der Medizinischen Universität Wien.

Die Wartung der Datenstrukturen und der Datenkonsistenz erfolgt durch die Mitarbeiter des Instituts MIAS.

Die Wartung des Applikations- und Webservers soll ebenfalls von den Mitarbeitern des Instituts MIAS vorgenommen werden.

## **4.3 Ergebnisse der Eigenentwicklung**

Die Entscheidung gegen eine Standardsoftware, fällt auf Grund des hohen Maßes der im Pflichtenheft verlangten Integration des Systems in das bestehende, selbst entwickelte System Archimed. Konfiguration, Datenüberleitungen, zusätzliche Kommunikationskonzepte würden den Rahmen einer Standardsoftware sprengen, und wären im Falle einer Anschaffung Teilprojekte erheblichen Ausmaßes.

### **4.3.1 Auswahl der Programmiersprache und der Entwicklungsumgebung**

Dieser Abschnitt konzentriert sich auf Programmiersprachen und Entwicklungsumgebungen, die im Zusammenhang mit Web-Engineering von Relevanz sind. Hierbei lassen sich mehrere Kategorien von Programmiersprachen unterscheiden (19):

1. Nicht objektorientierte Programmiersprachen: in dieser Kategorie werden alle Programmiersprachen zusammengefasst, die das objektorientierte Paradigma in keiner Form unterstützen wie PHP, Perl, C.
2. Objektbasierte Programmiersprachen: Objekte werden unterstützt (z.B. VBScript und JScript).
3. Objektorientierte Programmiersprachen: Klassen und Vererbung werden unterstützt (z.B. J2EE /JSP, C#/ASP.NET).
4. Hybridsprachen: Programmiersprachen, deren objektorientierte Mechanismen im Laufe ihrer Entwicklung hinzugefügt wurden wie z.B. C++
5. Voll objektorientierte Sprachen, die entsprechend dem objektorientierten Paradigma entwickelt wurden, wie Smalltalk.

Auf Grund der relativ hohen Komplexität der Problemstellung besteht der Wunsch des Einsatzes von objektorientierten Methoden zur Modellierung des Problembereichs. Objektorientierte Methoden zur Modellierung erhöhen den Abstraktionsgrad eines Problembereichs. Sie ermöglichen einen Prozess der objektorientierten Zerlegung des Problembereichs was eine Vereinfachung der Teilproblembereiche zur Folge hat (32). Aus der Anforderung der objektorientierten Modellierung resultiert die Notwendigkeit, nach einer objektorientierten Programmiersprache das entstandene Modell so effizient wie möglich umzusetzen. Also kommen aus obiger Liste lediglich 2 Technologien für den Einsatz als Web-Applikationen in Frage: J2EE / JSP und C# / ASP.NET.

Die proprietäre ASP.NET und C# Technologie von Microsoft bietet zwar durch das Visual Studio eine mächtige und komfortable Entwicklungsumgebung (33) und ermöglicht den Zugriff auf die Funktionalitäten bzw. Klassenbibliotheken des .NET Frameworks, ist aber von Microsoft Plattformen als Webserver und dem .NET Framework abhängig (34). Anders verhält es sich mit dem offenen J2EE Standard (35), welcher plattformunabhängig ist. Die zugrunde liegende Programmiersprache ist Java2 (36). Java2 bietet ebenfalls eine mächtige Klassenbibliothek. Des Weiteren soll, laut Erfahrungsberichten im Internet, die ASP.NET Technologie Performanceengpässe aufweisen, die der J2EE Standard nicht verzeichnen soll. Die relativ neue Technologie der Java Server Faces (JSF) im Umfeld des J2EE Standards ist ein Framework-Standard im Bereich der Webanwendungen (37, 38). Mit Hilfe von JSF kann der Entwickler auf einfache Art und Weise Komponenten für Benutzerschnittstellen in Webseiten einbinden und die Navigation definieren. Der

Programmierer entwickelt nicht die Ausgabeseite mit HTML, sondern setzt sie auf einem höheren Abstraktionsniveau aus Komponenten zusammen. Ähnlich wie bei der klassischen Anwendungsentwicklung kann sich der Entwickler stattdessen auf die eigentliche Anwendungslogik konzentrieren. Um die Anwendung besser zu strukturieren, wird strikt zwischen „Modell“, „Ausgabe“ und „Steuerung“ getrennt, wie es das Model-View-Controller-Konzept vorsieht.

Zu kritisieren an der J2EE Technologie ist die hohe Komplexität des Standards und die damit verbundenen umständlichen Konfigurationen sowie die aufwendigen Implementierungs- und Deploymentzyklen bei der Entwicklung. Ohne entsprechende Entwicklungsumgebung, die einen Programmierer bei diesen Vorgängen unterstützt, wäre eine Entwicklung nur mit großem Aufwand möglich und würde als „KO Kriterium“ für die Entwicklung mit J2EE Standard stehen. Mit dem Rational Application Developer v6.0 (RAD) (39) von IBM steht jedoch ein vergleichbar mächtiges und komfortables Entwicklungswerkzeuge für den J2EE Standard zur Verfügung wie das Visual Studio 2005 für ASP.Net. Diese Entwicklungsumgebung ermöglicht trotz aufwendigen Deployment- und Entwicklungszyklen eine sehr effiziente Arbeit im Umgang mit dem J2EE Standard.

Alles in allem scheinen die Vorteile des J2EE Standards mit dem Rational Application Developer v6.0 als Entwicklungswerkzeug zu überwiegen.

## **4.3.2 Namenskonventionen und Codegestaltung**

Um während der Implementierung den wechselnden und verschiedenen Personen eine gute Lesbarkeit des Quelltextes zu gewährleisten, sind Namenskonventionen und die Codegestaltung eine Notwendigkeit.

### **4.3.2.1 Namenskonventionen**

- Paketbezeichner werden in Kleinbuchstaben geschrieben.
- Klassennamen beginnen mit Großbuchstaben, zusammengesetzte Klassennamen werden mit der "Camelbacknotation" geschrieben z.B. DatabaseObject
- Konstante werden in Großbuchstaben geschrieben.
- Variablen fangen immer mit Kleinbuchstaben an, zusammengesetzte Variablenbezeichner werden in Camelbacknotation geschrieben z.B. fovePersId.

- Methoden fangen immer mit Kleinbuchstaben an und werden bei zusammengesetzten Methodennamen in Camelbacknotation geschrieben z.B. getInsertProcedure().
- Bezeichnungen für Interfaces werden so geschrieben wie Klassennamen.

#### 4.3.2.2 Codegestaltung

Für die Codegestaltung gilt die im RAD als „Java Conventions [built-in]“ verwendete Vorlage mit der voreingestellten Auswahl.

Abbildung 14 zeigt wie der Code in der Entwicklung zu gestaltet ist.

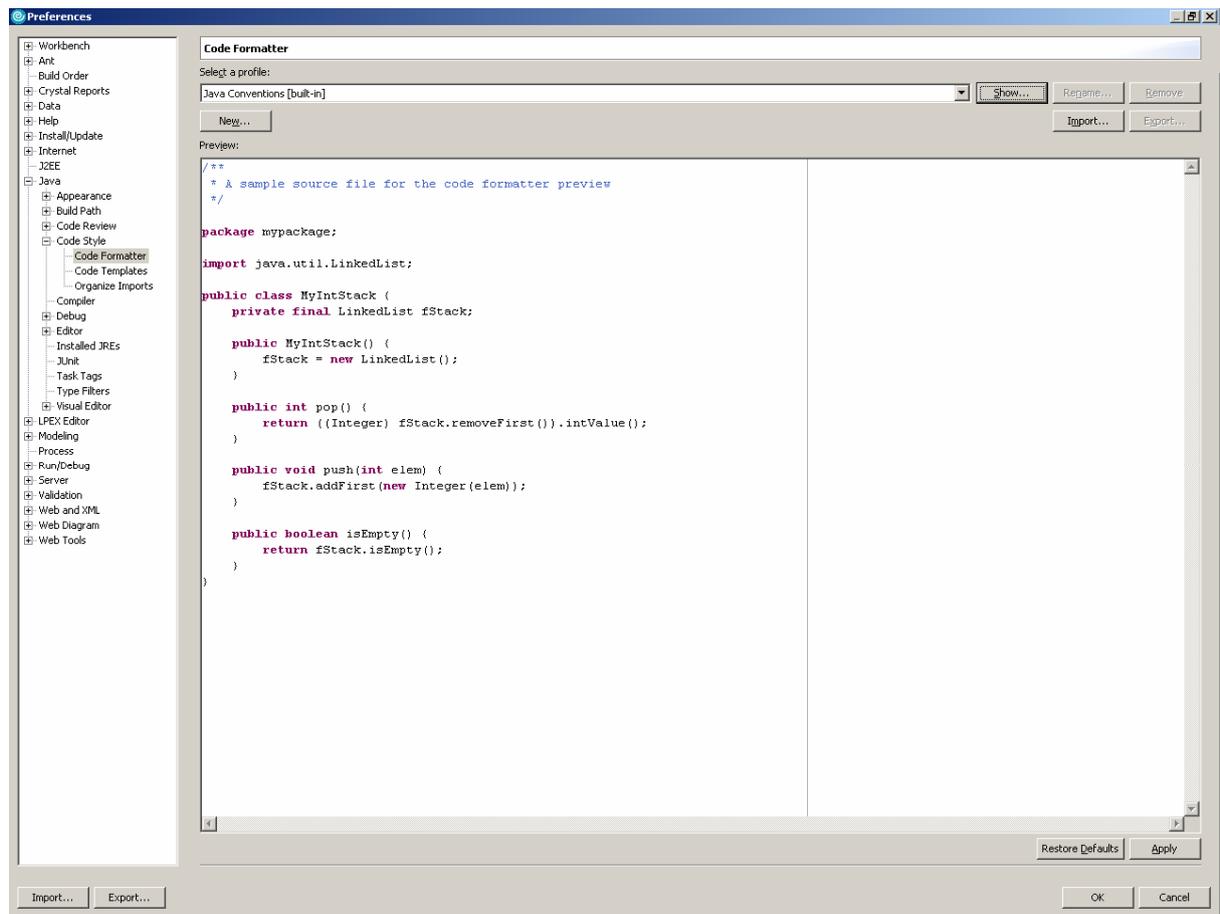


Abbildung 14: Codegestaltung

#### 4.3.3 Integrationsplanung - Bottom Up – Integration

Bei der Bottom Up Integration werden die Hardwareschnittstellen (Datenbankschnittstellen) zuerst entwickelt und getestet, um dann eine fundierte Basis zu bilden, auf der das Programm Schicht für Schicht aufgesetzt werden kann (19).

Als Erstes erfolgt die Implementierung der Datenbankschnittstelle. Die Datenbanktreiber die zur Kommunikation und zum Datenaustausch zwischen dem Applikationsserver Websphere von IBM und der Datenbank notwendig sind, werden vom Hersteller der Datenbank Oracle zur Verfügung gestellt. Der Verbindungspool des Applikationsservers stellt die Verbindungen zur Datenbank der Applikation zur Verfügung. Er hält die Verbindungen zur Datenbank und über Java2 Befehle können diese Verbindungen in der Applikation genutzt werden.

Darauf aufbauend werden die Datenbankobjekte implementiert und getestet. Sie entsprechen dem Modell der Datenbank und halten deren Daten. Auf das Modell der Datenbank und der Applikation wird im Abschnitt 4.3.4 (Paket database und Paket generic) näher eingegangen.

Anschließend werden die GUI Webseiten und die Controller wie Ablaufsteuerung, Validierung, Sortierung, Konvertierung dem Modell aufgesetzt.

#### **4.3.4 Implementierung der Klassen**

Bei der Implementierung der Klassen wurde darauf geachtet diese sinnvoll in Pakete zu untergliedern. Pakete sind Organisationseinheiten die es ermöglichen, ähnlich wie z.B. Ordner in einem Dateisystem, die Klassen zu organisieren. Das Konzept der Pakete ermöglicht mehrere Klassen innerhalb eines Paketes zusammenzufassen. Dadurch kann man Klassen, die semantisch ähnlich sind, gruppieren und verbessert somit die Lesbarkeit des Quelltextes. Auf die Pakete sowie deren Inhalte, also die Klassen, werde ich in den folgenden Absätzen genauer eingehen. Abbildung 15 gibt einen Überblick über die verwendeten Pakete.



**Abbildung 15: Pakete der Applikation.**

#### **4.3.4.1 Paket handler**

Die Klassen des Paketes handler (siehe Abbildung 16) haben die Aufgabe die Zustände, die in der Applikation auftreten zu modellieren und zu implementieren. Zustände sind Instanzen von Klassen aber auch Variablenwerte, die während der Lebensdauer der Applikation gesetzt und bereitgestellt werden müssen, um den Informationsfluss innerhalb der Applikation sicherzustellen. Ein Beispiel hierfür wären die Authentifizierungsmerkmale des angemeldeten Benutzers, die während der gesamten Lebensdauer der Applikation zur Verfügung stehen müssen.

Eine Instanz der Klasse SessionHandler steht ab seiner ersten Instanzierung beim Loginvorgang bis zu seiner Zerstörung beim Logoutvorgang den View- und Controller-Klassen zur Verfügung. Eine nähere Beschreibung was man unter View- und Controller-Klassen versteht, findet sich in den Abschnitten 4.3.4.5 und 4.3.5. Der SessionHandler instanziiert seinerseits die Klassen DatabaseHandler, ApplicationHandler und OptionsHandler.

Der DatabaseHandler hält alle Informationen zum Aufbau der Datenbankverbindung. Über den DatabaseHandler kann in der Applikation eine Verbindung zur Datenbank angefordert werden.

Der ApplicationHandler hält alle Informationen, die innerhalb der Applikation benötigt werden. Die Studiendokumentation ist ein Applikationszweig der Hauptapplikation. Unter einem Applikationszweig versteht man eine Teilapplikation die innerhalb der Gesamtapplikation ausgeführt wird. Diese Applikationszweige sind als Subklassen

des ApplicationHandler modelliert und implementiert. Eine Instanz dieser Subklassen (StudyHandler, WorkflowHandler) hält die Zustände der Gesamtapplikation und des jeweiligen Applikationszweiges. Die Instanz des StudyHandlers hält zusätzlich die Zustände der Studiendokumentation. Unter den Zuständen für die Studiendokumentation befindet sich ebenfalls eine Instanz des FormHandlers. Der FormHandler hält jene Zustände die nötig sind, um ein Dokument neu anzulegen, zu öffnen und zu speichern.

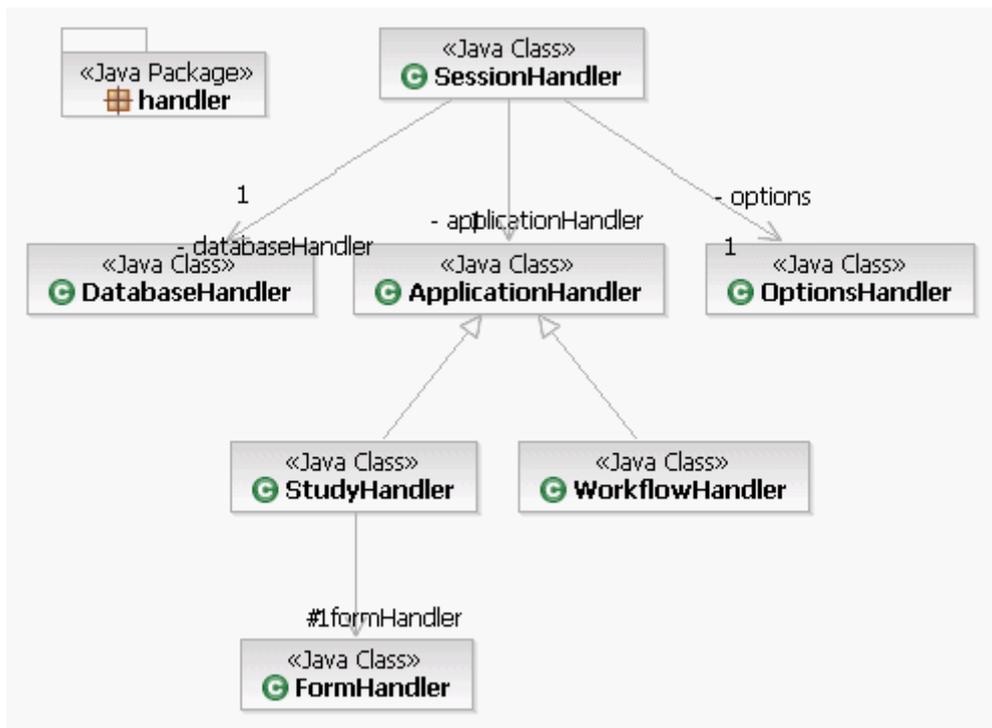


Abbildung 16: Paket handler und dessen Klassen.

#### 4.3.4.2 Paket database

Das Paket database (siehe Abbildung 17) beinhaltet die Modell-Klassen der Applikation. Das Modell enthält die darzustellenden Daten. Die Klassen entsprechen dem Datenmodell der Datenbank. In den Klassen werden die SQL Queries zur Datenbank bereitgestellt und ausgeführt.

#### 4.3.4.3 Paket generic

Das Paket generic beinhaltet ebenfalls Modell-Klassen. Im Gegensatz zu den Klassen des Paketes database, die das Datenmodell der Datenbanktabellen modellieren, bilden die Klassen des Paketes generic das Datenmodell zur Darstellung und Typisierung der in einem Web Formular (eCRF) befindlichen Ein-

und Ausgabefelder ab. Die Applikation unterstützt mehrere Typen von Feldern, siehe Abbildung 18.

Texteingabe:

Einfacheingabe über Textfelder siehe 1)

Einfacheingabe über Auswahllisten siehe 2)

Mehrfacheingabe über Auswahllisten siehe 3)

Einfacheingabe über Checkboxen siehe 4).

Numerische Eingabe:

Einfacheingabe über Textfelder siehe 5)

Einfacheingabe über Auswahllisten siehe 6)

Mehrfacheingabe über Auswahllisten siehe 7)

Datums Eingabe: Einfacheingabe über Textfelder siehe 8).

Des Weiteren gibt es Ausgabefelder siehe 9), Gruppen mit Hilfe derer man mehrere Ein und Ausgabefelder Glieder kann siehe 10), und Tabellen siehe 11). Dann gibt es noch das Formularblatt siehe 12) und Befehlsfelder siehe 13).

#### **4.3.4.4 Paket faces**

Das Paket faces (siehe Abbildung 19) beinhaltet Controller Klassen. Diese Klassen sind Hilfsklassen zur Steuerung der Abläufe innerhalb der Applikation. Die Steuerung der Benutzerinteraktion (Dateneingabe, Wechsel zu einer Webseite, Anmeldung, etc.) passiert in den Klassen des Paketes pagecode.

Die Klasse PhaseTracker überwacht den kontrollierten Seitenwechsel. Durch das Verwenden des Browser-Backbuttons kann es passieren, dass man in einen undefinierten Zustand der Applikation kommt, da die Benutzerinteraktion beim Betätigen des Browser Backbuttons nicht mit der Webapplikation stattgefunden hat sondern mit dem Browser. Daraufhin werden Informationen angezeigt die nicht mehr aktuell sein könnten. Die Korrektheit der Benutzerinteraktion wird über die Klasse PhaseTracker überwacht.

Die Klasse AuthenticationFilter bildet einen Sicherheitsmechanismus über die gesamte Applikation. Dieser Mechanismus stellt sicher, dass nur authentifizierte Benutzer die Seiten der Webapplikation aufrufen dürfen.

Die Klasse SafeActionListener dient zum Abfangen von Fehlern.

Der SafeNavigationHandler kontrolliert, ob ein authentifizierter Benutzer über ausreichend Rechte verfügt, um eine Funktion ausführen zu dürfen. Welche Funktionen ein Benutzer ausführen darf wird über die Datenbank abgefragt.

Die Funktionalität der Klasse Messages ist die Erzeugung von internationalisierten Textausgaben auf den Webseiten. Je nachdem welche Sprache im Browser als Standard eingestellt ist, oder welche Sprache (Englisch, Deutsch, Ausbau auf andere Sprachen möglich) der Benutzer in der Webapplikation auswählt, werden webapplikationsweit die Textausgaben auf den Webseiten gesetzt.

#### **4.3.4.5 Paket pagecode**

Das Paket pagecode (siehe Abbildung 20) beinhaltet die Controller-Klassen zur Steuerung der Benutzerinteraktionen. Diese Klassen stehen in Verbindung mit ihren jeweiligen GUI Webseiten (View). Die zur pagecode Klasse gehörende GUI Webseite hat den gleichen Namen plus der Endung „.jsp“.

Die Klasse TemplateUI stellt sämtliche Funktionalitäten bereit, die für den dynamischen Aufbau der Menüs innerhalb der Webapplikation nötig sind.

Die Klasse LoginUI ermöglicht dem Benutzer sich am System anzumelden. Je nachdem ob der Benutzer für die Webapplikation berechtigt ist, wird die Anfrage an eine Fehlerseite error.jsp oder an die Home-Seite homeUI.jsp weitergereicht.

Die Klasse HomeUI ermöglicht der homeUI.jsp Seite die Darstellung von Systeminformationen aus der Datenbank.

Die Klasse ChangePasswordUI ermöglicht dem Benutzer das Ändern seines Passwortes.

Die Klasse StudySelectorUI steuert die Anzeige der Liste Studien zu denen ein Benutzer berechtigt ist, empfängt die Benutzerauswahl der Studie und leitet den Benutzer auf die studyHandlerUI.jsp Seite weiter.

Die Klasse StudyHandlerUI steuert die Anzeige der Formulare, der Dokumente und der Studienpatienten. Des Weiteren übernimmt die StudyHandlerUI Klasse die Befehle der studyHandlerUI.jps Seite, einen neuen Patienten anzulegen, ein neues Dokument anzulegen, ein Dokument zu löschen und ein Dokument zu editieren.

Die Klasse FormUI steuert die Anzeige eines Dokuments. Je nach dem ob ein Dokument geöffnet oder ein neues angelegt wird, werden die Patientendaten in das auf der studyHandlerUI.jsp Seite ausgewählte Formular geladen. Nachdem der Benutzer das Dokument ausgefüllt und den Befehl zum Speichern erteilt hat,

übernimmt die FormUI Klasse die Daten in die Modell Klassen der Pakete generic und database und kümmert sich um die Speicherung.

Die Klasse PatientenEditorUI ist für die Steuerung der Anzeige und der Steuerung der Speicherung von Patientenstammdaten wie Geburtsdatum, Geschlecht, Kontakt etc. verantwortlich. Im Fall des Aufrufs der patientenEditorUI.jsp Seite zu einem bestimmten Patienten liefert die Klasse PatientenEditorUI die Patientenstammdaten einer Instanz der Klasse Studienpatient aus dem Paket database. Im Fall des Aufrufs ohne Patienten wird ein neuer Patient angelegt. Die Klasse PatientenEditorUI übernimmt, nachdem der Benutzer den Befehl Speichern abgeschickt hat, die Daten in die Modell Klassen und kümmert sich um die Speicherung.

Die Klasse ErrorUI ist für die Steuerung der Anzeige von Fehlermeldungen zuständig, die innerhalb der Webapplikation auftreten.

#### **4.3.4.6      Paket error**

Die Klassen des Pakets error (siehe Abbildung 21) beinhalten die Exceptions, die während der Laufzeit der Applikation abgefangen werden.

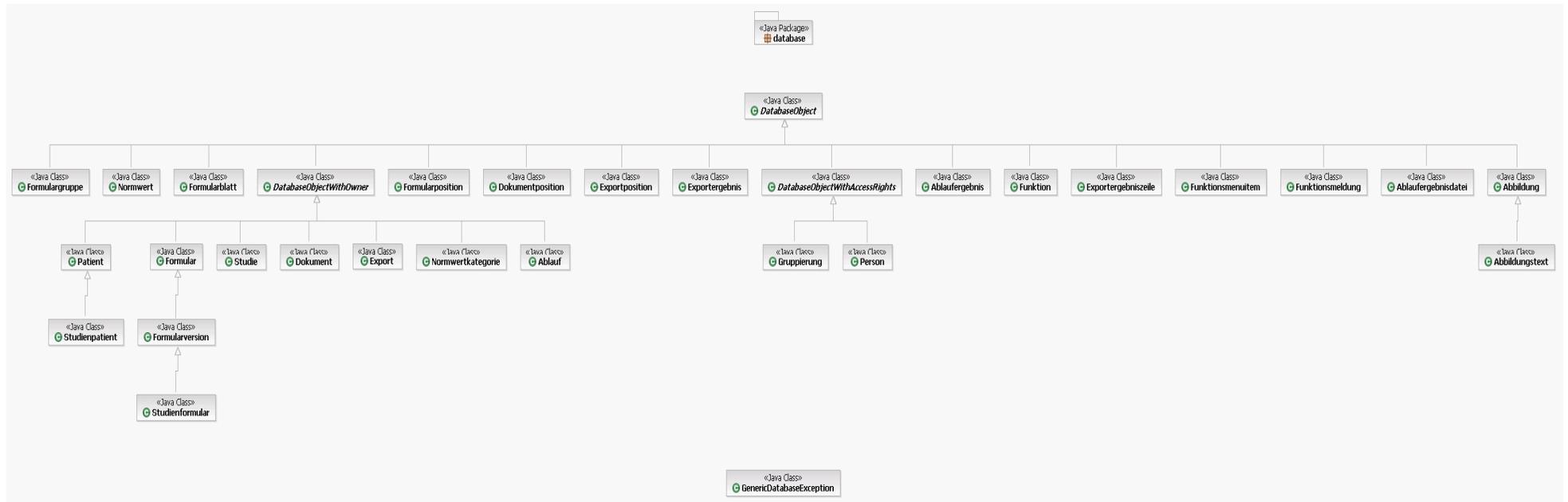


Abbildung 17: Paket database und dessen Klassen.

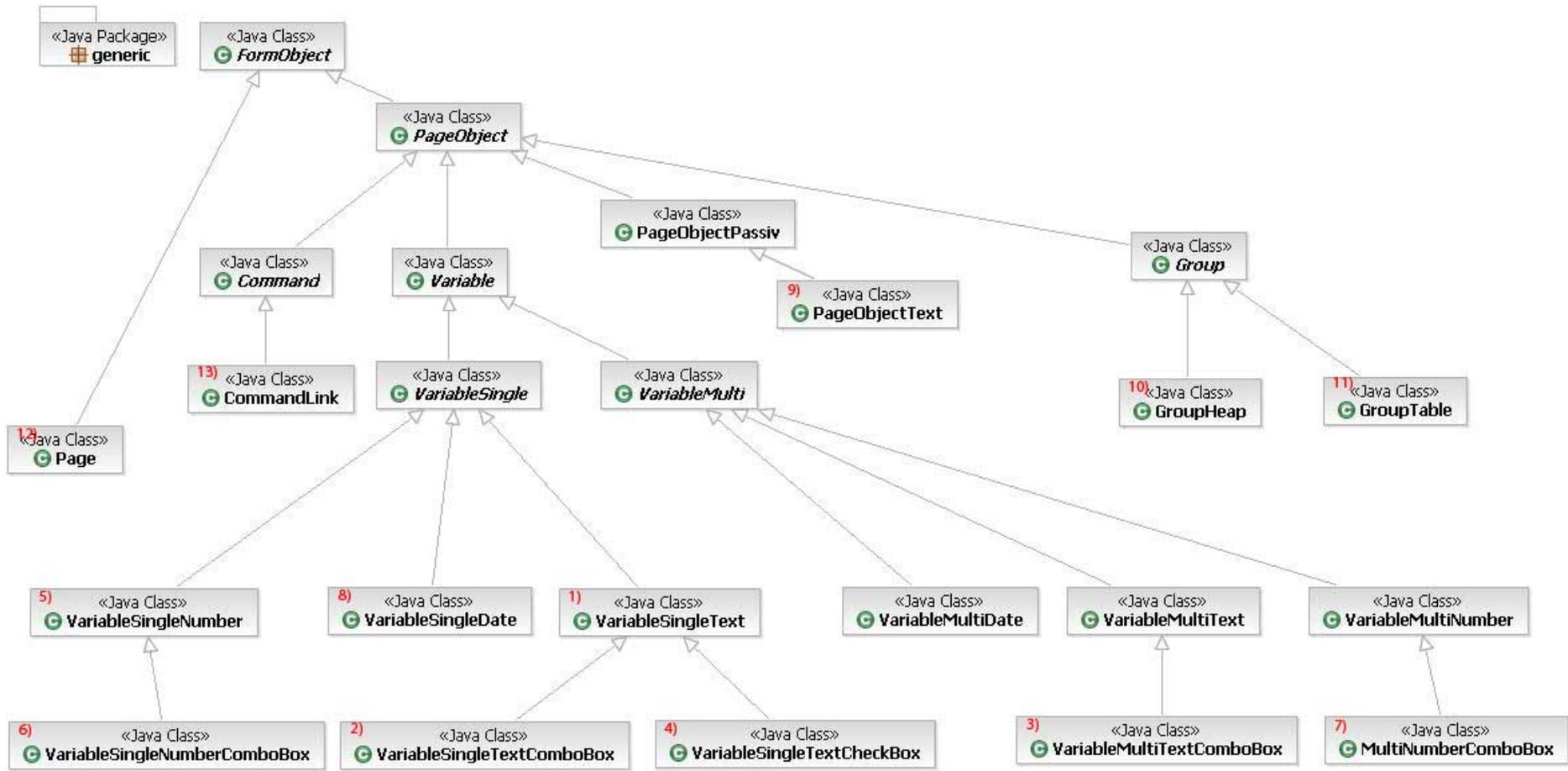


Abbildung 18: Paket generic und dessen Klassen.



Abbildung 19: Paket faces und dessen Klassen.

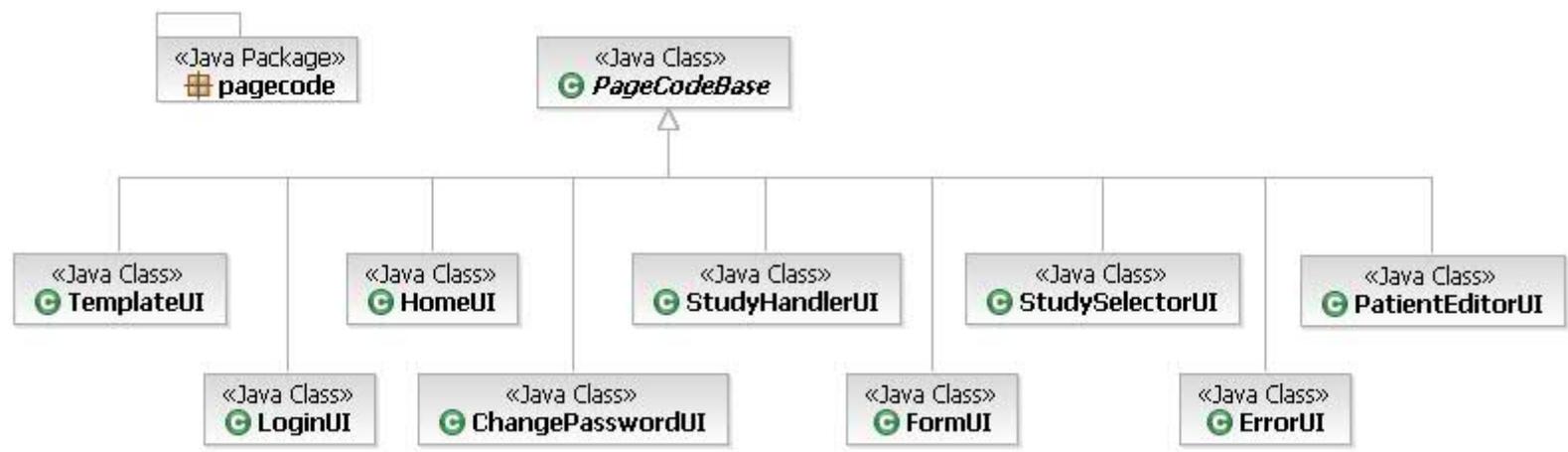


Abbildung 20: Paket pagecode und seine Klassen.



Abbildung 21: Paket error und seine Klassen

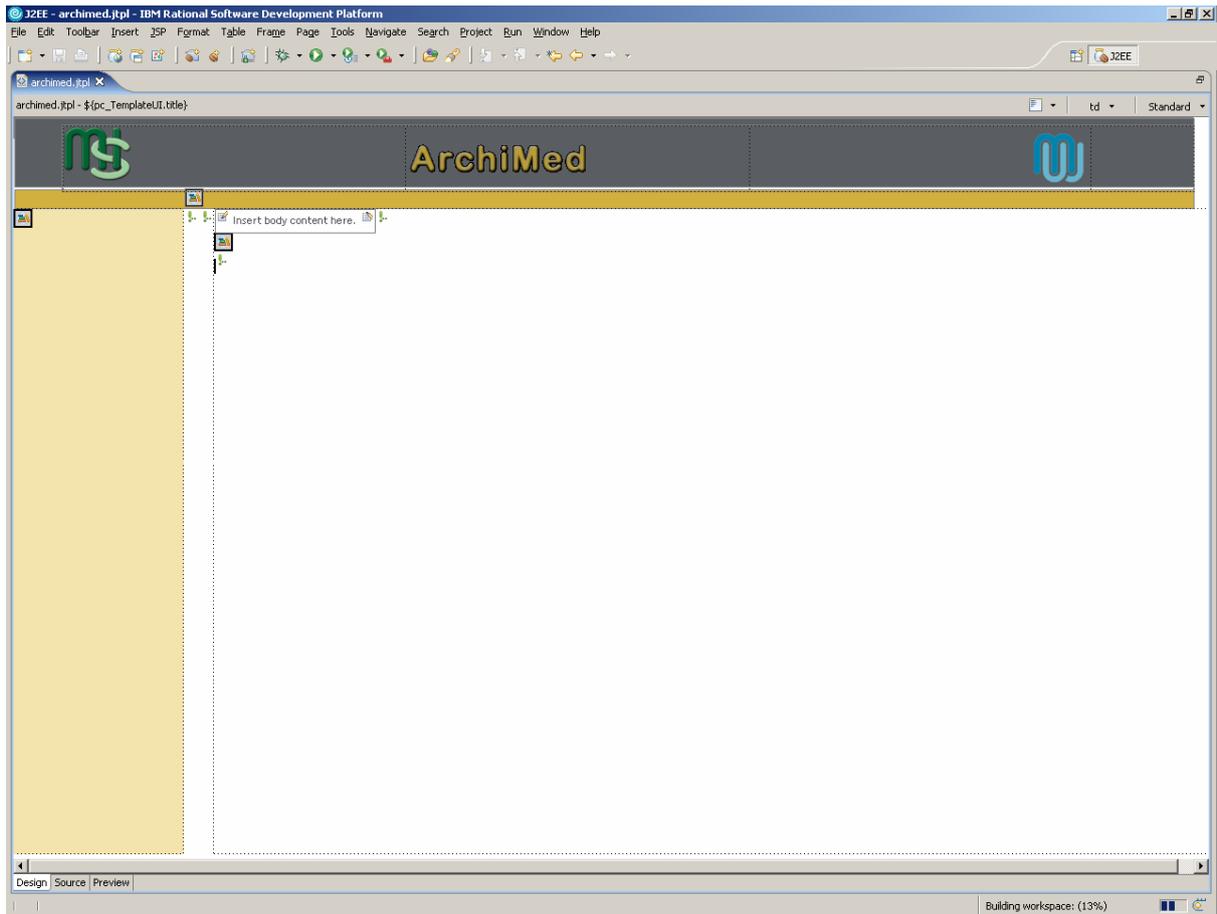
## 4.3.5 Implementierung der grafischen Benutzerschnittstelle

Bei der Implementierung der grafischen Benutzerschnittstelle soll auf die folgenden Punkte besonderer Wert gelegt werden:

1. Konsistenz der grafischen Benutzerschnittstelle: es wird eine Vorlage (*Template*) verwendet, damit alle Seiten das gleiche Layout haben. Die verwendeten Grafikobjekte haben durch die Verwendung von *Stylesheets* auf jeder Seite das gleiche Layout.
2. Namensrichtlinien: durch die Verwendung von zentralen *Ressource Bundles* werden Bezeichnungen an einer zentralen Stelle (in den Resource Bundles) festgelegt und über einen Zeiger zur Laufzeit an die betreffenden Stellen in die Applikation geladen (z.B. die Schaltfläche zur Durchführung des Speichervorganges heißt immer „Speichern“).
3. Meldungen an den Benutzer werden klar und unmissverständlich ausgegeben z.B. bei Falscheingabe eines numerischen Datenwertes wird angezeigt, wo der Fehler im Formular passiert ist und was für eine korrekte Eingabe erwartet wird.
4. Bei der Anordnung der Menüs und Befehlsschaltflächen wird darauf geachtet, dass sie ergonomisch gruppiert sind.

### 4.3.5.1 Vorlage – archimed.jtpl

Das Template archimed.jtpl dient als Vorlage aller GUI Seiten (siehe Abbildung 22). In dem weißen *Hauptbereich* werden die seitenspezifischen Informationen dargestellt. Im linken hellbraunen Bereich befindet sich das funktionale Menü welches abhängig von der angezeigten Seite gebildet wird. Im oberen dunkelbraunen Balken befindet sich das nicht funktionale applikationsweite Menü (Hilfe, Logout, etc.).



**Abbildung 22: Template der Seiten der Webapplikation in der Entwicklungsumgebung.**

#### **4.3.5.2 Login – loginUI.jsp**

Die loginUI.jsp Seite beschreibt die grafische Benutzerschnittstelle die angezeigt wird wenn der Benutzer die Webapplikation startet (siehe Abbildung 23). Hier werden über eine Instanz der zur loginUI.jsp Seite gehörenden LoginUI pagecode-Klasse die eingegebenen Authentifizierungsmerkmale (Benutzername und Passwort) an eine Instanz der Modell-Klasse gebunden. Durch Drücken der Login Schaltfläche wird der LoginUI Instanz der Befehl erteilt, die Authentifizierung durchzuführen. Dabei wird eine Verbindung mit der Datenbank hergestellt, und die Webapplikation versucht, sich mit den Authentifizierungsmerkmalen des Benutzers an der Datenbank anzumelden. Ist der Vorgang erfolgreich, leitet die Applikation den Benutzer auf die homeUI.jsp Seite weiter, andernfalls wird eine Fehlermeldung ausgegeben.

Des Weiteren befinden sich auf dieser Seite noch Textausgaben, die abhängig von der gewählten Sprache in Englisch oder Deutsch erscheinen.

Das funktionale Menü erscheint auf der loginUI.jsp Seite nicht, da der Benutzer noch nicht angemeldet ist und deshalb die Funktionen des Systems nicht verwenden darf.

Das nichtfunktionale Menü besteht aus den Punkten Hilfe und Englisch (bzw. Deutsch).

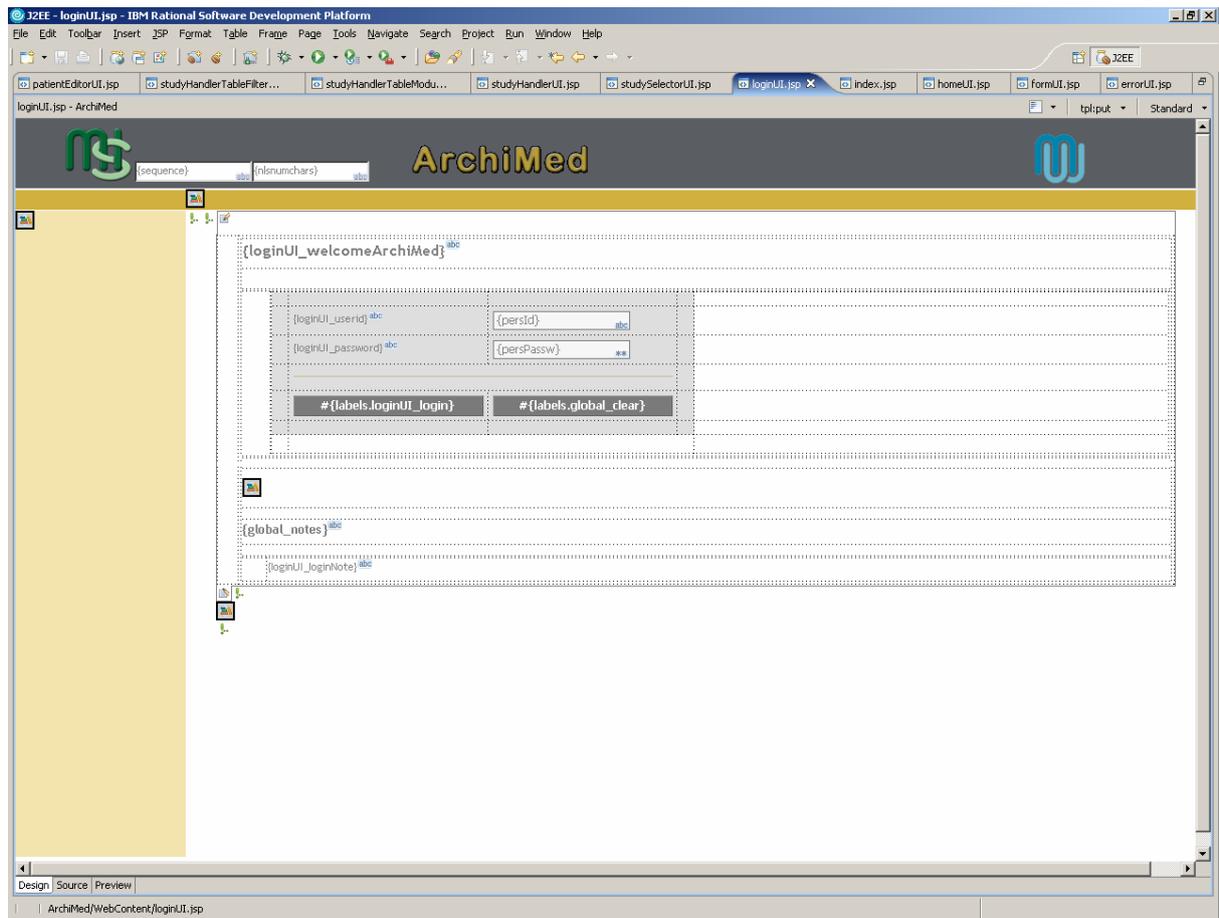
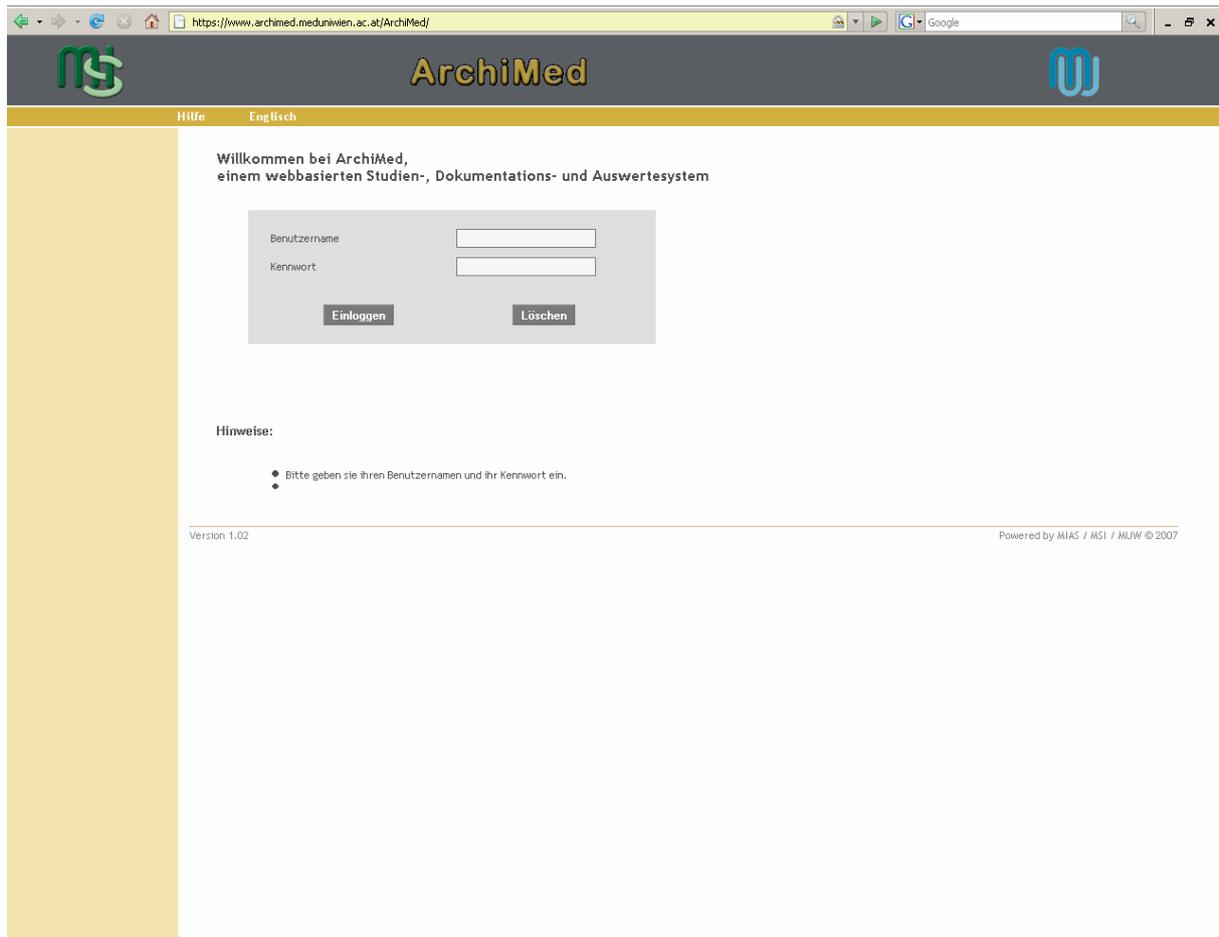


Abbildung 23: loginUI.jsp Seite in der Entwicklungsumgebung.

In Abbildung 24 sieht man, wie die loginUI.jsp Seite im Browser interpretiert wird.

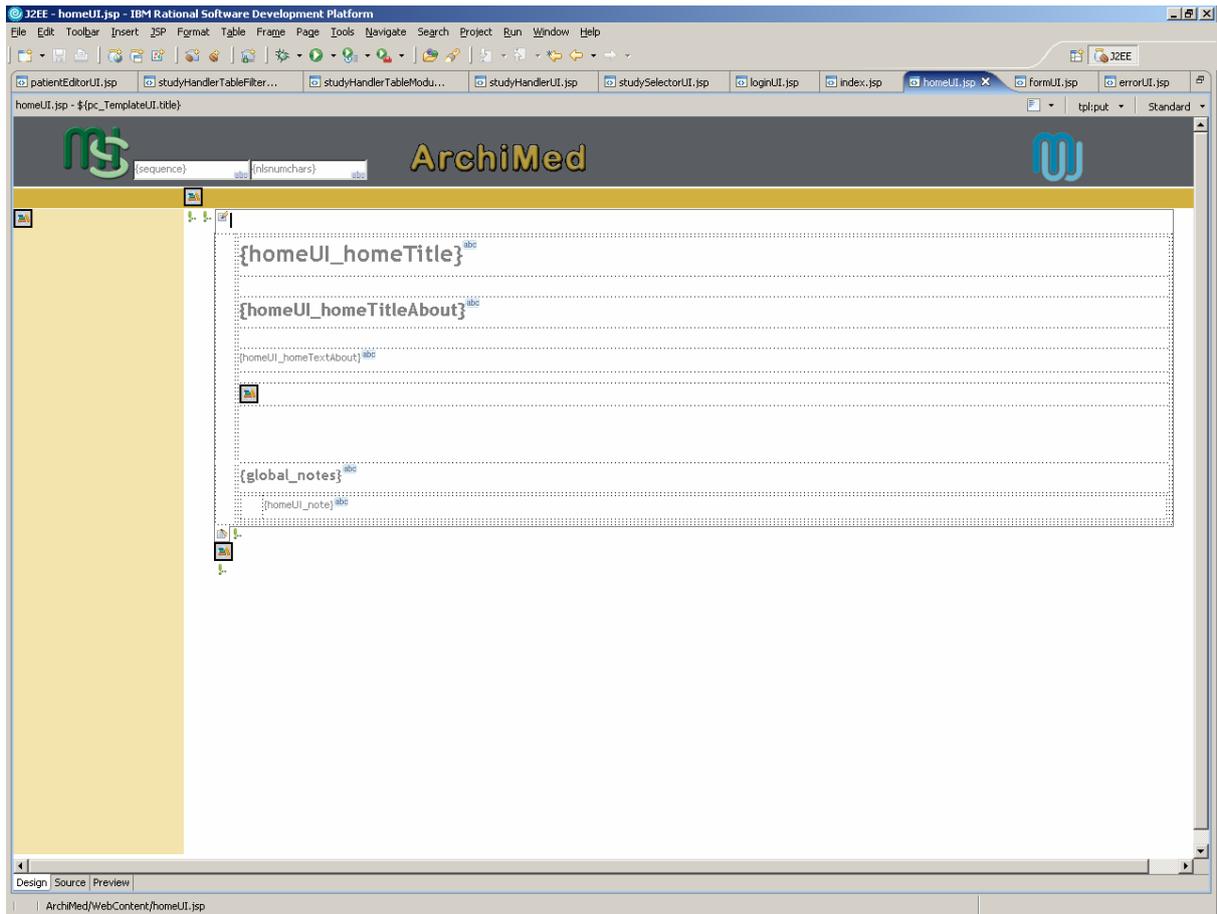


**Abbildung 24: Login Seite im Browser.**

#### **4.3.5.3 Hauptseite – homeUI.jsp**

Diese Seite bietet die Möglichkeit, aktuelle Systeminformationen auszugeben (siehe Abbildung 25). Ab hier bekommt der Benutzer das funktionale Menü angezeigt. Die Menüpunkte sind Hauptseite, Studienauswahl, Auswertung (auf diesen Menüpunkt wird in dieser Arbeit nicht näher eingegangen). Das nichtfunktionale Menü besteht aus den Punkten Hauptseite, Kennwortwechsel, Englisch, Hilfe, Beenden. Näheres zu den Ausprägungen des Menüs unter Präsentation des Systems. Die Seiteninhalte und die Bezeichnungen der Menüpunkte sind sprachabhängig.

Je nachdem für welchen Menüpunkt der Benutzer sich entscheidet, wird er zur betreffenden Seite weitergeleitet.



**Abbildung 25: homeUI.jsp Seite in der Entwicklungsumgebung.**

Die oben beschriebenen Ausprägungen der Menüs sind in Abbildung 26 gut zu sehen.

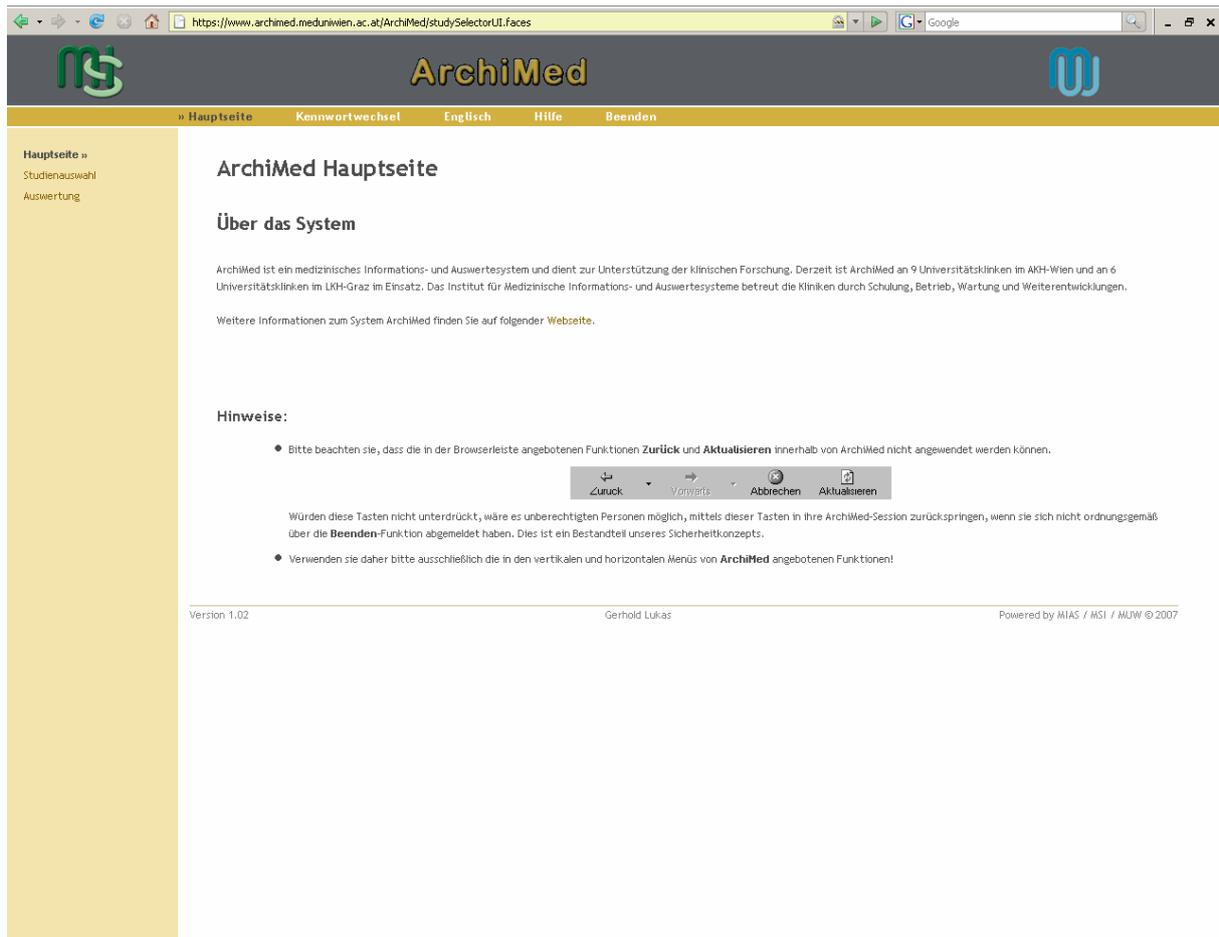
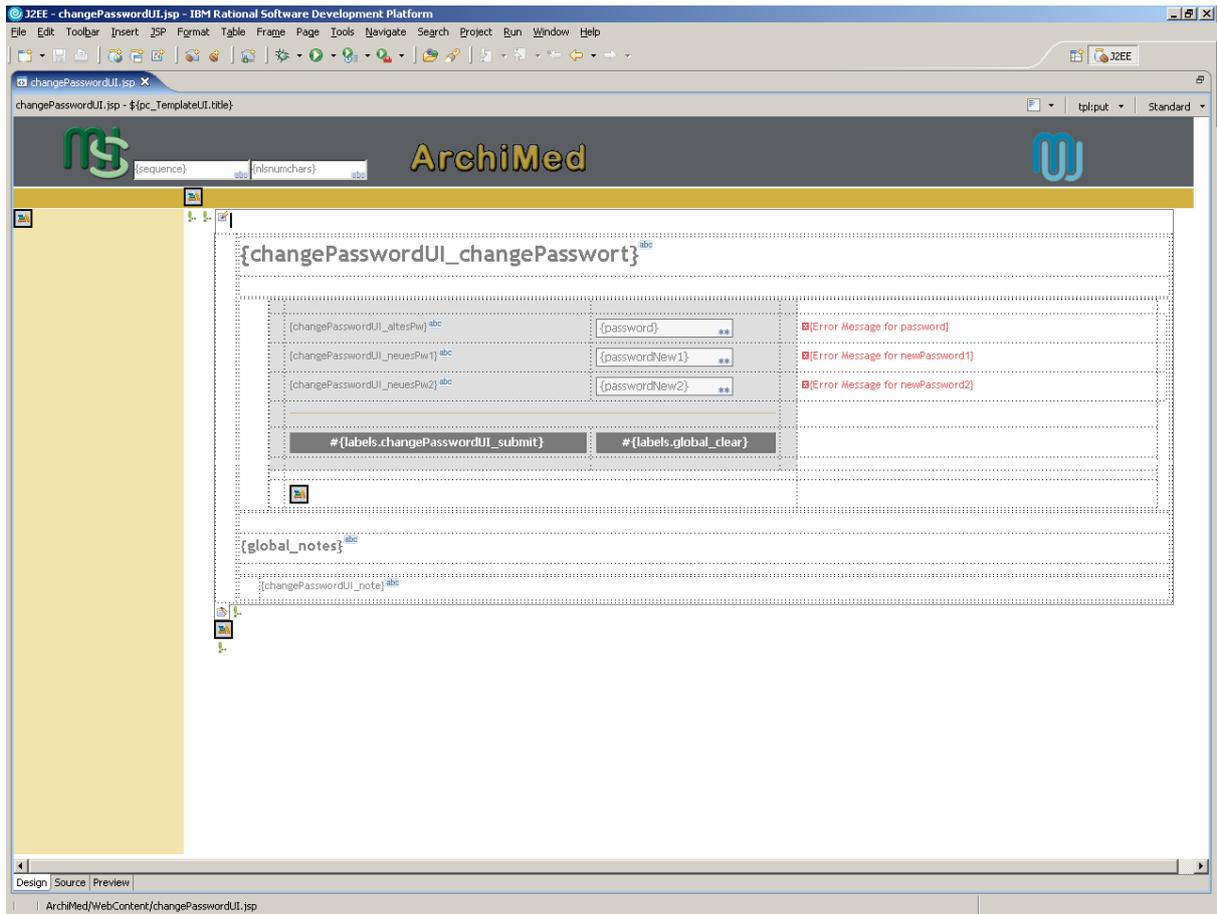


Abbildung 26: homeUI.jsp Seite interpretiert durch den Browser.

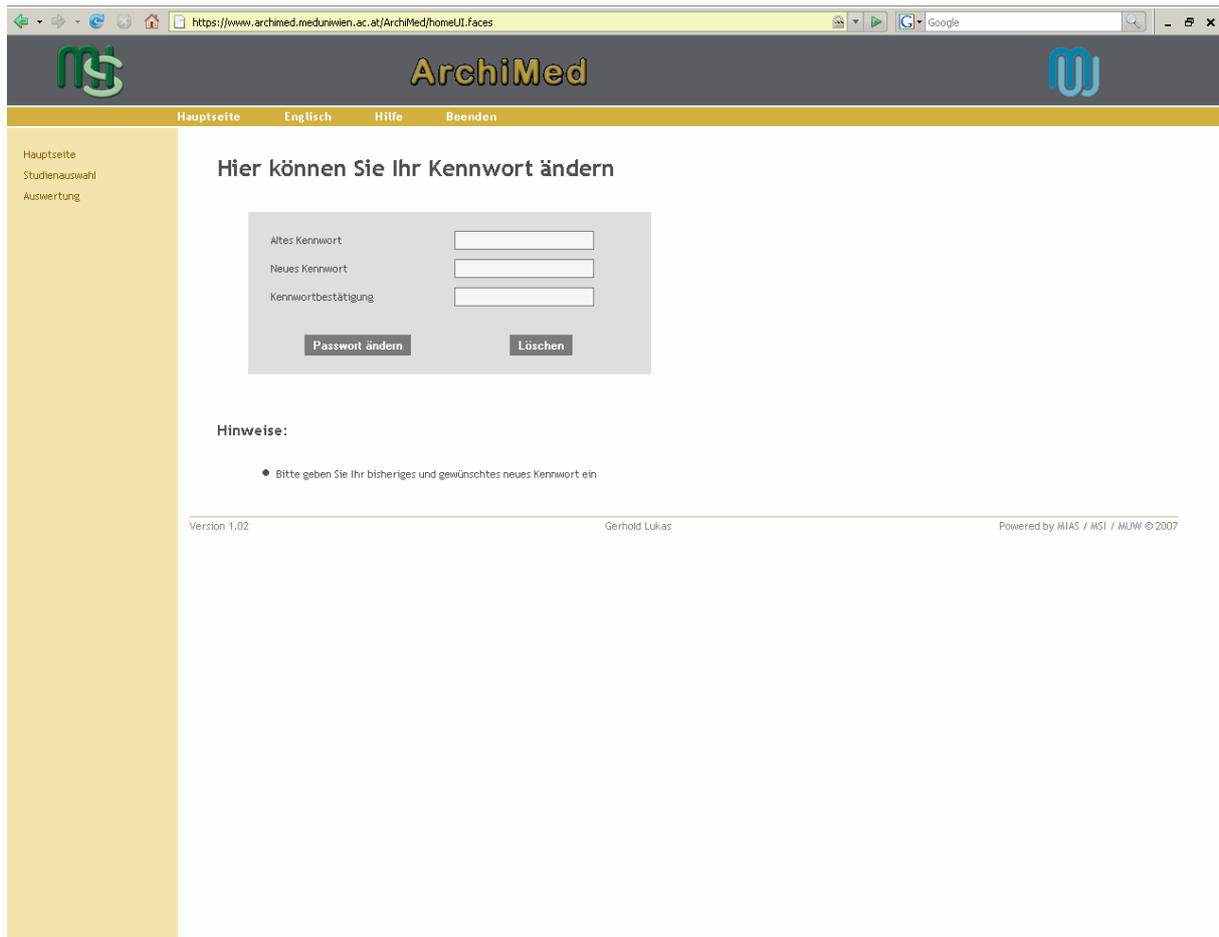
#### 4.3.5.4 Kennwort ändern – changePasswordUI.jsp

Durch Anklicken des Menüpunktes Kennwortwechsel des nichtfunktionalen Menüs, wird der Benutzer auf die changePasswordUI.jsp Seite weitergeleitet (siehe Abbildung 27). Hier wird er aufgefordert sein altes Kennwort und 2-mal sein neues Kennwort einzugeben (1-mal zur Überprüfung von Tippfehlern). Nachdem der Benutzer die Schaltfläche „Passwort ändern“ angeklickt hat werden die eingegebenen Daten zuerst in einer Instanz der Klasse ChangePasswordUI geprüft. Geprüft werden die Übereinstimmung der 2-mal eingegebenen neuen Kennwörter und ob das Kennwort aus Buchstaben und Zahlen besteht. Anschließend wird das neue Kennwort an eine Instanz der Klasse Person des Paketes database gebunden und in die Datenbank gespeichert.



**Abbildung 27: changePasswordUI.jsp Seite in der Entwicklungsumgebung.**

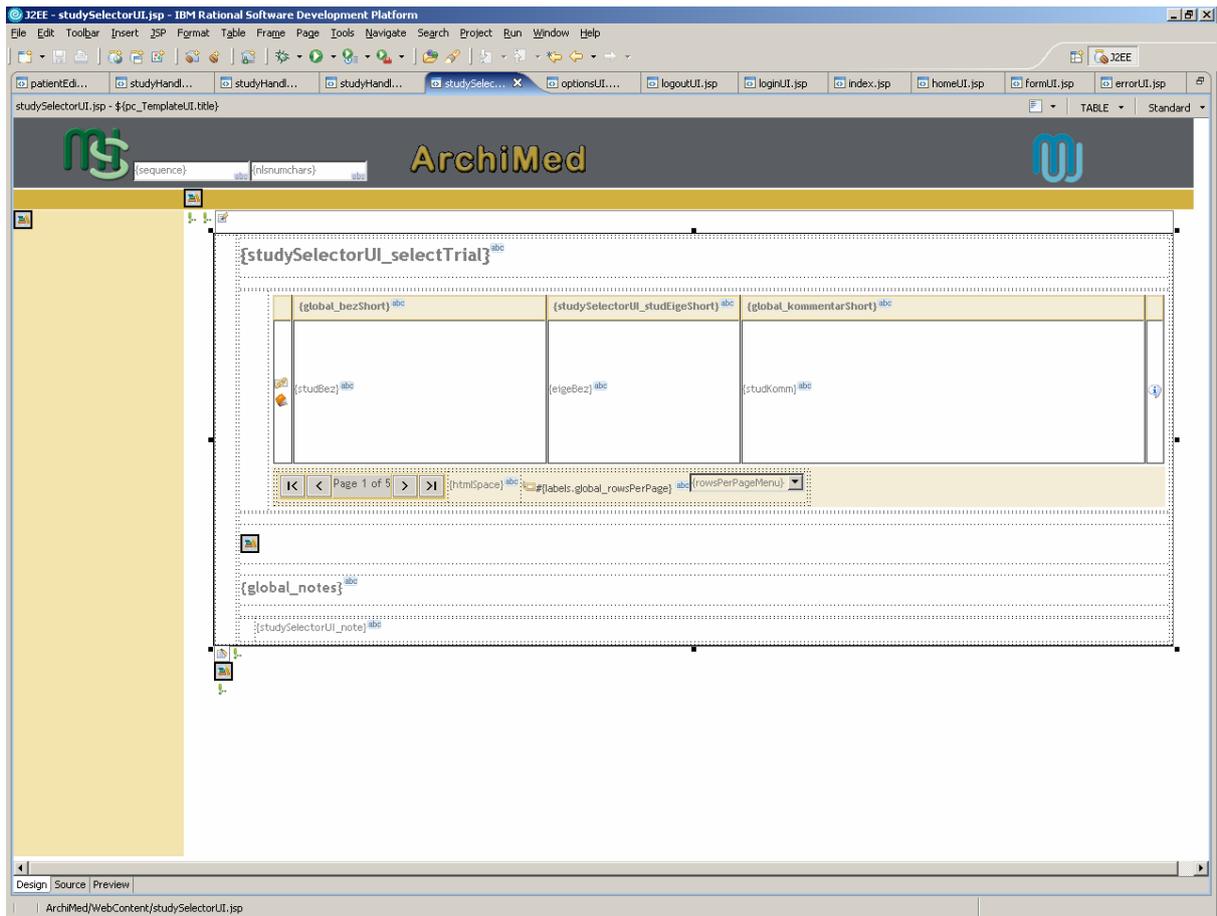
In Abbildung 28 sieht man die browserseitige Darstellung der changePasswordUI.jsp Seite, mit den Ausprägungen der deutschen Bildschirmausgaben, sowie den Eingabefeldern zum Wechseln des Kennwortes.



**Abbildung 28: changePasswordUI.jsp Seite wie sie im Browser interpretiert wird.**

#### **4.3.5.5 Studienauswahl – studySelectorUI.jsp**

Über den Menüpunkt Studienauswahl des funktionalen Menüs gelangt man auf die studySelectorUI.jsp Seite (siehe Abbildung 29). An dieser Stelle wird über eine Instanz der pagecode-Klasse StudySelectorUI die Studienliste einer Instanz der database-Klasse Studie angezeigt. Die Studienliste der Instanz der Klasse Studie beinhaltet nur jene Studien zu denen der Benutzer berechtigt ist. Somit wird je nachdem welcher Benutzer gerade angemeldet ist sichergestellt, dass er nur jene Studien auf der Seite angezeigt bekommt, an denen er auch teilnehmen darf. Beim Klicken auf eine Studie öffnet sich die Studiendokumentation – studyHandlerUI.jsp Seite.



**Abbildung 29: studySelectorUI.jsp Seite in der Entwicklungsumgebung.**

In Abbildung 30 sieht man wie die `studySelectorUI.jsp`-Seite im Browser angezeigt wird. Im weißen Hauptbereich befinden sich die sprachabhängigen Bildschirmausgaben wie z.B. „Wählen sie eine Studie“ und die Studienliste mit ihren Attributen Bezeichnung, Eigentümer und Kommentar.

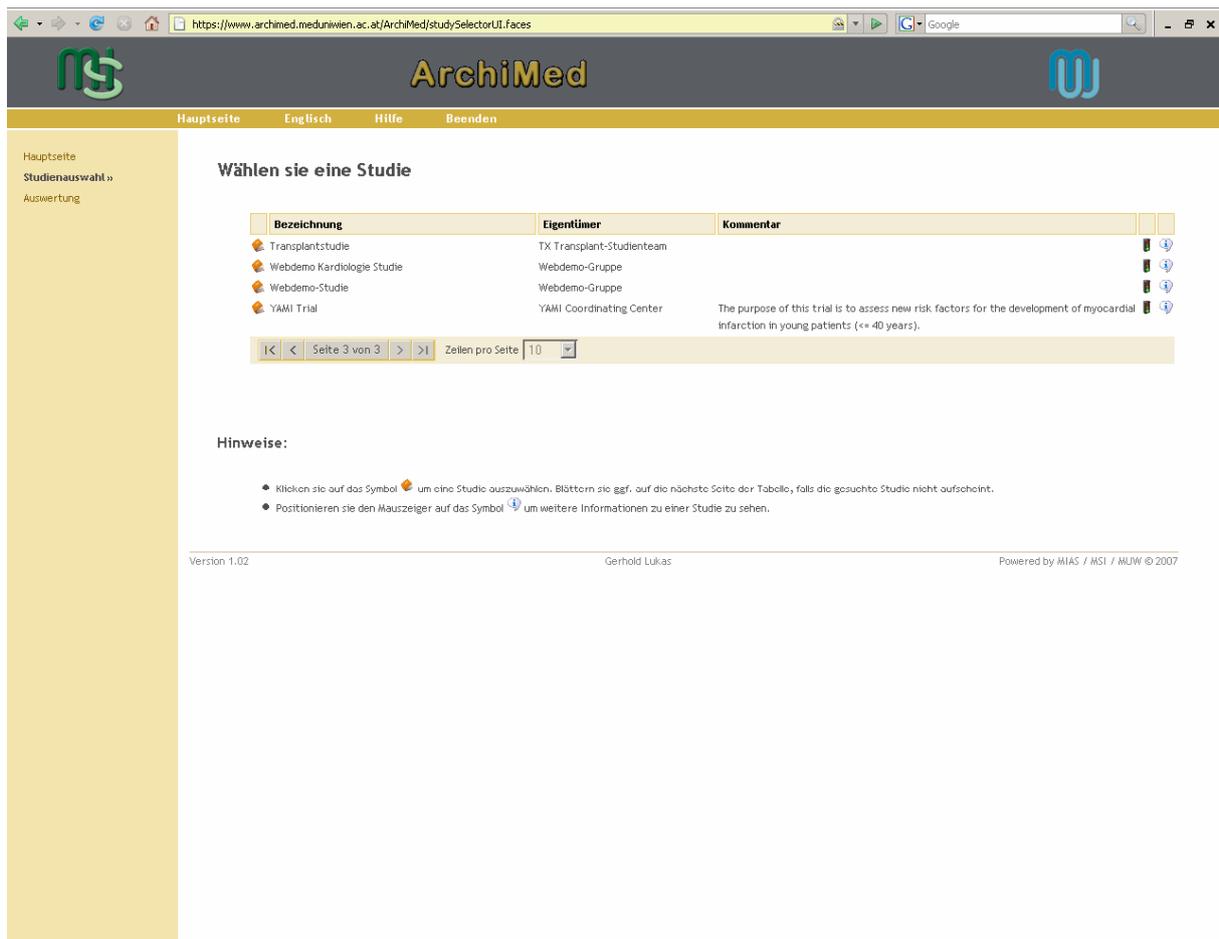


Abbildung 30: browserseitige Interpretation der studySelectorUI.jsp Seite.

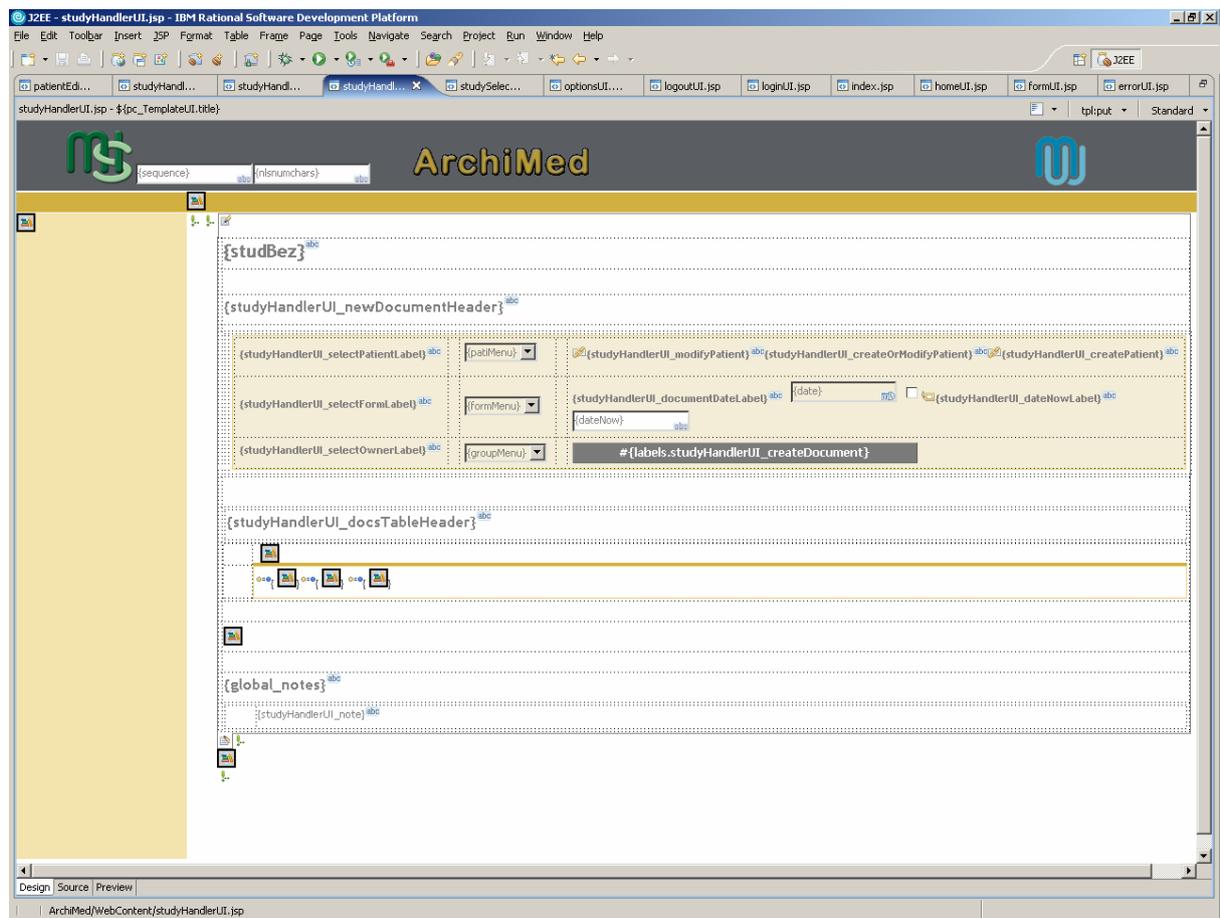
#### 4.3.5.6 Studiendokumentation – studyHandlerUI.jsp

Über das Anklicken einer Studie in der studySelectorUI.jsp-Seite gelangt man auf die studyHandlerUI.jsp-Seite (siehe Abbildung 31). Hier werden vor dem Aufbau der Seite über eine Instanz der pagecode-Klasse StudyHandlerUI die database-Klassen Studienpatient Dokument und Studienformular instanziiert. Die Daten dieser Instanzen werden dann zum Zeitpunkt des Aufbaus der studyHandlerUI.jsp-Seite in die Seite geladen und stehen dort in Form von Patientenlisten, Formularlisten und Dokumentenlisten zur Verfügung.

Durch Auswahl eines Patienten, eines Formulars und durch Klicken auf die Schaltfläche „Dokument anlegen“ werden die ausgewählten Informationen einer Instanz der StudyHandlerUI-Klasse übergeben. Diese Instanz kümmert sich darum, dass die ausgewählten Informationen in einer Instanz der StudyHandler-Klasse des handler-Paketes gespeichert werden und leitet den Benutzer auf die formUI.jsp-Seite weiter.

Sehr ähnlich funktioniert der Aufruf eines bestehenden Dokumentes. Hier wird die Information der Auswahl des Dokumentes nach dem Klicken auf ein Dokument an die Instanz der SudyHandlerUI-Klasse weitergeleitet, welche diese Information der Instanz der StudyHandler-Klasse (Paket handler) weiterreicht und den Aufruf der Seite formUI.jsp ausführt.

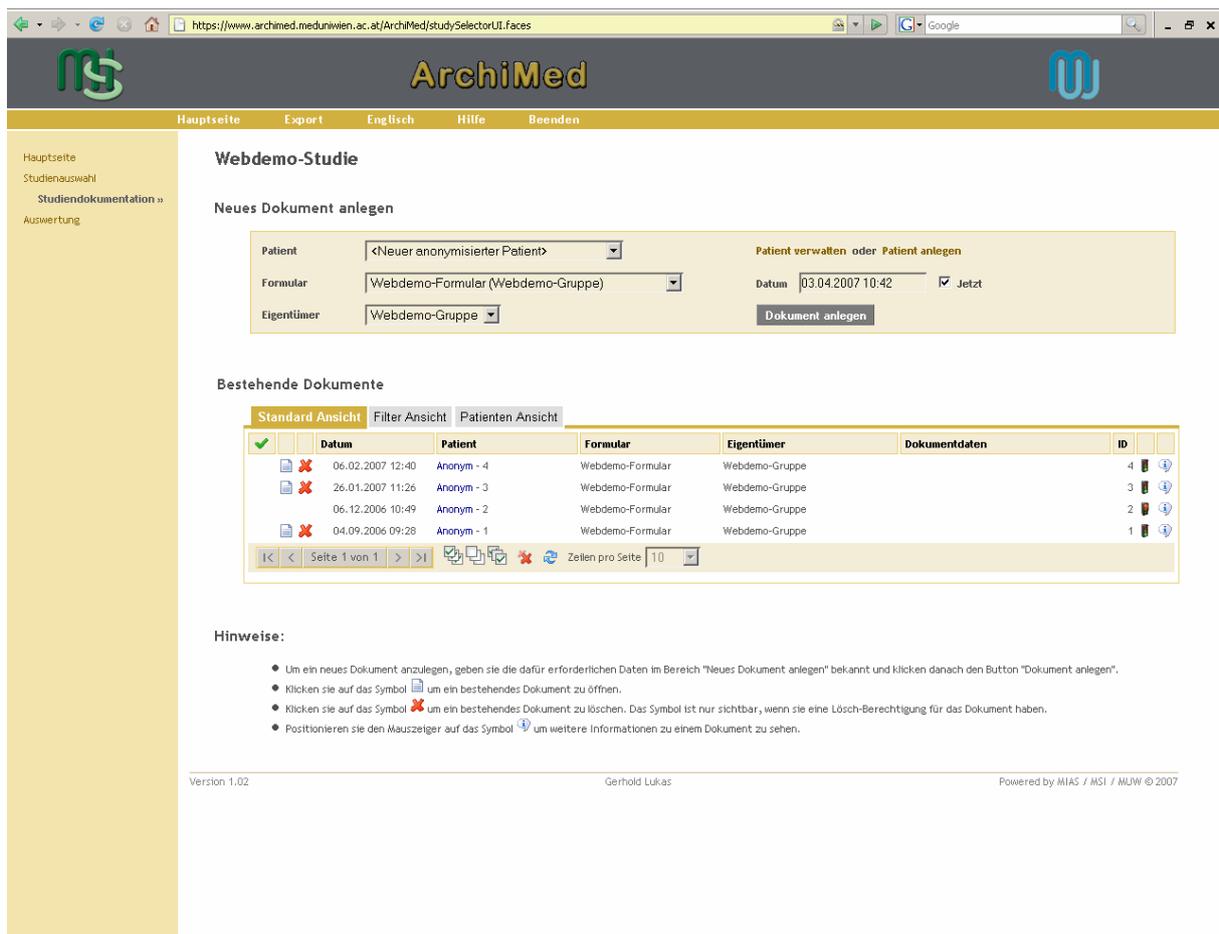
Zum Löschen eines Dokumentes klickt man in der Dokumentenliste auf den Befehl Löschen wodurch in der Instanz der Klasse StudyHandlerUI über die Instanz der Klasse Dokument eine SQL DELETE-Anweisung mit der Identifikation des Dokumentes ausgeführt und an die Datenbank geschickt wird.



**Abbildung 31: studyHandlerUI.jsp Seite in der Entwicklungsumgebung.**

In Abbildung 32 sieht man wie die studyHandlerUI.jsp Seite im Browser interpretiert wird. Im weißen Hauptbereich der Applikation befinden sich wiederum die sprachabhängigen Bildschirmausgaben wie „Webdemo-Studie“ als Bezeichner der aktuell geöffneten Studie, „Neues Dokument anlegen“, „Bestehende Dokumente“, etc. Unterhalb des Schriftzuges „Neues Dokument anlegen“ befindet sich der braune Header des weißen Hauptbereiches. Hier kann man durch vorherige Auswahl des

Patienten, des Formulars, des Eigentümers des Formulars, sowie des Dokumentendatums über die Schaltfläche „Dokument anlegen“, ein neues Dokument anlegen. Unterhalb des Schriftzuges „Bestehende Dokumente“ befindet sich die Liste der zu der Studie bestehenden Dokumente (siehe Abbildung 32). Es gibt 3 verschiedene Ansichten der Dokumentenliste. Die Attribute der Dokumentenliste sind Datum, Patient, Formularname, Eigentümer, Dokumentendatum und Identifikation des Dokuments. Die „Standard Ansicht“ sortiert die Dokumente umgekehrt chronologisch nach dem Dokumentendatum. Die „Filter Ansicht“ bietet die Möglichkeit die Dokumente nach den Ausprägungen der Attribute zu filtern (siehe Abbildung 33), z.B. wenn Patient „Anonym 1“ im Header ausgewählt wird, wird nur der in der Liste angezeigt. Die „Patienten-Ansicht“ ermöglicht das Gruppieren der Dokumente zu einem Patient (siehe Abbildung 34). Durch Klicken auf die Symbole , , kann ein Dokument aus der Liste zum Editieren geöffnet oder gelöscht werden.



**Webdemo-Studie**

Neues Dokument anlegen

Patient: <Neuer anonymisierter Patient> Patient verwalten oder Patient anlegen

Formular: Webdemo-Formular (Webdemo-Gruppe) Datum: 03.04.2007 10:42  Jetzt

Eigentümer: Webdemo-Gruppe Dokument anlegen

Bestehende Dokumente

Standard Ansicht	Filter Ansicht	Patienten Ansicht			
Datum	Patient	Formular	Eigentümer	Dokumentendaten	ID
  06.02.2007 12:40	Anonym - 4	Webdemo-Formular	Webdemo-Gruppe		4 
  26.01.2007 11:26	Anonym - 3	Webdemo-Formular	Webdemo-Gruppe		3 
  06.12.2006 10:49	Anonym - 2	Webdemo-Formular	Webdemo-Gruppe		2 
  04.09.2006 09:28	Anonym - 1	Webdemo-Formular	Webdemo-Gruppe		1 

Hinweise:

- Um ein neues Dokument anzulegen, geben sie die dafür erforderlichen Daten im Bereich "Neues Dokument anlegen" bekannt und klicken danach den Button "Dokument anlegen".
- Klicken sie auf das Symbol  um ein bestehendes Dokument zu öffnen.
- Klicken sie auf das Symbol  um ein bestehendes Dokument zu löschen. Das Symbol ist nur sichtbar, wenn sie eine Löscht-Berechtigung für das Dokument haben.
- Positionieren sie den Mauszeiger auf das Symbol  um weitere Informationen zu einem Dokument zu sehen.

Version 1.02 Gerhold Lukas Powered by MIAS / MSI / MUJW © 2007

Abbildung 32: studyHandler.jsp Seite mit der Standarddokumentenansicht im Browser.

https://www.archimed.meduniwien.ac.at/ArchiMed/studyHandlerUI.faces

**ArchiMed**

Hauptseite Export Englisch Hilfe Beenden

Hauptseite  
Studienauswahl  
Studiendokumentation »  
Auswertung

### Webdemo-Studie

#### Neues Dokument anlegen

Patient: <Neuer anonymisierter Patient> **Patient verwalten oder Patient anlegen**

Formular: Webdemo-Formular (Webdemo-Gruppe) Datum: 17.04.2007 11:43  Jetzt

Eigentümer: Webdemo-Gruppe **Dokument anlegen**

#### Bestehende Dokumente

Standard Ansicht **Filter Ansicht** Patienten Ansicht

<input checked="" type="checkbox"/>	<Datum>	Anonym - 1	<Formular>	<Eigentümer>	<Dokument Info>	ID
<input checked="" type="checkbox"/>	04.09.2006 09:28	Anonym - 1	Webdemo-Formular	Webdemo-Gruppe		1

Seite 1 von 1 | Zellen pro Seite: 10 | Filterwirkung umstellen: Irgendein Filter

#### Hinweise:

- Um ein neues Dokument anzulegen, geben sie die dafür erforderlichen Daten im Bereich "Neues Dokument anlegen" bekannt und klicken danach den Button "Dokument anlegen".
- Klicken sie auf das Symbol  um ein bestehendes Dokument zu öffnen.
- Klicken sie auf das Symbol  um ein bestehendes Dokument zu löschen. Das Symbol ist nur sichtbar, wenn sie eine Lösch-Berechtigung für das Dokument haben.
- Positionieren sie den Mauszeiger auf das Symbol  um weitere Informationen zu einem Dokument zu sehen.

Version 1.02 Gerhold Lukas Powered by MIAS / MSI / MJW © 2007

Abbildung 33: studyHandler.jsp Seite mit der Filteransicht der Dokumente im Browser.

**Webdemo-Studie**

**Neues Dokument anlegen**

Patient: <Neuer anonymisierter Patient> Patient verwalten oder Patient anlegen

Formular: Webdemo-Formular (Webdemo-Gruppe) Datum: 17.04.2007 11:46  Jetzt

Eigentümer: Webdemo-Gruppe Dokument anlegen

**Bestehende Dokumente**

Standard Ansicht Filter Ansicht **Patienten Ansicht**

	Datum	Patient	Formular	Eigentümer	Dokumentdaten	ID
▼	04.09.2006 09:28	Anonym - 1	Webdemo-Formular	Webdemo-Gruppe		1
▶						
▶						
▶						

Seite 1 von 1 Zeilen pro Seite: 10

**Hinweise:**

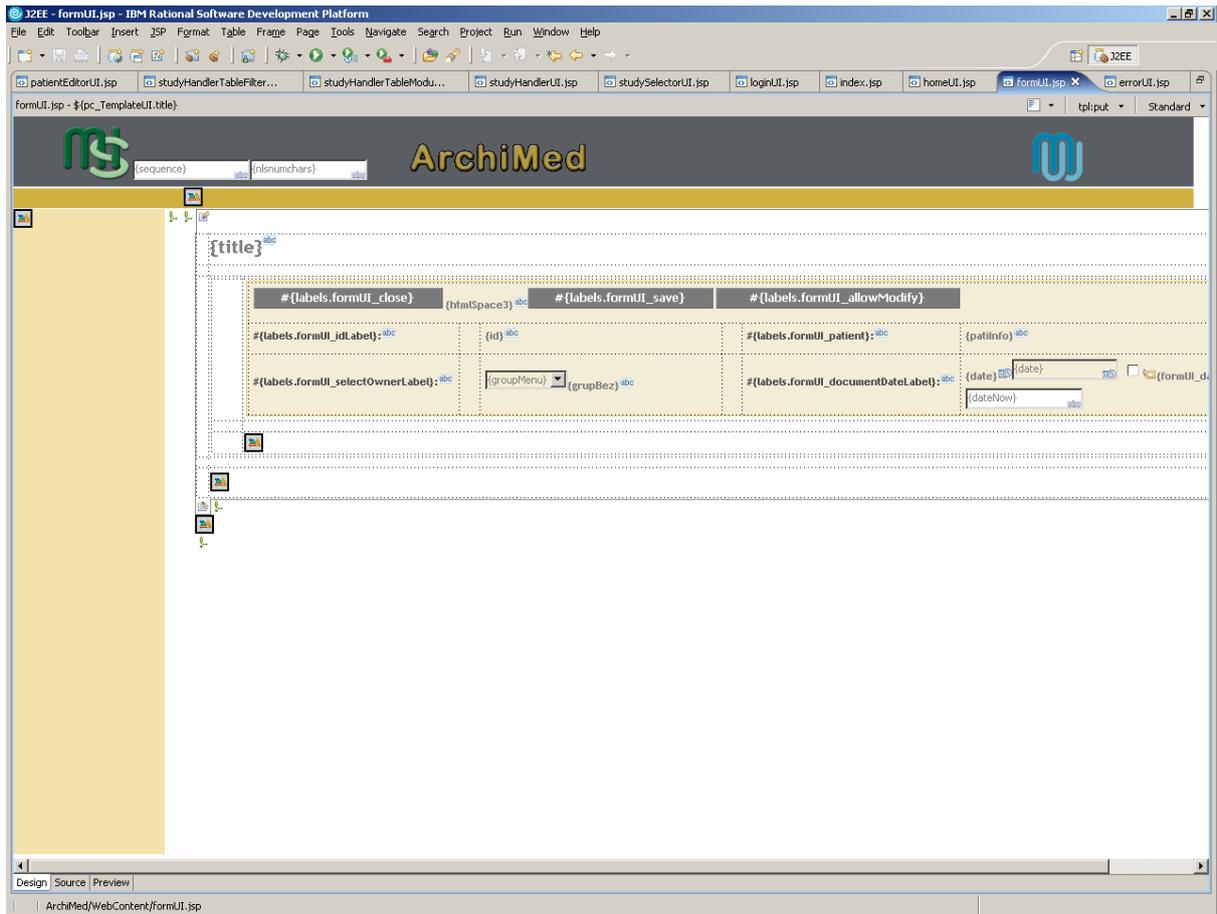
- Um ein neues Dokument anzulegen, geben sie die dafür erforderlichen Daten im Bereich "Neues Dokument anlegen" bekannt und klicken danach den Button "Dokument anlegen".
- Klicken sie auf das Symbol um ein bestehendes Dokument zu öffnen.
- Klicken sie auf das Symbol um ein bestehendes Dokument zu löschen. Das Symbol ist nur sichtbar, wenn sie eine Lösch-Berechtigung für das Dokument haben.
- Positionieren sie den Mauszeiger auf das Symbol um weitere Informationen zu einem Dokument zu sehen.

Version 1.02 Gerhold Lukas Powered by MIAS / MSI / MUW © 2007

**Abbildung 34: studyHandler.jsp Seite mit der patientenorientierten Dokumentenansicht im Browser.**

#### 4.3.5.7 Dokument (eCRF) – formUI.jsp

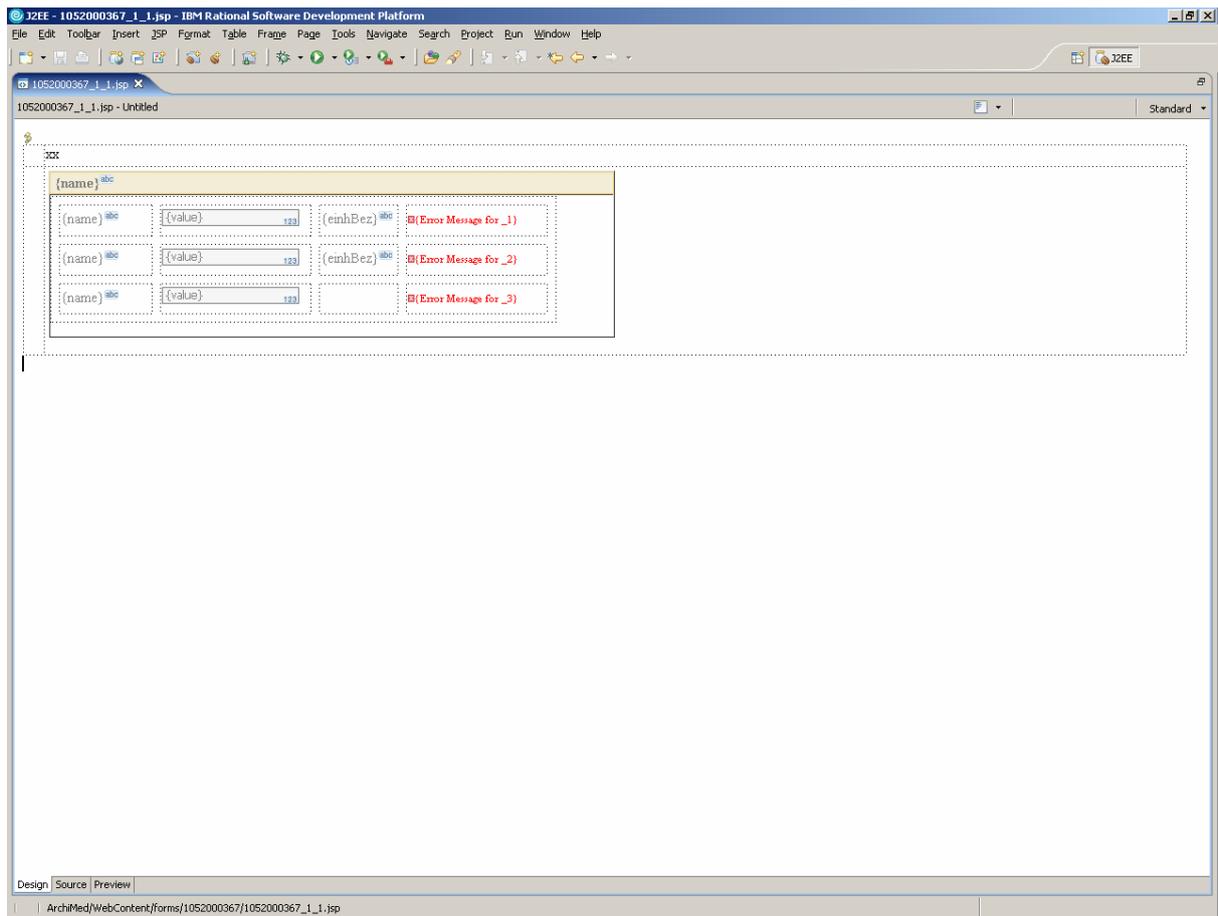
Nach dem Neuanlegen oder Öffnen eines Dokumentes wird der Benutzer an die formUI.jsp-Seite weitergeleitet (siehe Abbildung 35). Hier wird abhängig davon, welches Formular ausgewählt wurde, das betreffende Formular im weißen Hauptbereich der Seite unter dem orangebraunen Header geladen. Ein Beispiel eines Formulars zeigt die Abbildung 36.



**Abbildung 35: formUI.jsp Seite in der Entwicklungsumgebung.**

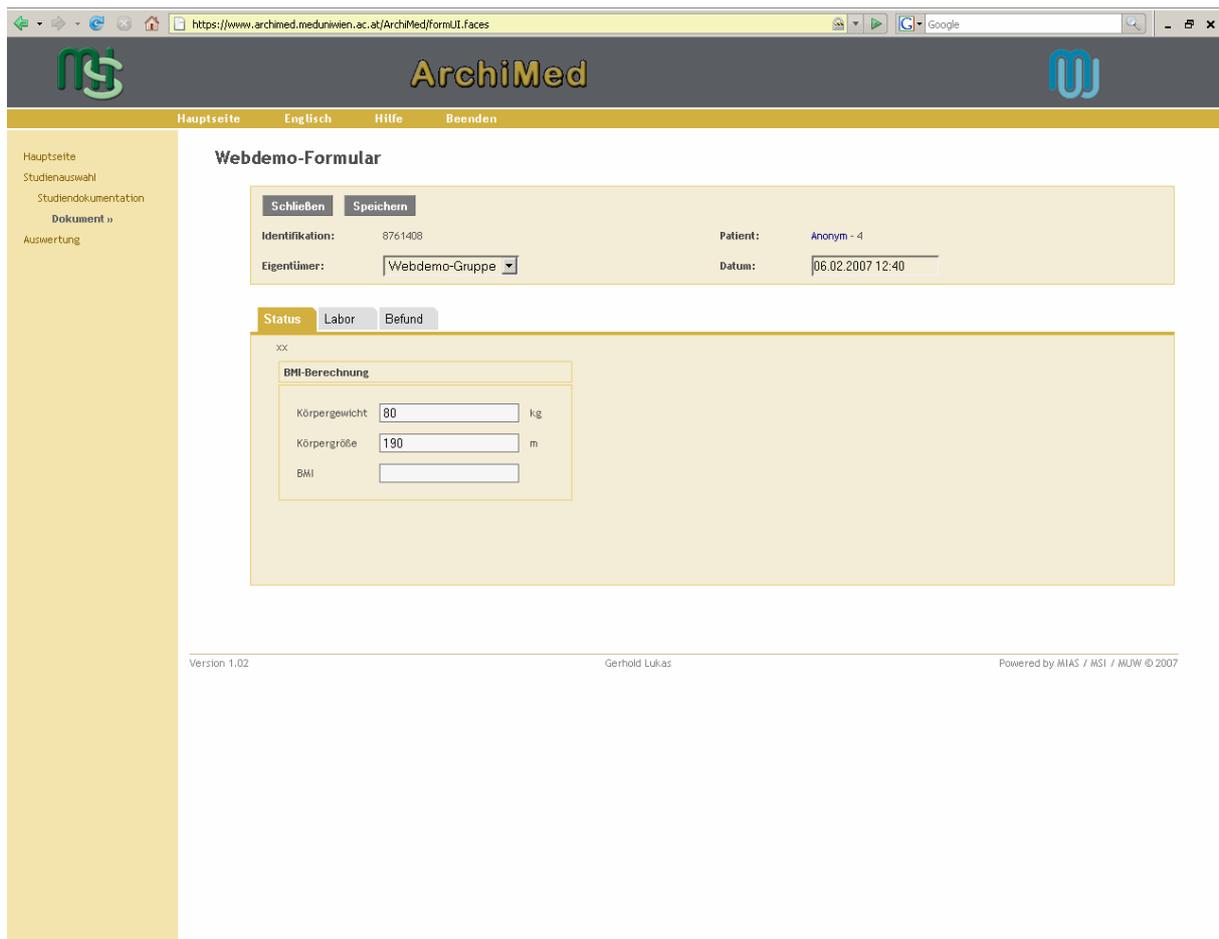
Beim Öffnen eines Dokumentes werden über eine Instanz der pagecode-Klasse FormUI die entsprechenden Patienteninformationen durch eine Instanz der database-Klasse Dokumentposition geladen. Anschließend wird in der formUI.jsp-Seite das Formular (siehe Abbildung 36) geladen, mit den Patienteninformationen befüllt und als Dokument angezeigt. Analog dazu funktioniert das Neuanlegen eines Dokumentes, wobei hier keine Patienteninformationen der Instanz der Klasse Dokumentenpositionen geladen werden müssen.

Nach dem Befüllen eines Dokumentes werden die Patienteninformationen über eine Instanz der FormUI-Klasse an eine Instanz der Dokumentenposition Klasse übergeben und beim Anklicken der „Speichern“-Schaltfläche wird der SQL-Befehl INSERT oder UPDATE (je nachdem ob ein Dokument neu angelegt oder zum Editieren geöffnet wird) an die Datenbank geschickt.



**Abbildung 36: Eingabefelder eines Formularblattes, das in die formUI.jsp Seite geladen wird.**

Abbildung 37 zeigt ein Dokument mit mehreren Blättern (Status, Labor, Befund). Im weißen Hauptbereich der Seite sieht man unterhalb des Formular Bezeichners den Header des Dokumentes mit den Schaltflächen „Schließen“ und „Speichern“, sowie der Ausgabe der Identifikation des Dokumentes, des Eigentümers, des Patienten und des Dokumentendatums. Unterhalb des Headers befinden sich die Blätter des Dokumentes. In Abbildung 37 sieht man das geöffnete Blatt Status des Dokuments mit der Identifikation 8761408. Unterhalb der Blattauswahl befinden sich die Eingabefelder des Dokumentes.



**Abbildung 37: geöffnetes Dokument im Browser.**

#### **4.3.5.8 Patienteneditor – patientEditor.jsp**

Falls zu einer Studie nicht anonymisiert dokumentiert werden soll, bietet der Patienteneditor die Möglichkeit, die Stammdaten (Vorname, Nachname, SVN, etc.) eines Patienten zu erfassen.

Beim Anklicken auf „Patient verwalten“ oder „Patient anlegen“ in der studyHandlerUI.jsp-Seite (siehe Abbildung 32), öffnet sich die patientEditor.jsp-Seite (siehe Abbildung 38). Bei Ersterem werden die Stammdaten jenes Patienten, der in der Auswahlliste angeklickt worden ist über eine Instanz der pagecode-Klasse PatientEditorUI aus einer Instanz der database-Klasse Patient geladen und anschließend in der patientEditorUI.jsp-Seite zum Editieren angezeigt. Beim Anklicken auf „Patient anlegen“ wird analog zum Klicken auf „Patient verwalten“ die patientEditorUI.jsp-Seite geöffnet, jedoch wird in diesem Fall ein neuer Patient angelegt. Durch Anklicken der „Speichern“-Schaltfläche werden die

Patientenstammdaten über eine Instanz der PatientenEditorUI-Klasse an eine Instanz der Patienten-Klasse gebunden und der SQL-Befehl INSERT oder UPDATE wird an die Datenbank geschickt.

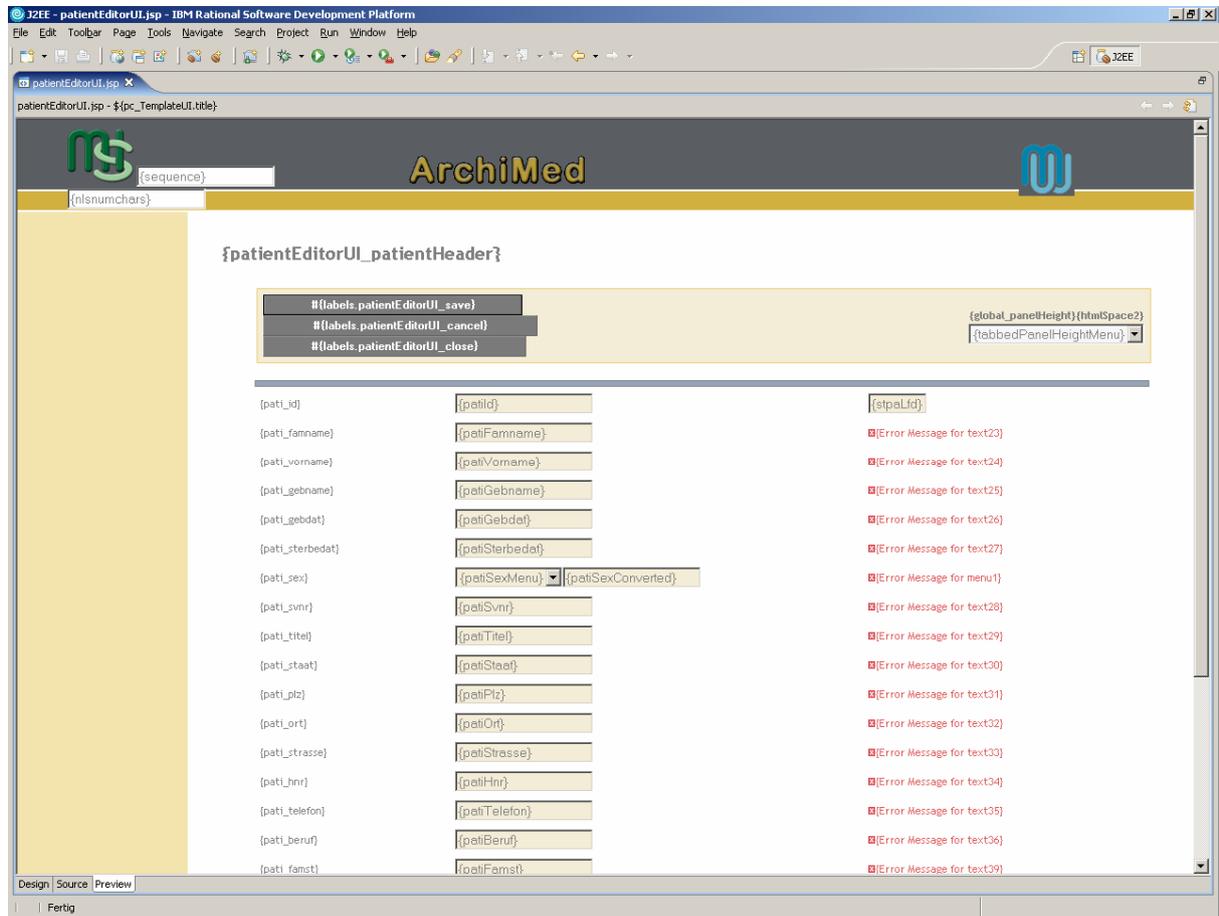


Abbildung 38: patientEditorUI.jsp-Seite in der Entwicklungsumgebung.

Abbildung 39 zeigt die patientEditorUI.jsp-Seite im Browser.

https://www.archimed.meduniwien.ac.at/ArchiMed/studyHandlerUI.faces

ArchiMed

Hauptseite Englisch Hilfe Beenden

Hauptseite  
 Studienauswahl  
 Studiendokumentation  
 Patientenstammdaten »  
 Auswertung

### Patientenstammdaten

Speichern Abbrechen Höhe des Ausschnittes 650

Identifikation	<input type="text" value="k wird automatisch vergeben"/>
Zuname	<input type="text" value="Mustermann"/>
Vorname	<input type="text" value="Mustermannvorname"/>
Geburtsname	<input type="text"/>
Geburtsdatum	<input type="text" value="01.01.1977"/>
Sterbedatum	<input type="text"/>
Geschlecht	<input type="text" value="unbekannt"/>
Svnr	<input type="text" value="1111"/>
Titel	<input type="text"/>
Staat	<input type="text"/>
Plz	<input type="text" value="1111"/>
Ort	<input type="text" value="Musterort"/>
Straße	<input type="text" value="Musterstraße"/>
Hausnummer	<input type="text" value="1"/>
Telefon	<input type="text" value="11111111"/>
Beruf	<input type="text" value="Musterberuf"/>
Familienstand	<input type="text"/>
Staatsbürgerschaft	<input type="text"/>
Eigentümer	<input type="text" value="Webdemo-Gruppe"/>
Änderungszeitpunkt	<input type="text"/>

**Hinweise:**

- Führen Sie die gewünschten Eintragungen bzw. Änderungen an den Stammdaten durch

**Abbildung 39: patientEditorUI.jsp-Seite im Browser.**

## 5 Diskussion

---

Die Verwendung des Java J2EE Standards und der Java Server Faces JSF Technologie hat sich als sich als gute Entscheidung herausgestellt. Die Nutzung der objektorientierten Spracheigenschaften von Java2 resultiert in einer leicht zu wartenden und stabilen Applikation. Die umfangreichen Klassenbibliotheken der Java2 Programmiersprache, sowie die große Anzahl an GUI Komponenten der JSF erleichterten die Entwicklung.

Die Entwicklungsumgebung von IBM, Rational Application Developer 6.0.1 ist enorm umfangreich, bietet sehr viel Funktionalität im Bereich der J2EE Entwicklung und ist für die Entwicklung einer J2EE Applikation für den IBM Websphere Application Server optimiert. Jedoch ist die IBM-seitige Implementierung des JSF Standards mangelhaft. Einige Methoden sind zum Teil gar nicht vorhanden, bei anderen sind nur die Methodenstumps, das sind leere Methoden ohne jegliche Funktionalität, implementiert. Regelmäßige Updates der Entwicklungsumgebung und des Applikationsservers zeigen jedoch, dass die IBM-Entwicklungen vorangetrieben werden. Einige der Schwachstellen der IBM-Entwicklung wurden bereits durch den Updatesupport behoben und es ist zu hoffen dass kommende Updates die übrig gebliebenen und neu entstandenen Entwicklungsfehler beheben.

Die Entwicklung der Webapplikation begann im Juni 2005 und endete im Dezember 2006. Während dieser Zeit waren 2 Personen vollzeitig und 1 Person halbezeitig mit der Programmierung beschäftigt. Der Aufwand der Entwicklung war enorm und der Fortschritt langsamer als gedacht. Auf Grund der hohen Komplexität der J2EE-Technologie war zum Erlernen und effektiven Nutzen dieser Technologie ca. ½ Jahr nötig. Der J2EE-Standard setzt sich aus vielen Spezifikationen zusammen. Die Kombination dieser Technologien schafft die Komplexität. Die Komplexität des J2EE-Standards ist so groß, dass der beim Entwickeln nötige Wissensstand immer wieder aufgefrischt werden muss, was den Fortschritt der Entwicklung bremst da immer wieder nachgelesen werden muss.

Das Hinzuziehen zusätzlicher Ressourcen zur Entwicklung scheiterte an der langen „Aufwärmphase“, die zum Erlernen des Standards und des Umgangs mit der Entwicklungsumgebung nötig gewesen wären.

Eine zukünftige Entwicklung mit der J2EE Technologie sollte gut überdacht werden. Es sollten unbedingt Reserven bezüglich Ressourcen eingeplant werden.

Die Geschwindigkeit der Webapplikation ist mäßig bis gut und ist von mehreren Faktoren abhängig:

1. der Geschwindigkeit der Datenbank
2. der Verbindung zwischen Applikationsserver und Datenbank
3. der IBM-Implementierung der virtuellen Maschine des Applikationsserver auf dem betreffenden Betriebssystem zur Interpretation der Instanzen, und .jsp-Dateien
4. der Verbindung zwischen Applikationsserver und Webbrowser
5. der Interpretation der vom Applikationsserver übertragenen HTML- und JavaScript-Informationen im Webbrowser.

Die Geschwindigkeit, mit der Abfragen an der Datenbank bearbeitet werden ist abhängig von der Tageszeit, da je nachdem mehrere Benutzer die Applikation verwenden. Jedoch sind die Abfragen der Webapplikation nicht groß und können auch bei Spitzenzeiten sehr schnell beantwortet werden.

Tests zufolge funktioniert die Verbindung zwischen Applikationsserver und Datenbank sehr gut. Der Verbindungsaufbau wird schnell durchgeführt und die Größe des kommunizierten Datenvolumens ist gering.

Schwierig festzustellen ist die Effizienz der virtuellen Maschine. Die Interpretation des großen Klassenframeworks sowie die Interpretation der .jsp-Seiten sind Tests zufolge langsam. Hier ist die Schuld sowohl bei der Effizienz der Programmierung der virtuellen Maschine des Applikationsserver als auch bei der Effizienz der Programmierung der Webapplikation zu suchen.

Große Geschwindigkeitsverluste kann die Verbindung zwischen Applikationsserver und Webbrowser bringen. Zwar ist die Größe des an den Webbrowser kommunizierten Datenvolumens bis auf die formUI.jsp-Seite gering, bei schlechter Internetverbindung jedoch kann der Seitenaufbau mitunter sehr lange dauern.

Die HTML- und JavaScript-Informationen müssen im Browser interpretiert werden. Die Geschwindigkeit der Interpretation ist abhängig von der Effizienz der

Programmierung des Browser und der Geschwindigkeit der Maschine, auf der der Browser ausgeführt wird. Besonders bei der browserseitigen Interpretation der Informationen, die über die formUI.jsp-Seite an den Browser geschickt werden, können Geschwindigkeitsverluste beobachtet werden, da die Anzahl dieser Informationen von der Größe des Formulars abhängt. In diesem Fall könnte eine Umprogrammierung der formUI.jsp-Seite eine Verbesserung bringen. Momentan werden alle Blätter eines Dokumentes in die Seite geladen. Dadurch ist das Wechseln der Dokumentblätter sehr schnell, weil es auf JavaScript basiert. Um die Geschwindigkeit der browserseitigen Interpretation der HTML- und JavaScript-Informationen zu verbessern, könnte man die Dokumentblätter einzeln laden. Dies hätte allerdings Geschwindigkeitseinbußen beim Dokumentblattwechsel zur Folge, da die Informationen des neuen Dokumentblattes erneut vom Applikationsserver übermittelt werden müssen. Beide Ansätze haben Vor- und Nachteile, falls bei der jetzigen Lösung der Seitenaufbau „zu lange“ (>30 Sekunden) dauert und Benutzerbeschwerden anfallen, dann sollte unbedingt die zweite Lösung angestrebt werden.

Im Jänner 2007 wurde die erste Studie mit der Archimed-Webapplikation umgesetzt. Diese Studie wurde komplett neu im System Archimed aufgesetzt und für das Web geöffnet. Es handelt sich in diesem Fall also um keine bestehende Studie die nachträglich für das Web geöffnet wurde. Ziel dieser Studie ist die Bewertung neuer Risikofaktoren bei Herzinfarktpatienten jünger als 40 Jahre. Die Studie wird multizentrisch - vorerst mit dem Wilhelminenspital, weitere Zentren werden aber noch folgen - durchgeführt. Für diesen Zweck wurden laut Studienprotokoll 3 Formulare entwickelt. Ein Formular „Baseline“ mit Hilfe dessen die Anamnesedaten, das sind Patienteninformationen die die Vorgeschichte des Patienten betreffen im Bezug auf aktuelle Beschwerden, erhoben werden. Ein Formular „Follow-Up“ zur Erhebung von Statusinformationen des Patienten bei einer Nachuntersuchung. Sowie ein Formular „Family Members“ zur Untersuchung von Familienangehörigen mit dem Ziel der Auffindung erblicher Veranlagungen, die einen Einfluss auf den Gesundheitszustand des Patienten haben können.

Die Erstellung der Formulare anhand der im Studienprotokoll definierten Dateneingabefelder dauerte ca. 1 Woche. Dabei war der insgesamt benötigte Aufwand zur Erstellung eines Webformulars ca. 8 bis 16 Stunden. Wesentlich

aufwendiger waren die Client- also HTML-seitigen Verfeinerungen z.B. Abhängigkeiten einzelner Dateneingabefelder zueinander. Diese Verfeinerungen entstanden größten Teils durch JavaScript-Funktionalitäten die für jedes Dateneingabefeld separat zu programmieren waren. Durch die hohe Anzahl der im Studienprotokoll geforderten Eingabehilfen, Konsistenz- und Plausibilitätsprüfungen lag der Aufwand der HTML-seitigen Verfeinerungen bei insgesamt ca.100 Stunden. Die Datenerhebung schreitet zügig voran, es wurden bereits 121 Dokumente erstellt. Mit ersten Ergebnissen der Studie ist jedoch nicht vor Ende 2007 zu rechnen.

## 6 Zusammenfassung und Ausblick

---

Ziel dieser Arbeit war die Planung und Realisierung eines webbasierten medizinischen Informationssystems zur Unterstützung klinischer Multicenterstudien. Die Aufgabe bestand darin das bestehende System Archimed zu analysieren, zu bewerten, herauszufinden ob eine Entwicklung einer Webkomponente einem kommerziellen Standardsystem vorzuziehen ist und gegebenenfalls die Entwicklung durchzuführen.

Die Systemanalyse ergab, dass aufgrund des generischen Datenmodells mit dem System Archimed die Durchführung von Multicenterstudien prinzipiell möglich wäre. Der Grund weshalb bisher noch nicht multizentrisch dokumentiert wurde, lag darin, dass dies wegen organisatorischer Schwierigkeiten und aus Sicherheitsgründen nicht möglich war.

Die Systembewertung ergab, dass eine Erweiterung des Systems um eine Webapplikation dank des generischen Datenmodells möglich wäre. Die Vorteile einer Webapplikation die auf die bestehende Datenbank und das bestehende Datenmodell aufsetzt wären, dass klinische Studien die bereits im System existieren für das Web geöffnet werden könnten. Bei Ankauf einer kommerziellen Standardsoftware wäre ein Zugriff auf den Datenbestand nur durch aufwendige Schnittstellenprogrammierung möglich gewesen. Des Weiteren würden einer Webapplikation weder organisatorische noch sicherheitskritische Schwierigkeiten im Weg stehen, wie das bei der Client-Server-Lösung der Fall ist.

Die zur Entwicklung verwendete Programmiersprache war Java2. Die im Rahmen des J2EE Standards spezifizierten Servlet, JSP und JSF Technologien bieten die Möglichkeit serverseitige Webanwendungen zu programmieren. Die zur Umsetzung dieser Technologien verwendete Entwicklungsumgebung war der IBM Rational Applications Developer 6.0. Die Anforderungen an die zu entwickelnde Webapplikation lagen in der Erfassung, dem Monitoring und Datenmanagement von Patienteninformationen sowie die Patientenverwaltung bei Multicenterstudien. Ein Hauptaugenmerk bei der Entwicklung der Webapplikation lag weiters auf der Optimierung der Ergonomie und Benutzbarkeit des GUI.

Zukünftige Weiterentwicklungen der Webapplikation sollen dem Kliniker die Möglichkeiten eröffnen, statistische Auswertungen über das Web zu starten. Am Applikationsserver sollen SAS-Prozeduren den Datenbestand der Datenbank auswerten. Der Kliniker soll über die Webapplikation wählen können welche Statistik gerechnet wird. Das Ergebnis (z.B. Grafik, Tabellen) der SAS-Prozeduren soll anschließend wieder in die Webapplikation eingebettet werden oder als Export zur Verfügung stehen.

## 7 Literatur

---

- (1) Gaus W. Dokumentation und Datenverarbeitung bei klinischen Studien: Books on Demand GmbH; 2004.
- (2) Tagnon H. General aspects of clinical trials, introduction. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 3-8.
- (3) Firket P. Computer system for multicentric clinical trials. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 205-217.
- (4) Gabbeke MV, Staquet M. The role of a Data Center. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 109-125.
- (5) Wikipedia. Klinische Studien. 2007.  
[http://de.wikipedia.org/wiki/Studie\\_\(Medizin\)](http://de.wikipedia.org/wiki/Studie_(Medizin))
- (6) Pauw MD. Forms: Design and contents. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 127-145.
- (7) Renard J, Glabbeke MV. Quality control and data handling. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 147-162.
- (8) Kerr M, Mill L. Guidelines for data collection and interpretation of patient charts. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V.; p. 73-89.
- (9) Putten E, Boon MC. Data Managers, skills and training. In: Data management and clinical trials; 1989: Elsevier Science Publishers B.V. p. 51-55.
- (10) Wikipedia. Klinischer Monitor. 2007.  
[http://de.wikipedia.org/wiki/Klinischer\\_Monitor](http://de.wikipedia.org/wiki/Klinischer_Monitor)
- (11) López-Carrero C, Arriaza E, Bolanos E, Ciudad A, Municio M, Ramos J, et al. Internet in clinical research based on a pilot experience. Contemporary Clinical Trials 2005;2005(26):234-243
- (12) Litchfield J, Freeman J, Schou H, Elsley M, Fuller R, Chubb B. Is the future for clinical trials internet-based? A cluster randomized clinical trial. Clinical Trials 2005;2005(2):72-79
- (13) Kuchenbecker J, Dick HB, Schmitz K, Behrens-Baumann W. Use of Internet Technologies for Data Acquisition in Large Clinical Trials. Telemedizin Journal and E-Health 2001;7(1):73-76
- (14) Sippel H, Ohmann C. A web-based data collection system for clinical studies using Java. Med. Inform. 1998;23(3):223-229
- (15) Wübbelt P, Hecker H. Management und Monitoring multizentrischer klinischer Studien im Internet. Telemedizinführer Deutschland 2000 <http://www.mh-hannover.de/institute/biometrie/Projekte/Webprojekt/Reisensburg/telemed.pdf>.
- (16) FDA. Guidance for Industry, Computerized Systems Used In Clinical Trials. 1999
- (17) FDA. 21 CFR Part 11: Food and Drug Administration; 1997.
- (18) Ammenwerth E, Haux R. IT-Projektmanagement im Krankenhaus und Gesundheitswesen: Schattauer; 2005.
- (19) Zuser W. Software Engineering, mit UML und dem Unified Process: Pearson Studium; 2001.

- (20) Duftschmid G, Gall W, Eigenbauer E, Dorda W. Management of data from clinical trails using the ArchiMed System. Med Inform Internet Med 2002;27(2):85-98  
<http://www.meduniwien.ac.at/msi/mias/papers/Duftschmid2002b.pdf>.
- (21) Dorda W, Wrba T, Duftschmid G, Sachs P, Gall W, Rehnelt C, et al. Archimed A Medical Information- and Retrieval System. Methods Inf Med 1999;38(1):16-24  
<http://www.meduniwien.ac.at/msi/mias/papers/Dorda1999a.pdf>.
- (22) ITSC. IT Systems & Communication, Richtlinie MUW User ID. 2007.  
<http://www.meduniwien.ac.at/itsc/policies/richtlinien/MUWUserID-R.pdf>
- (23) Huff SM, Haug PJ, Stevens LE, Dupont RC, Pryor TA. HELP the next generation: a new client-server architecture. Proc Annu Symp Comput Appl Med Care 1994:271-275
- (24) Friedman C, Hripcsak G, Johnson S, Cimino J, Clayton P. A generalized relational schema for an integrated clinical patient database. In: 14. Annual Symposium on Computer Applications in Medical Care; 1990; Washington D. C.; p. 335-339.
- (25) Brandt CA, Nadkarni P, Marengo L, Karras BT, Lu C, Schacter L, et al. Reengineering a database for clinical trials management: lessons for system architects. Control Clin Trials 2000;21(5):440-461
- (26) Oracle. Oracle Clinical Version 4.0 Application-wide Activities. 2000
- (27) ReseachTechnology. eResNet. 2007.  
[http://www.ert.com/products/clinical\\_research\\_.htm](http://www.ert.com/products/clinical_research_.htm)
- (28) InferMed. Macro. 2007. <http://www.infermed.com/index.php/macro>
- (29) SecuTrial. SecuTrial. 2007. <http://www.secutrial.com/>
- (30) Kuchinke W, Ohmann C, Bussar-Maatz R. Studiensoftware in den Koordinierungszentren für Klinische Studien (KKS): Status und Empfehlung für den Einsatz von Electronic Data Capture. In: GMDS; 2006; Leipzig.
- (31) interActiveSystems. interActiveSystems. 2007. [http://www.interactive-systems.de/html/2/2b/fs\\_2b.html](http://www.interactive-systems.de/html/2/2b/fs_2b.html)
- (32) Booch G. Objektorientierte Analyse und Design: Addison-Wesley; 1994.
- (33) Microsoft. Visual Studio 2007. 2007. <http://msdn2.microsoft.com/en-us/vstudio/default.aspx>
- (34) Beer W, Birngruber D, Moessenboeck H, Woesz A. Die.NET-Technologie: dpunkt.verlag; 2003.
- (35) Allamaraju S, Avedal K, Browett R, Diamond J, Griffin J, Holden M. Java Server Programming J2EE Edition: Wrox; 2000.
- (36) Sun, Microsystems. Java 2. 2007. <http://java.sun.com/reference/api/>
- (37) Mann KD. JavaServer Faces in Action: Manning; 2004.
- (38) Marinschek M, Schnabl A, Muellan G. JSF@Work: dpunkt.verlag; 2007.
- (39) IBM. IBM Rational Application Developer for WebSphere Software. 2007.  
<http://www-306.ibm.com/software/awdtools/developer/application/>