

iMedLearn

E-Learning Applikation für das iPhone

Diplomanden	Christian Ramseier & Sebastian Hunkeler
Examinator	Hans Rudin
Themengebiet	Softwareentwicklung auf dem iPhone
Industriepartner	Dr. Ulrich Woermann (IML Universität Bern) Markus Fretz (Crealogix)

Ausgangslage:

Die bestehende E-Learning Plattform des Instituts für Medizinische Lehre der Universität Bern (IML) ist eine öffentlich zugängliche Website für Medizinstudenten. Sie vereint statische und interaktive Lerninhalte aus verschiedenen medizinischen Fachbereichen. Nebst Texten, Bildern, Videos und Audiodateien werden auch Funktionen wie z.B. mit der Maus bedienbare Instrumente zum Vermessen von EKG-Daten angeboten. Dank solchen Funktionen können die Inhalte praxisbezogen vermittelt werden.

In Webbrowsern auf mobilen Endgeräten sind solche Inhalte aufgrund der kleineren Bildschirmgrösse und der limitierten Eingabemöglichkeiten leider nur beschränkt nutzbar.

Im Rahmen dieser Arbeit sollte darum das Themengebiet „Elektrokardiogramm“ aus dem Bereich der Kardiologie massgeschneidert für das iPhone verfügbar gemacht werden. Die systematische Befundung von aufgezeichneten Elektrokardiogrammen war bei der Umsetzung auf dem iPhone insbesondere bezüglich des User Interfaces eine grosse Herausforderung.



Abbildung 1:
Lernen auf dem iPhone

Vorgehen/Technologien:

In einem ersten Schritt wurden in Zusammenarbeit mit dem IML die exakten Anforderungen an die iPhone Applikation erarbeitet. Mit Hilfe von regelmässigen Sitzungen, Feedbacks und Paper Prototypes (siehe Abb. 2) konnte ein gutes Bild des angestrebten Resultats aufgezeigt werden.

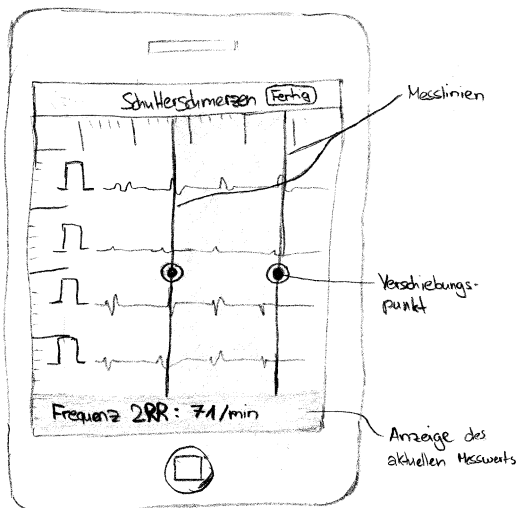


Abbildung 2: Paper Prototype
„Vermessung eines EKGs“

Für die in der Datenbank bereits vorhandenen Daten musste nun eine dateibasierte Schnittstelle definiert werden, welche auch vom iPhone gelesen werden kann.

Mit Hilfe der am IML bereits eingesetzten Web-Technologien wurde ein automatisierter Datenexport entwickelt. Für die Umwandlung der zum Teil unstrukturierten Daten musste zusätzlich ein Parser entwickelt werden.

Die Visualisierung auf dem iPhone konnte nur selten mit bereits vorhandenen User Interface Bausteinen implementiert werden. Meist mussten für die Problemstellung massgeschneiderte Elemente komplett neu entwickelt und bestehende Komponenten erweitert werden.

Ergebnisse:

Mit iMedLearn steht nun für das iPhone eine umfangreiche Lernapplikation zum Thema „Elektrokardiogramm“ bereit. Nebst klassischen Lerninhalten wie Texte und Bilder können sämtliche bei der Befundung relevanten Informationen in einer Herzrhythmusaufzeichnung gemessen, notiert und mit der Musterlösung verglichen werden. Eine durchdachte Architektur stellt dabei sicher, dass die Applikation für andere Lerninhalte einfach adaptierbar ist.

Zusätzlich steht dem IML ein Werkzeug zur Verfügung, um die Lerninhalte aus der bestehenden Umgebung zu exportieren und auf iPhone-Kompatibilität zu prüfen.

Ausblick:

Das Projekt wurde nach Abschluss an die Mitarbeiter des IML übergeben. Die Anwendung soll nach Bereinigung der Lerninhalte im AppStore von Apple publiziert werden.



Abbildung 3: Umsetzung der
„Vermessung eines EKGs“

Anforderungsspezifikation iPhone Applikation iMedLearn

Christian Ramseier & Sebastian Hunkeler
Studienarbeit HS 09

18. Dezember 2009

Version: 0.5

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

Änderungsgeschichte

Datum	Version	Änderung
5.10.2009	0.1	Erstellung des Dokuments
12.10.2009	0.2	Ergänzt um Thema «Schnittstellen»
13.10.2009	0.3	Anpassungen gemäss Sitzungsprotokoll vom 13.10.2009
16.10.2009	0.4	Anpassungen gemäss Review vom 13.10.2009
23.10.2009	0.5	Neue Schnittstellenspezifikation integriert

Referenzen

- [documentation/Analyse/PDF/anforderungsliste.pdf](#)
- [documentation/Sitzungsprotokolle/PDF/sitzungsprotokoll-09-10-13.pdf](#)
- [documentation/Analyse/Schnittstelle und Deployment/datastructure/Datei-Ordnerstruktur.eps](#)

Contents

1	Einleitung	4
1.1	Übersicht	4
2	Ausgangslage	4
2.1	Zweck der Software	4
2.2	Lerninhalte	4
2.3	Eingesetzte Technik	5
3	Allgemeine Beschreibung	5
3.1	Produktperspektive	5
3.2	Produkt Funktion	5
3.3	Benutzer Charakteristik	6
3.4	Einschränkungen	6
3.5	Annahmen	6
3.6	Abhängigkeiten	6
4	Umfang	6
5	Use Case Überblick	7
6	Funktionale Anforderungen	7
6.1	Paketierungskomponente	7
6.1.1	Funktionalität	7
6.1.2	Interoperabilität	7
6.2	iPhone Applikation	7
6.2.1	Erweiterbarkeit	7
6.2.2	Funktionsmodule	8
6.2.3	Deployment der Inhalte	8
6.3	Funktionalitätsübersicht	8
6.3.1	Muss-Anforderungen	8
6.3.2	Kann-Anforderungen	8
6.4	Überprüfbarkeit	9
6.5	Bedienbarkeit	9
6.5.1	Verständlichkeit	9
6.5.2	Erlernbarkeit	9
7	Modulspezifikation	9
8	Schnittstellen	10
8.1	Zuständigkeit	10
8.1.1	IML	10
8.1.2	Projektteam	10
8.1.3	IML/Projektteam	10
8.2	Umfang der Schnittstelle	10
8.2.1	Navigationsdaten	10

8.2.2	Gemeinsam verwendete Daten	11
8.2.3	Artikeldaten	11
8.2.4	Visualisierung	11

1 Einleitung

Es gilt in Zusammenarbeit mit dem Institut für Medizinische Lehre (IML) der Universität Bern eine E-Learning Applikation für das iPhone zu entwickeln. Die bestehende Plattform¹ des IML ist sehr umfangreich, in einem ersten Schritt soll ein spezieller Augenmerk auf die Übung im Umgang mit Elektrokardiogrammen (EKG) gelegt werden.

Die Anwendung ist insofern Attraktiv, als dass man die Lernmaterialien immer dabei hat und so beispielsweise eine Wartezeit überbrücken kann. Ausserdem kann man so jederzeit offline auf die Lernmaterialien zugreifen.

1.1 Übersicht

Im weiteren Verlauf dieses Dokuments wird genauer auf die Anforderungen der iPhone Applikation eingegangen. Dies heisst konkret Produkt Perspektiven, Funktionalität sowie die Benutzer Charakteristik. Es werden auch die Bedienbarkeit, Zuverlässigkeit, Leistung und Wartbarkeit behandelt.

2 Ausgangslage

2.1 Zweck der Software

Die iPhone Applikation ermöglicht es, mobil auf die Lerninhalte der bestehenden E-Learning Plattform des IML zuzugreifen. Die Präsentation der Daten und die Struktur soll für das iPhone optimiert werden. Das Frontend des bestehenden E-Learning Portals bleibt unverändert.

2.2 Lerninhalte

Die E-Learning Plattform des IML ist eine öffentlich zugängliche Website für Medizinstudenten, welche auch an anderen Universitäten verwendet wird. Sie vereint statische und interaktive Lerninhalte aus verschiedenen medizinischen Fachbereichen. Neben Texten, Bildern, Videos und Audiodateien werden auch Funktionen wie z.B. eine Lupe zum Vergrössern von Bildausschnitten oder mit der Maus bedienbare Instrumente zum Vermessen von EKG-Daten angeboten. Dank solchen Funktionen können die Inhalte praxisbezogener vermittelt werden.

Die E-Learning Website des IML ist nach den verschiedenen Fachgebieten strukturiert. Die Gliederung sieht wie folgt aus:

- Abteilung > Fachgebiet > Lernmodul > Kapitel > Seite / Slide

Zum besseren Verständnis hier ein konkretes Beispiel:

- Klinische Abteilung > Kardiologie > Elektrokardiogramm > Einführung > Historische Entwicklung des EKGs

¹<http://e-learning.studmed.unibe.ch>

Je nach Inhalt ist die jeweilige Lerneinheit einem Funktionsmodul zugordnet. Aufgrund dieser Zuteilung werden dem Benutzer die verschiedenen Funktionen präsentiert. Beispiele:

- Befundung aufgrund eines Schnittes (Funktionsmodul "Grafik") - Lupe wird angezeigt, der Benutzer kann einzelne Bildausschnitte vergrößern.
- Beurteilung eines EKGs (Funktionsmodul "Elektrokardiogramm") - Diverse interaktive Messinstrumente sowie ein Formular, in welchem der Lernende seinen Befund eintragen kann, wird angezeigt. Nach dem Messen kann er seinen Befund mit der Musterlösung vergleichen.

Durch den modularen Aufbau der Lerninhalte sollte es möglich sein, unabhängig und schrittweise einzelne Funktionsmodule zu implementieren.

2.3 Eingesetzte Technik

Die bereits vorhandene E-Learning Plattform des IML besteht in Form einer Website. Vorhandene Lerninhalte sind in einer strukturierten Form abgelegt. Die Darstellung der Website erfolgt dynamisch. Zur Aufbereitung der Darstellung wird JavaScript verwendet, die Berechnung geschieht auf dem Client. Funktionen wie z.B. die Lupe sind ebenfalls mit JavaScript realisiert.

Die meisten Inhalte werden in einem Backend verwaltet (PHP). Dabei werden die Inhalte aus einer Datenbank gelesen und in JavaScript Dateien geschrieben. Diese Dateien werden dann zur Anzeige verwendet. Selten kommt es vor, dass die Funktionalität des Backends nicht ausreicht und ein Eingreifen über FTP Zugang nötig wird.

3 Allgemeine Beschreibung

3.1 Produktperspektive

Es existieren bereits ähnliche Applikationen im Medizinbereich. Jedoch sind diese meistens kostenpflichtig und decken den Inhalt der IML Lernplattform nicht ab. Für das Projektteam ist es zudem sehr wichtig Erfahrungen im Bereich iPhone Entwicklung zu sammeln und einen Software-Entwicklungsprozess für ein embedded System bzw. für ein Smartphone einmal konkret erlebt zu haben.

3.2 Produkt Funktion

Das Produkt ist eine E-Learning Applikation, welche den Inhalt der bestehenden online Plattform auf dem iPhone anbietet. Die Präsentation der Daten und die Darstellung der Bedienelemente werden so konzipiert, dass sie auf dem kleinen Multitouch-Display bedient werden können. Ausserdem soll das User Interface der Applikation den Apple Human Interface Guidelines gerecht werden.

3.3 Benutzer Charakteristik

Unsere Applikation soll primär Medizinstudenten und Ärzte ansprechen.

3.4 Einschränkungen

Es besteht eine Einschränkung betreffend der Maximalgrösse einer iPhone Applikation von 2 GB. Ausserdem muss das Deployment über den App Store erfolgen, da es nicht möglich ist Applikationen direkt auf das Gerät zu kopieren.

3.5 Annahmen

Es existieren Restriktionen betreffend der möglichen Funktionalität einer Applikation basierend auf den Apple Policies für den AppStore. Unsere Applikation wird von Apple angenommen und im App Store zur Verfügung gestellt. Wir nehmen an, dass die Bilder der Lerninhalte in geeigneter Auflösung vorhanden sind und gemäss der im Abschnitt «Schnittstellen» spezifizierten Schnittstelle vom IML bereitgestellt werden.

3.6 Abhängigkeiten

Die Software hängt vom iPhone SDK bzw. vom Cocoa Framework sowie dem Sandboxing der iPhone Plattform ab und baut komplett darauf auf.

Die Software wird für die iPhone OS Version 3.0 entwickelt, und erfordert im Minimum diese Version um korrekt zu funktionieren.

4 Umfang

Die E-Learning Plattform beinhaltet folgende Lernprogramme:

- MorphoMed
- CliniSurf
- RadioSurf
- PediRad
- DentoSurf

Im Rahmen dieser Studienarbeit konzentrieren wir uns auf die das Lernmodul “Elektrokardiogramm” des CliniSurf Lernprogrammes. Dies beinhaltet primär die Entwicklung von 3 Modulen:

- Text mit Bild (mit ein-/ausschaltbaren Beschriftungen)
- Text mit Video (optional, siehe Abschnitt «Funktionalitätsübersicht»)
- Spezialmodul mit den Vermessungswerkzeugen für Elektrokardiogramme

Der Schwerpunkt dabei liegt in der Umsetzung der Benutzeroberfläche und dem Aufbereiten der Inhalte.

5 Use Case Überblick

- Der Benutzer lädt die Applikation herunter worauf sie installiert wird
- Text/Bild Lerninhalt anzeigen: Der Benutzer wählt den gewünschten Lerninhalt aus und lässt ihn anzeigen
- Text/Video Lerninhalt anzeigen: Der Benutzer wählt den gewünschten Lerninhalt aus, lässt ihn sich anzeigen und spielt den Video ab
- Elektrokardiogramm
 - Informationen sammeln: Der Benutzer öffnet den gewünschten EKG Lerninhalt und sammelt mit Hilfe von Messwerkzeugen die nötigen Informationen
 - Befund ausfüllen: Der Benutzer holt sich in der EKG Lerneinheit die benötigten Informationen und erstellt den systematischen Befund
 - Befund auswerten: Der Benutzer wertet seinen systematischen Befund aus und vergleicht ihn mit der Musterlösung

6 Funktionale Anforderungen

Die zu erstellende Applikation besteht aus einer für die Inhabtaufbereitung zuständigen Paketierungskomponente auf der Serverseite, sowie der eigentlichen iPhone Lernapplikation.

6.1 Paketierungskomponente

6.1.1 Funktionalität

Die Aufgabe der Paketierungskomponente ist es, die vom Backend (siehe Nomenklatur) gespeicherten Inhalte in eine für unsere iPhone Applikation lesbare Form zu bringen.

6.1.2 Interoperabilität

Die Paketierungskomponente darf bestehende Inhalte nicht verändern. Die genaue Schnittstelle zwischen dem bestehenden Backend und der Paketierungskomponente wird in Zusammenarbeit mit U. Woermann festgelegt.

6.2 iPhone Applikation

6.2.1 Erweiterbarkeit

Die Grundfunktionen der Applikation soll im Stil eines Frameworks aufgebaut werden. Damit gemeint ist allerdings kein Objective-C Framework, sondern die Projektstruktur im Bezug auf Erweiterbarkeit. Der Grund dafür ist, dass es aus

Performance Gründen nicht möglich ist eigene Frameworks auf dem iPhone zu verwenden (siehe Apple Reference Library²). Zum Framework gehören die Navigation und Schnittstellenspezifikation. Es soll möglich sein, auf einfache Weise die Applikation um weitere Module zu ergänzen, ohne den bestehenden Code zu ändern. Die eigentliche Darstellung der Lerninhalte erfolgt über individuell implementierte Module.

6.2.2 Funktionsmodule

Jedem Slide sind im Backend je nach Inhalt mehrere Funktionen zugewiesen. Wir bilden eine Teilmenge dieser Funktionen als Funktionsmodul auf das iPhone ab.

6.2.3 Deployment der Inhalte

Die Lerninhalte (Texte, Bilder, etc) werden mit der iPhone Applikation im AppStore ausgeliefert. Es muss keine Funktionalität zum Nachladen und Aktualisieren der bestehenden iPhone Inhalte implementiert werden.

6.3 Funktionalitätsübersicht

6.3.1 Muss-Anforderungen

1. Implementierung des Frameworks für das iPhone. Dies umfasst standardisierte Navigation und einfach erweiterbare Schnittstellenspezifikation.
2. Text / Bild Modul - Darstellung des Textes / Bildes, Zoomfunktion
3. EKG Modul - lupenähnliche Zoomfunktion, Beurteilungsformular, Referenzbild, Vermessungswerkzeuge, Diagnoseauswertung

6.3.2 Kann-Anforderungen

- **Priorität 1: Serverseitige Paketierungskomponente** - Verpackt die Lerninhalte in ein für unsere iPhone Applikation lesbares Format
 - Die Paketierungskomponente wird vom IML umgesetzt. Da der zuständige Entwickler momentan ausgelastet ist, werden wir die Umsetzung der Paketierungskomponente in Angriff nehmen. Dies allerdings nur, falls es die Zeit erlaubt. In einem ersten Schritt werden Dummydaten verwendet, welche den im Abschnitt «Schnittstellen» aufgeführten Schnittstellen entsprechen.
- **Priorität 3: Annotations für Grafikmodule** (Sound-, Grafik-Einbettung innerhalb des Textes)
- **Priorität 4: Internationalisierung**

²<http://developer.apple.com/mac/library/documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html>

- Priorität 4: Anleitung auf iPhone
- Priorität 5: Entwicklung weiterer Module

6.4 Überprüfbarkeit

Zur Überprüfung der Analyse Kriterien ist das Dokument `anforderungsliste.pdf` (siehe Abschnitt Referenzen) beizuziehen, welches eine Liste der oben erwähnten Anforderungen mit dem Status derer Umsetzung enthält.

6.5 Bedienbarkeit

Die Bedienbarkeit des Produktes ist ein sehr wichtiger Aspekt. Da die Bedienung des iPhones über eine kleine Touch Oberfläche läuft, muss die Applikation einfach zu bedienen sein.

6.5.1 Verständlichkeit

Die Benutzeroberfläche soll möglichst intuitiv und einfach zu verstehen sein, da auf dem kleinen Bildschirm wenig Platz bleibt für Bedienungshilfen und Anleitungen.

6.5.2 Erlernbarkeit

Es soll dem Benutzer nach kurzer Zeit möglich sein (5-10 Minuten) das Programm grundsätzlich zu verstehen und die wichtigsten Bedienelemente ihrer Funktion zuzuordnen. Dies sollte für die meisten Anwender kein Problem darstellen, da diese die Web Version schon kennen.

7 Modulspezifikation

Text/Bild Modul: Im Text/Bild Modul soll ein Text und ein Bild angezeigt werden, welches zoom-bar ist.

Im EKG Modul soll ...

- ... die Aufgabenstellung angezeigt werden
- ... eine systematische Befundung erstellt/ausgefüllt werden können
- ... das EKG Bild mit Referenzbildern angezeigt werden
- ... mit einem Werkzeug, welches von der Funktionalität her der Schublehre aus der E-Learning Plattform entspricht, die nötigen Vermessungen vorgenommen werden können.

8 Schnittstellen

Um die Daten von der bestehenden IML E-Learning Plattform auf das iPhone zu bringen, wird eine klar definierte Schnittstelle benötigt. Der folgende Abschnitt spezifiziert die Anforderungen an diese Schnittstelle und beinhaltet eine Visualisierung der auszuliefernden Datenpakete.

8.1 Zuständigkeit

8.1.1 IML

Für die Bereitstellung der Daten der E-Learning Plattform ist das IML zuständig. Die Neuaufbereiten der Applikation bei einer Änderung der Daten im bestehenden CMS wird vom IML übernommen.

8.1.2 Projektteam

Die Programmierung der iPhone Applikation mit den im Abschnitt «Funktionalitätsübersicht» besprochenen Modulen wird vom Projektteam übernommen.

8.1.3 IML/Projektteam

Die Implementierung der Aufbereitung & Paketierung der Inhalte wird wie im Abschnitt «Funktionalitätsübersicht» beschrieben, in einer späteren Phase vom IML oder von uns umgesetzt.

8.2 Umfang der Schnittstelle

Sämtliche Lerndaten werden auf einem Webserver des IML bereitgestellt. Der Download und die Integration in die Entwicklungsumgebung und somit in die iPhone Applikation erfolgt manuell durch den Entwickler. Die Schnittstelle muss in der Lage sein, folgende Daten bereitzustellen:

8.2.1 Navigationsdaten

Pro Kapitel werden folgende Informationen geliefert:

- Name des Kapitels
- Titel des Kapitels
- Typ des Kapitels (z.B. Unterkapitel oder “EKG Befund”)

Dieser Teil der Schnittstelle soll so generell wie nur möglich gehalten werden, um die Erweiterbarkeit auf andere Lernmodule zu gewährleisten.

8.2.2 Gemeinsam verwendete Daten

Funktionsmodulspezifische Daten sollen separat Abgelegt werden können. Damit sind Ressourcen gemeint, welche von mehreren Slides des gleichen Funktionsmodul verwendet werden können (z.B. Referenzbilder)

8.2.3 Artikeldaten

Sämtliche für die Darstellung eines spezifischen Artikels notwendigen Daten. Für einen Artikel des Funktionsmoduls "EKG" wären das beispielsweise folgende Daten:

- Das Elektrokardiogramm (als Bilddatei)
- Die Aufgabenstellung
- Die Musterlösung

8.2.4 Visualisierung

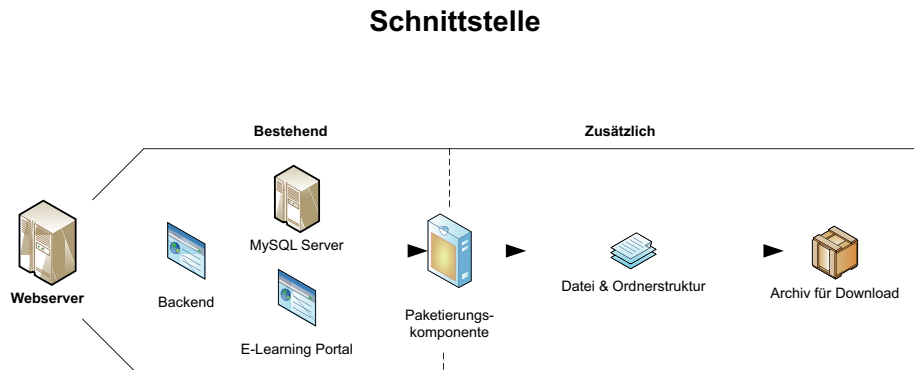


Figure 1: Übersicht Deployment

Die Paketierungskomponente soll als Resultat ein Archiv mit sämtlichen für die iPhone Applikation aufbereiteten Daten zur Verfügung stellen. Dieses Archiv soll vom Entwickler einfach in die Entwicklungsumgebung bzw. das iPhone Programm importiert werden können.

Nomenclature

Backend Bei vielen Programmen ist eine zentrale Verwaltung der Daten nötig. Dieses Programm läuft dann auf dem Server und wird Backend genannt.

EKG Das Elektrokardiogramm (EKG) ist die Registrierung der Summe der elektrischen Aktivitäten aller Herzmuskelfasern. Elektrokardiogramm

heißt auf Deutsch Herzspannungskurve, gelegentlich wird es auch Herzschritt genannt.

Frontend Das Frontend bezeichnet in unserem Fall die Visualisierung bzw. Präsentation der Lerninhalte

Funktionsmodul Als Funktionsmodul bezeichnen wir den Programmcode der auf eine bestimmte Funktionalität spezialisiert ist.

Anforderungskatalog

Christian Ramseier & Sebastian Hunkeler
Studienarbeit HS 09

December 18, 2009

Version: 0.6

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

Änderungsgeschichte

Datum	Version	Änderung
10.10.2009	0.1	Erstellung des Dokuments
28.10.2009	0.2	Erfüllte Muss Anforderung eingetragen
2.11.2009	0.3	Erfüllte Kann Anforderung eingetragen
17.11.2009	0.4	Erfüllte Muss Anforderung eingetragen
11.12.2009	0.5	Erfüllte Muss Anforderung eingetragen
14.12.2009	0.6	Erfüllte Kann Anforderung eingetragen

Referenzen

- anforderungsspezifikation.pdf

1 Muss Anforderungen

Anforderung	Erfüllt
Navigation	✓
Text / Bild Modul	✓
EKG Modul	
Aufgabenstellung	✓
Befundung	✓
Winkelmesser	✓
Schubleere horizontal	✓
Schubleere vertikal	✓
Referenzbilder	✓

2 Kann Anforderungen

Anforderung	Erfüllt
Serverseitige Paketierungskomponente	✓
Validierung der Datenbankinhalte	✓
In den Text eingebettete Grafiken	✓
Automatisierungsskript für iPhone App Content Deployment	✓
Text / Video Modul	
Annotations für Grafikmodule	✓
Anleitung auf iPhone / Hilfestellung zu Messwerten	
Internationalisierung	✓

Projektplan

iPhone Applikation iMedLearn

Christian Ramseier & Sebastian Hunkeler
Studienarbeit HS 09

December 18, 2009

Version: 0.1

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
23.9.2009	0.1	Erstellung	SH,CR

Referenzen

- documentation/Analyse/PDF/anforderungsliste.pdf
- documentation/Analyse/PDF/anforderungsspezifikation.pdf
- documentation/Planung/Zeitplan.xls

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Einführung	4
2.1	Zweck	4
2.2	Begriffe	4
3	Phasen und Meilensteine	5
4	Risikomanagement	5
4.1	Risikoverhalten & Verantwortlichkeit	5
4.2	Risiken & Massnahmen	7
5	Arbeitspakete	8
5.1	Referenzen	8
5.2	Projekt Managment	8
5.2.1	Ideenfindung	8
5.2.2	Grafisches / strukturelles Grundkonzept	8
5.2.3	Projektplan	8
5.2.4	Review / Anpassungen	8
5.2.5	Dokumentvorlagen / Einrichtung	8
5.2.6	Risk Management	8
5.3	Requirements	8
5.3.1	Use Cases «brief»	8
5.3.2	Anforderungsspezifikationen	8
5.3.3	Review Anforderungen	8
5.4	Analyse	9
5.4.1	User Environment Model	9
5.4.2	Domänenmodell	9
5.4.3	Contracts	9
5.4.4	System Sequence Diagram	9

5.4.5	Review	9
5.5	Design	9
5.5.1	Externes Design	9
5.5.2	Klassendiagramm	9
5.5.3	Kommunikationsarchitektur	9
5.5.4	Internes GUI Design	9
5.5.5	Review	9
5.6	Implementation	9
5.6.1	Prototypen (GUI, PD, Architektur)	9
5.6.2	Datenhaltung / Serverkomponente / Packaging	10
5.6.3	Fehlerbehandlung	10
5.6.4	Review und Refactoring	10
5.6.5	Projekt- / Codestruktur	10
5.6.6	Ausbau GUI / Grafik	10
5.6.7	Ausbau Problem Domäne	10
5.7	Test	10
5.7.1	Unit Tests	10
5.7.2	Usability Tests	10
5.7.3	Performance	10
5.7.4	System Tests	10
5.8	Dokumentation	11
5.8.1	Anleitung	11
5.8.2	Software Architecture Document	11
5.8.3	Abschlussbericht	11
5.8.4	Erfahrungsreport	11
5.9	Tasks wiederkehrend	11
5.9.1	Studium Technologien	11
5.9.2	Studium Cocoa Framework	11
5.9.3	Studium Objective C	11
5.10	Sitzungen	11
5.10.1	Sitzungsprotokoll	11
5.10.2	Besprechung mit Betreuer	11
5.10.3	Problemlösung	11
5.11	Qualitätssicherung	12
5.11.1	Dokumentation und Code Style	12
5.11.2	Arbeitsablauf diskutieren / optimieren	12
6	Infrastruktur	12
6.1	Geräte	12
7	Qualitätsmassnahmen	12
7.1	Dokumentation	12
7.2	Reviews	12
7.3	Codierrichtlinie	12
7.4	Sitzungsprotokolle & Taskliste	13
7.5	Versionierung	13

7.6	Unit Testing	13
7.7	Usability Tests	13
7.8	Zeitplanung	13
7.9	Performance	13
7.10	Kompatibilität / Portabilität	13
7.11	Change Management	14
7.12	Automatisierung	14

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt die Planung der Studienarbeit bzw. der zu entwickelnden iPhone Applikation der Studenten Sebastian Hunkeler und Christian Ramseier.

Die nachfolgenden Kapitel geben einen Überblick über das Projekt, dessen Organisation und Management Prozesse. Ein grosser Teil befasst sich mit der Planung der Arbeitspaketen und benötigter Stunden. Ausserdem wird genauer auf das Risiko- und Qualitäts-Management eingegangen sowie auf die verwendete Infrastruktur während der Projektlaufzeit.

2.2 Begriffe

Um Missverständnisse bei der Kommunikation unter den Projektbeteiligten zu vermeiden, müssen gewisse Begriffe in ihrer Bedeutung geklärt werden.

iPhone Applikation Das Programm, welches auf dem iPhone ausgeführt wird.

Seite / Slide Die eigentlichen Lerninhalte. Eine Lerneinheit umfasst inhaltlich einen Artikel auf der bestehenden E-Learning Seite. Jede Lerneinheit ist einem Funktionsmodul zugeordnet. Diese Funktionsmodulzuordnung ist unabhängig von der inhaltlichen Struktur.

Funktionsmodul Ein Teil der iPhoneApplikation, welcher für die Darstellung und Interaktion mit denjenigen Lerneinheiten zuständig ist, welche dem Modul zugeordnet sind. iStudMed kann folglich nur jene Lerninhalte anzeigen, für welche das Zugehörige Modul auf dem iPhone implementiert ist.

Framework Die von Funktionsmodul und Inhalten unabhängigen Grundfunktionen der iPhone Applikation (z.B. die Navigation oder der Download / Update von Lerninhalten).

Backend Bezieht sich auf die Webseiten StudiSurf, MorphMed, etc. Zum Bearbeiten der Lerninhalte besteht ein simples CMS (PHP).

3 Phasen und Meilensteine

	Meilenstein	Deliverables	Woche	Phase
MS1	Anforderungsanalyse	Anforderungsspezifikation	W03	Inception
MS2	Internal & External Design	Projektplan, Externes Design (Paper Prototypes), Domain Model	W06	Elaboration (1. Iteration)
MS3	Architektur Prototyp	Entwurf System Architecture Document, iStudMed Prototyp (Navigation, Funktionsmodul "Text / Bild")	W08	Elaboration (2. Iteration)
MS4	Release Candidate	iStudMed Release Candidate (Navigation, Funktionsmodul "Text / Bild", Funktionsmodul "EKG")	W13	Construction (2. Iteration)
MS5	Projektabschluss	Benutzerhandbuch, System Architecture Document	W14	Transition

4 Risikomanagement

4.1 Risikoverhalten & Verantwortlichkeit

In diesem Kapitel sind mit dem Projekt verbundene Risiken und zugehörige Massnahmen definiert. An den wöchentlichen Meetings wird beschlossen, ob ein solches Risiko eingetroffen ist und welche Massnahmen zu treffen sind.

4.2 Risiken & Massnahmen

Level	Risiko	Eintrittswahrscheinlichkeit / Schaden bei Inception Phase	Eintrittswahrscheinlichkeit / Schaden nach Elaboration Phase	Prävention	Massnahme
1	Ein Teammitglied fällt durch Krankheit aus	40% Schaden 12h	40% Schaden 8h	Unverplante Zusatzwoche	Arbeiten aufteilen, Umfang verkleinern, Abgabetermin in Rücksprache mit dem Betreuer verschieben.
1	Projektumfang unterschätzt (gilt z.B. auch für Einarbeitung in Objective C)	20% Schaden 40h	10% Schaden 10h	Zusätzlich Zeit einplanen	Wochenpensum erhöhen, evtl Umfang reduzieren
2	Streit / Uneinigkeit im Team	10% Schaden 30h	5% Schaden 12h	Meinungen aller Teammitglieder berücksichtigen	Gespräche im Team Externe Parteien hinzuziehen (Betreuer / Mitstudenten)
2	SVN Ausfall	70% Schaden 6h	70% Schaden 2h	Regelmässige Checkouts (Working Copy's), Backup durch HSR	Beanspruchung der HSR Informatikdienste
3	Technische Probleme	5% Schaden 30h	5% Schaden 20h	Zusammenarbeit mit Herrn Fretz (Crealogix) und Know How von Herrn Graf (HSR)	Zuziehen von weiteren Fachpersonen oder Apple Support
	Reserve 18h (Woche 14)				

5 Arbeitspakete

5.1 Referenzen

Eine Aufstellung der geschätzten Aufwände der einzelnen Arbeitspakete ist im Zeitplan zu finden. (Siehe documentation/Planung/Zeitplan.xls)

5.2 Projekt Managment

5.2.1 Ideenfindung

Im Gespräch mit den Projektteilnehmern soll eine gemeinsame Vision für die fertige iPhone Applikation gefunden werden.

5.2.2 Grafisches / strukturelles Grundkonzept

Erarbeiten von eines grundlegenden Konzeptes mit Hilfe von Paper Prototypes. Anpassung in Zusammenarbeit mit dem Kunden.

5.2.3 Projektplan

Erstellen und nachführen dieses Dokuments.

5.2.4 Review / Anpassungen

Besprechen und überarbeiten von Dokumentationen und Konzepten.

5.2.5 Dokumentvorlagen / Einrichtung

Erstellen der Dokumentvorlagen, Einrichten des SVN Repository, Aufsetzen der Development Umgebung

5.2.6 Risk Management

Riskio Management erstellen, Risiken überprüfen und ggf. Massnahmen ergreifen.

5.3 Requirements

5.3.1 Use Cases «brief»

Use Cases in “brief” Form erarbeiten.

5.3.2 Anforderungsspepifikationen

Anforderungen erarbeiten.

5.3.3 Review Anforderungen

Anforderungen in Zusammenarbeit mit dem Projektteam und dem Kunden

5.4 Analyse

5.4.1 User Environment Model

Erstellung eines User Environment Model für die iPhone Applikation

5.4.2 Domänenmodell

Erstellung eines Domänenmodells

5.4.3 Contracts

Erstellen evtl. zu verwendender Contracts.

5.4.4 System Sequence Diagram

Erstellung von sinnvollen System Sequence Diagram

5.4.5 Review

Review und Anpassungen an den Analyse Dokumenten.

5.5 Design

5.5.1 Externes Design

Erarbeiten des externen Designs mit Paper Prototypes, dem Apple UI Designer und Grafiktools.

5.5.2 Klassendiagramm

Erstellen und nachführen des Klassendiagramms.

5.5.3 Kommunikationsarchitektur

Erarbeiten einer internen Kommunikationarchitektur / Schichtenmodell.

5.5.4 Internes GUI Design

Erarbeiten des internen GUI Designs.

5.5.5 Review

Design Reviews / Dokumentation nachführen

5.6 Implementation

5.6.1 Prototypen (GUI, PD, Architektur)

Erstellung eines auf dem iPhone lauffähigen Prototyps. Grober Aufbau und die Struktur des GUIs müssen hier sichtbar sein.

5.6.2 Datenhaltung / Serverkomponente / Packaging

Implementierung der Datenhaltung und des Packagings

5.6.3 Fehlerbehandlung

Implementierung der Fehlerbehandlung.

5.6.4 Review und Refactoring

Der Code wird von den Projektmitarbeitern gegengelesen, Unschönheiten besprochen und mittels Refactoring ausgeräumt.

5.6.5 Projekt- / Codestruktur

Refactoring der Projekt- und Codestruktur

5.6.6 Ausbau GUI / Grafik

Detailimplementation des GUIs.

5.6.7 Ausbau Problem Domäne

Erweitern der Problem Domäne um weitere Funktionsmodule.

5.7 Test

5.7.1 Unit Tests

Unit Tests erstellen und damit Tests durchführen.

5.7.2 Usability Tests

Mit Hilfe von am Projekt unbeteiligten Drittpersonen Usability Tests durchführen.

5.7.3 Performance

Die Performance und die Lauffähigkeit wird auf dem iPhone 3G und 3GS mit der iPhone OS Version 3.0 getestet.

Bei Bedarf Implementierung von Performance Enhancements / Caching.

5.7.4 System Tests

Die Systemtests werden ausgearbeitet und auf dem Release Candidate ausgeführt.

5.8 Dokumentation

5.8.1 Anleitung

Benutzerhandbuch / Quick Start

5.8.2 Software Architecture Document

Erstellen des Software Architecture Documents

5.8.3 Abschlussbericht

Projektabschlussbericht erstellen.

5.8.4 Erfahrungsreport

Erfahrungen der einzelnen Projektmitglieder erfassen.

5.9 Tasks wiederkehrend

5.9.1 Studium Technologien

Einarbeiten in Mac OS X Entwicklungsumgebung, iPhone Handling, AppStore

5.9.2 Studium Cocoa Framework

Einarbeiten ins Cocoa Framework (GUI Elemente, Datenstrukturen, etc.)

5.9.3 Studium Objective C

Erlernen der Syntax, Datentypen, Memory Management, Unit Testing, Best Practices, etc.

5.10 Sitzungen

5.10.1 Sitzungsprotokoll

Vorbereiten, Erstellen und verschicken des Sitzungs

5.10.2 Besprechung mit Betreuer

Wöchentliche Meetings mit den Betreuern / dem Kunden.

5.10.3 Problemlösung

Lösen von unvorhergesehene Probleme innerhalb des Projektes.

5.11 Qualitätssicherung

5.11.1 Dokumentation und Code Style

Code Review und anpassungen (4 Augen Prinzip)

5.11.2 Arbeitsablauf diskutieren / optimieren

Analyse des Arbeitsablauf, wenn nötig Anpassen und Verfeinern.

6 Infrastruktur

6.1 Geräte

Die HSR stellt dem Projektteam während der Dauer des Projektes freundlicherweise folgende Geräte zur Verfügung:

- 1 x iPhone 3G
- 1 x iPodTouch
- 2 x iMac 24" 2.8 GHz Core 2 Duo

Zusätzlich werden die privaten Notebook verwendet.

Die Netzwerkinfrastruktur sowie der SVN Server werden ebenfalls von der HSR zur Verfügung gestellt.

7 Qualitätsmassnahmen

7.1 Dokumentation

Die werden laufend nachgeführt und aktualisiert. Sie werden zudem von den Teammitgliedern und Betreuern gegengelesen.

7.2 Reviews

In wöchentlichen Sitzung werden die erstellen Dokumente besprochen und Verbesserungsvorschläge eingebracht.

7.3 Codierrichtlinie

Bei der Programmierung werden wir uns an die von Apple vorgegebenen Codierrichtlinien halten. Eigene Richtlinien sind im "System Architecture Document" festzuhalten.

7.4 Sitzungsprotokolle & Taskliste

Die wöchentlichen Sitzungen werden protokolliert. Das Protokoll wird an alle Projektteilnehmer versandt. In jedem Protokoll sind die an der Sitzung verteilten Aufgaben und deren Status ersichtlich.

7.5 Versionierung

Als Versionierungssystem wird der von HSR zur Verfügung gestellte SVN Server verwendet. Das Repository ist unter `svn://svns.hsr.ch/Studienarbeit` erreichbar. Die Zugriffsrechte werden durch S. Hunkeler verwaltet.

7.6 Unit Testing

Domainklassen werden mit Unit Tests geprüft (Logic Tests / Clean Room Environment). Die Unit Tests werden in XCode direkt im Buildprozess integriert und werden bei jedem Ausführen mitausgeführt.

Die User Interface Komponenten werden manuell getestet.

7.7 Usability Tests

Die iPhone Applikation wird durch die Projektteilnehmer getestet. Das User Interface wird so durch kontinuierliches Feedback an den Sitzungen verbessert. Überwachte und protokollierte Usability Tests sind keine vorgesehen.

7.8 Zeitplanung

Die Zeitplanung wird in einem Excel Dokument nachgeführt (Zeitplan.xls). An den Sitzungen werden Soll- und Ist-Zustand verglichen und ggf. Massnahmen besprochen.

7.9 Performance

Als mobiles Gerät verfügt das iPhone über limitierte Ressourcen. Beim Programmieren muss daher auf effizientes Memory Management geachtet werden.

Die Performance wird erst dann als kritisch angesehen, wenn der Benutzer bei der Bedienung eingeschränkt wird und die Antwortzeiten zu gross werden.

7.10 Kompatibilität / Portabilität

Die Applikation wird im iPhone Simulator entwickelt und muss am Ende auf dem iPhone und dem iPod Touch ausgeführt werden können.

7.11 Change Management

Auf den Einsatz eines Bugtracking Systems wird bewusst verzichtet. Aufgrund der kurzen Projektdauer, regelmässigen Sitzungen und gemeinsames Programmieren würden sich Aufwand und Ertrag nicht die Waage halten.

7.12 Automatisierung

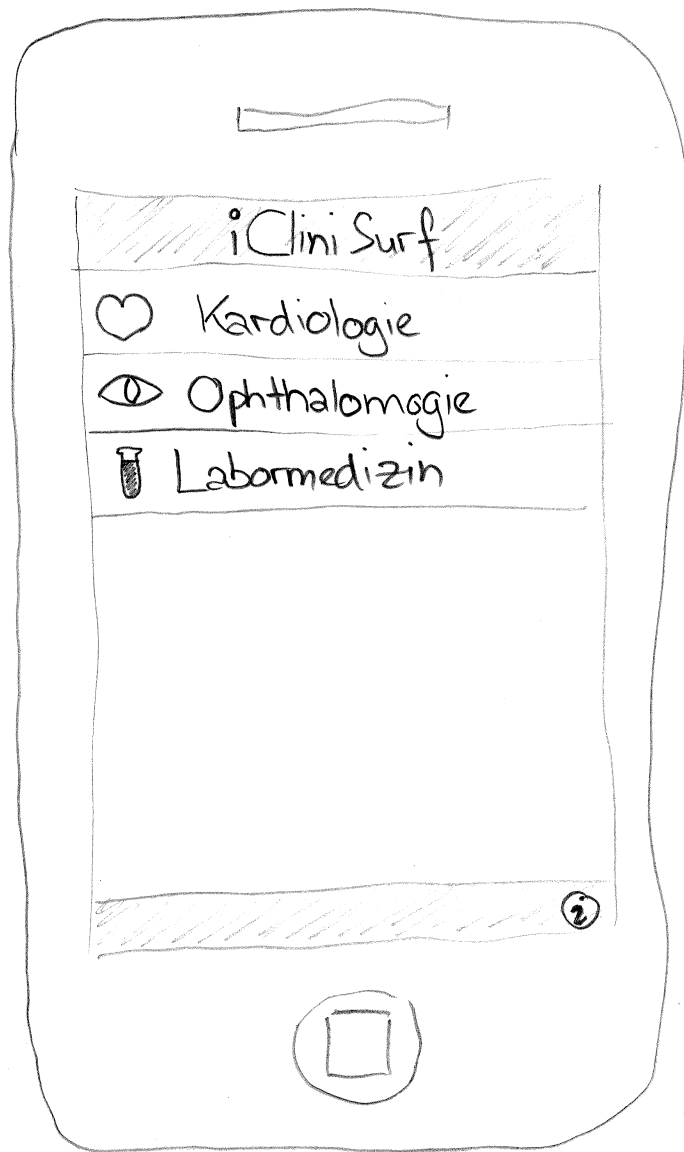
Die Automatisierung des Buildprozesses mit aktualisiertem Inhalt bzw. aktualisierte Daten welche von der Schnittstelle zum IML-Server (siehe Anforderungsspezifikationen) zur Verfügung gestellt werden, erfolgt mittels Automator¹ skript.

¹[http://en.wikipedia.org/wiki/Automator_\(software\)](http://en.wikipedia.org/wiki/Automator_(software))

Nomenclature

CR Christian Ramseier

SH Sebastian Hunkeler



1 Hauptmenü iClini Surf

- Erscheint nach dem Starten der Applikation



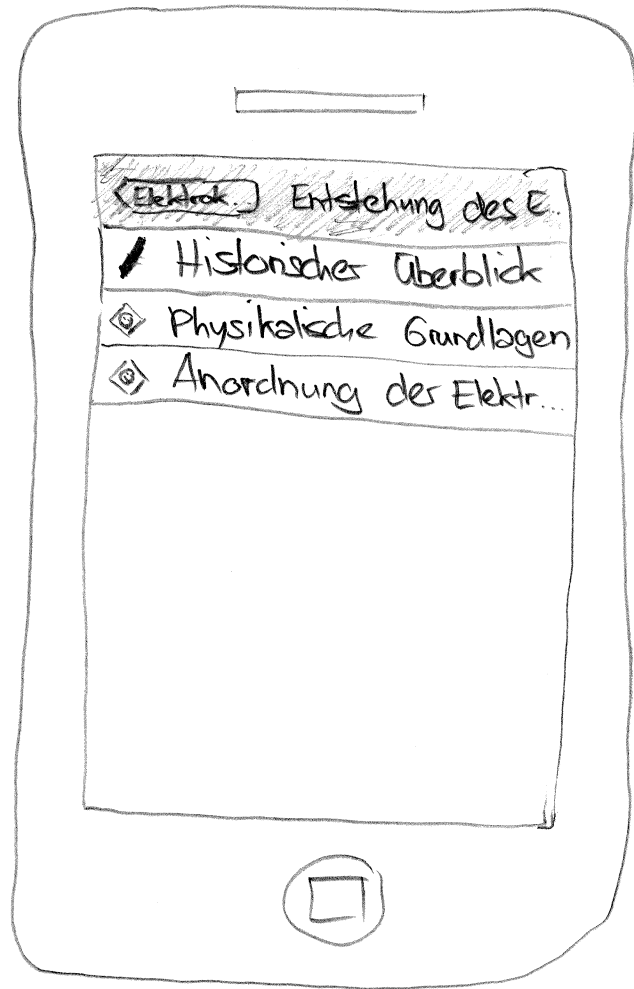
2 Untermenü

- Durch Anwählen eines Hauptmenunavigationspunktes, gelangt man in dieses Untermenü.

<u>Kardiologie</u> & Elektrokardiogra
Entstehung des EKGs
Beschreibung des EKG
Interpretation des E.
Erläuterte Beispiele
Niveau 3. Studienjahr
Niveau Staatsexamen
Niveau Internist

3 Kapitel

Hier werden die verschiedenen Kapitel der Rubrik aufgelistet.



4 Kapitelübersicht

- Zeigt die einzelnen Lerneinheiten eines Kapitels an.
- Das Icon informiert den Benutzer über die zu erwartende Funktionalität der entsprechenden Lerneinheit.

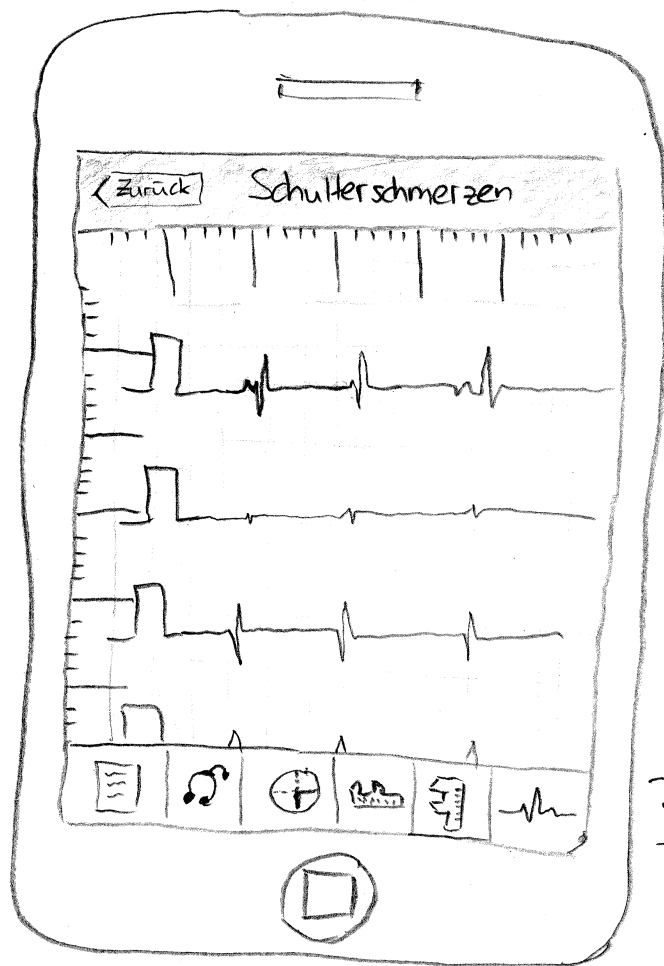
Beispiel:

EKG Icon - Lupenfunktion, systematische Befundung



5 Modalität "Abbildung", "Grafik" und "Fotografie"

- Das angezeigte Bild lässt sich in gewohnter iPhone-Manier mit zwei Fingern vergrößern und verkleinern.
- Das Text-Scrolling wird ebenfalls durch die "drag"-Funktion des iPhone ermöglicht.



6.0 Elektrokardiogramm

- Wird eine Seite des Funktionsmoduls "EKG" angewählt, wird dem Benutzer zunächst einmal das Diagramm präsentiert.
- In der Toolbar stehen folgende Funktionen zur Verfügung:

☰ Aufgabenstellung

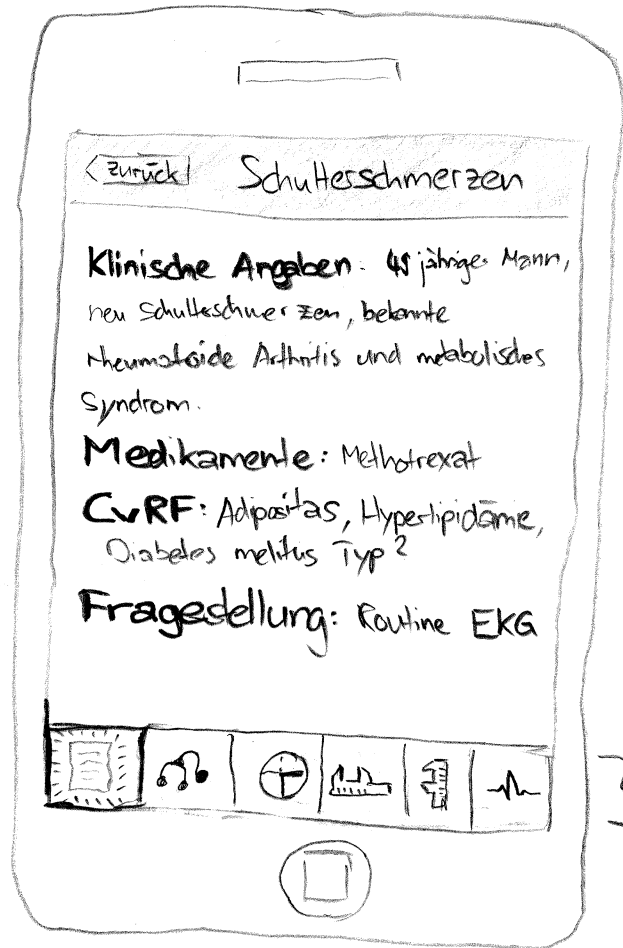
🎵 Befundung

⊕ Winkelmesser

📏 Horizontale Messung

📏 Vertikale Messung

📏 Referenzbilder



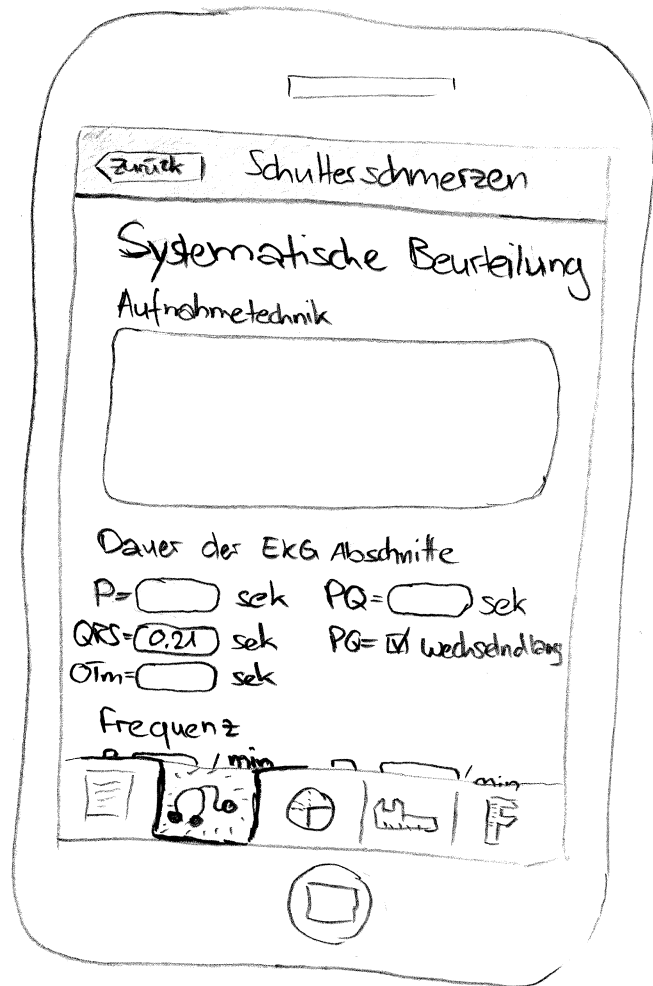
6.1 Aufgabenstellung (Variante 1)

- Nach dem Antippen des Aufgabenstellungssymbols erscheint diese Ansicht.
- Durch erneutes Antippen des Aufgabenstellungssymbols oder durch Wählen eines anderen Werkzeugs verschwindet die Aufgabenstellung wieder.



6.1 Aufgabenstellung (Variante 2)

- o Nach dem Antippen des Aufgabenstellung-Symbols erscheint diese Anzeige.
- o Mit "Fertig" kann man zum EKG-Bild zurückwechseln.



6.2 Systematische Befundung (Variante 1)

- Auf dieser Anzeige trägt der Benutzer über die iPhone-übliche Touch-Tastatur seinen Befund ein.
- Die Dauern der EKG Abschnitte können mit dem Vermessungswerkzeug automatisch in dieses Formular übernommen werden.
- Durch erneutes Anwählen des Befundungssymbols oder durch Wählen eines anderen Werkzeugs wird dieser Screen verlassen.

Systematische Befundung **Fertig**

Aufnahmetechnik

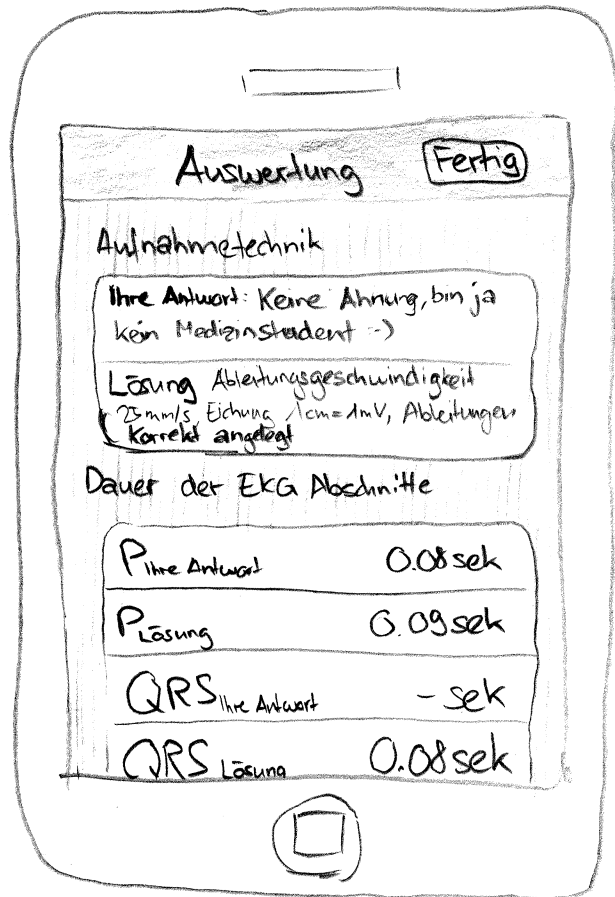
Dauer der EKG Abschnitte

P	0.08 sek	?
QRS	0.10 sek	?
QTm	sek	?
PQ	sek	?

Auswerten

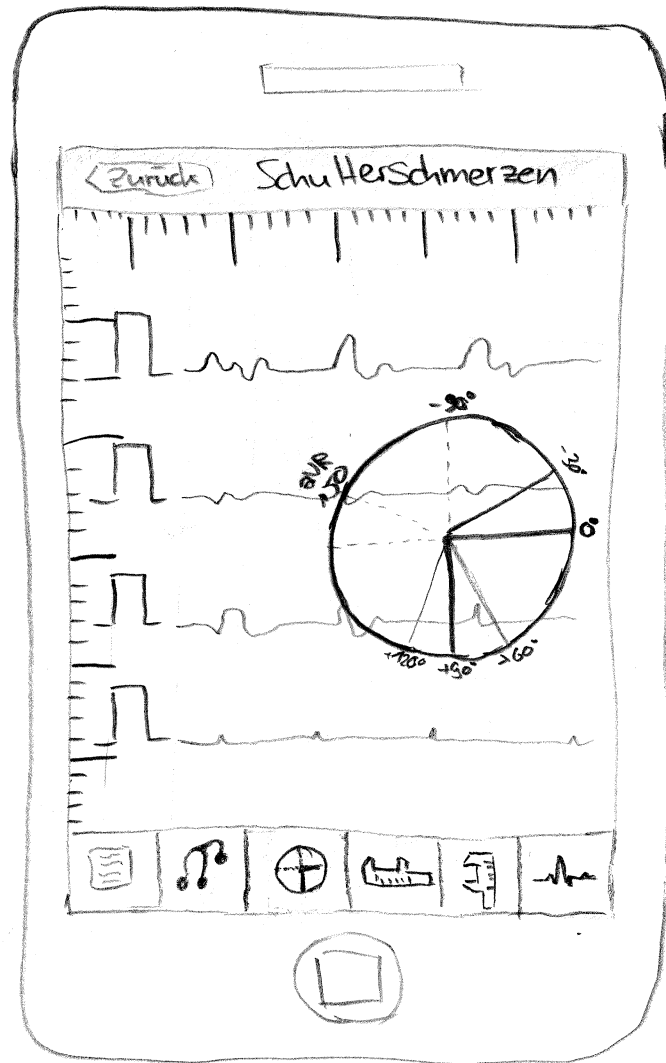
6.2.1 Systematische Befundung (Variante 2)

- o Auf diesem Screen trägt der Benutzer über die iPhone-übliche Touch-Tastatur seinen Befund ein.
- o Die Dauer der EKG Abschnitte können mit dem Vermessungswerkzeug automatisch in dieses Formular übernommen werden.
- o Die Ansicht wird über "Fertig" verlassen.



6.2.2. Auswertung des Befunds

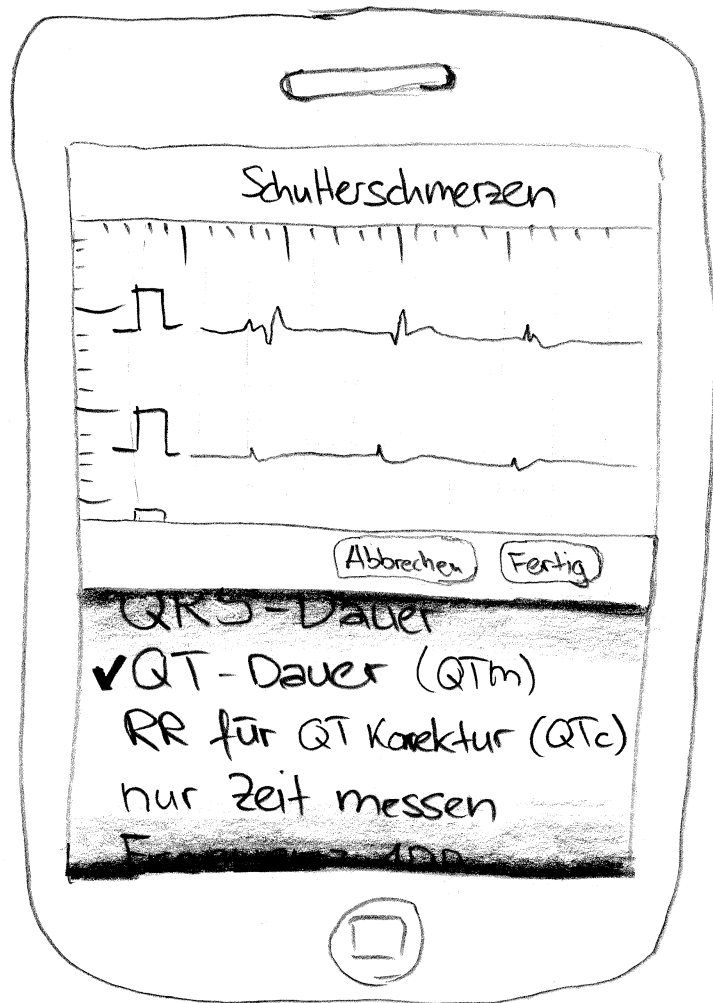
- o In dieser Ansicht kann der Benutzer seinen Befund mit der Musterlösung vergleichen
- o Über "Fertig" kommt er wieder in die Startansicht der Lerneinheit (6.0).



} Toolbar

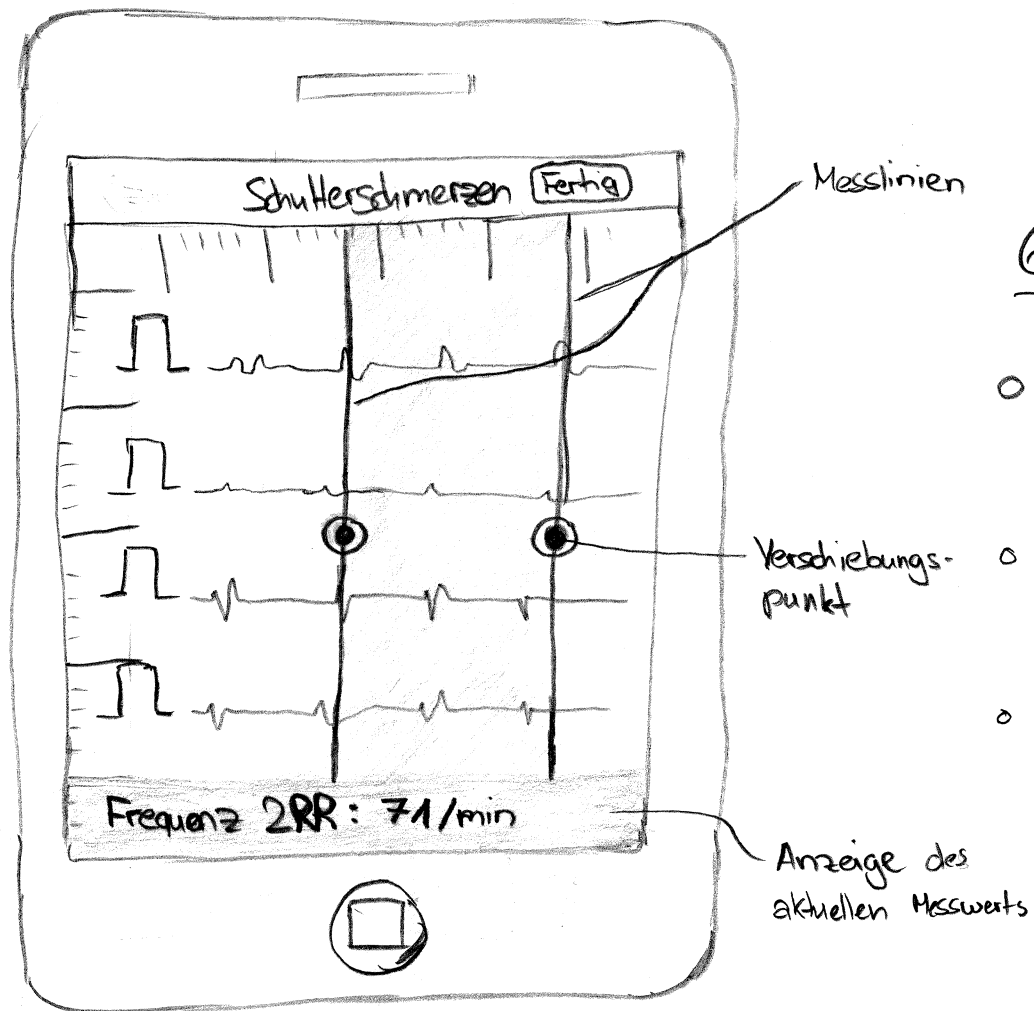
6.3 Winkelmesser

- Der Winkelmesser kann über ein Symbol in der Toolbar ein- und ausgeblendet werden.



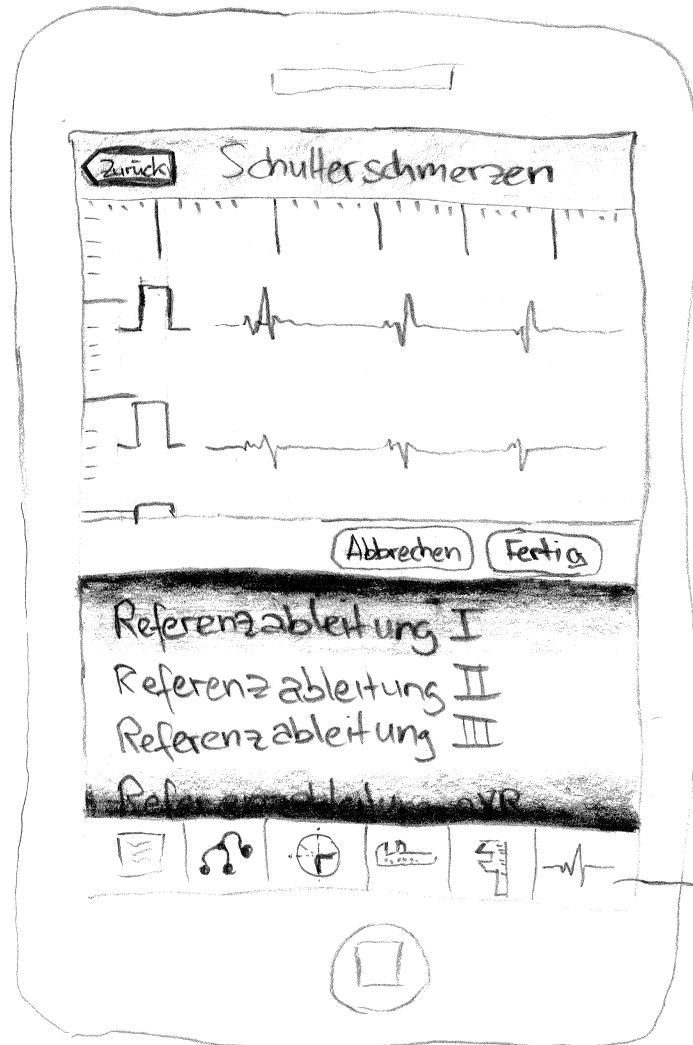
6.4.1 Vermessungswerkzeug

- Beim Auswählen eines Vermessungswerkzeuges in der Toolbar, muss zunächst einmal angegeben werden, was gemessen werden soll.
- Über die Wahl von "Fertig" gelangt der Benutzer zu 6.4.2.



6.4.2 Vermessungswerkzeug

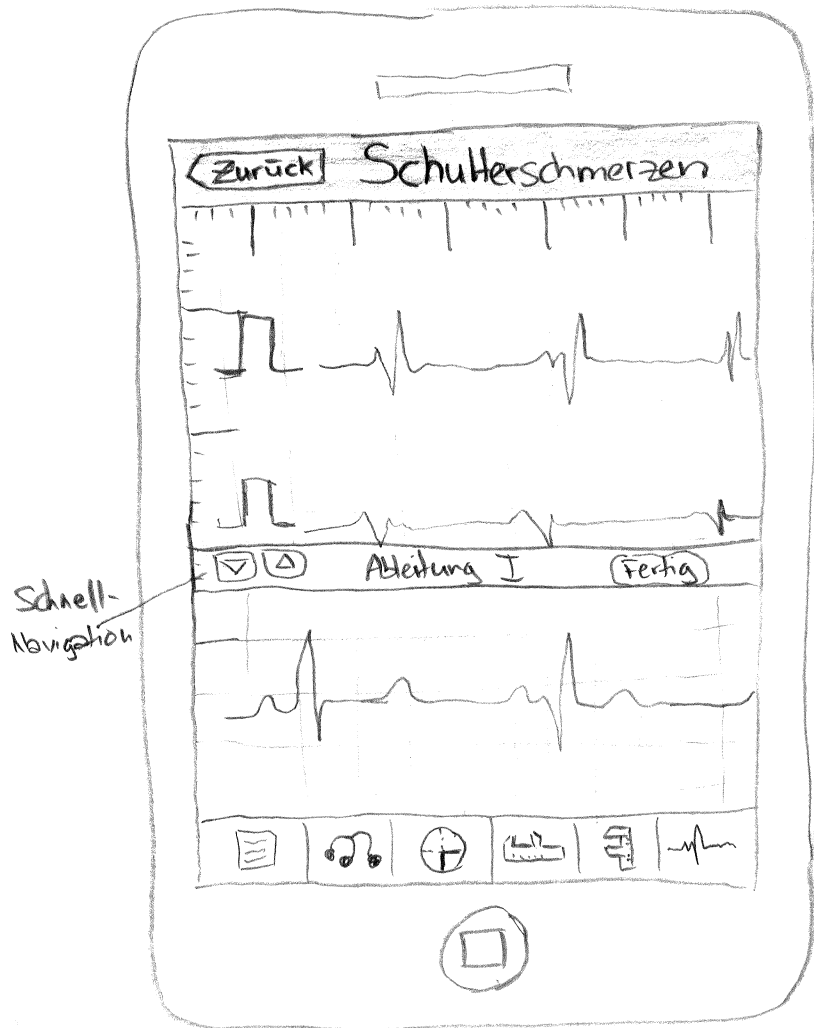
- Mit Hilfe der Verschiebungspunkte können die Messlinien verschoben werden.
- Am unteren Rand wird der aktuell gemessene Wert angezeigt.
- Über "Fertig" kann der Messwert in den Systematischen Befund übernommen werden.



6.5.1 Auswahl Referenzbilder

- Über das Icon "Referenzbilder auswählen" wird dem Benutzer die Auswahl der verschiedenen Ableitungen präsentiert.

Referenzbilder
auswählen



6.5.2 Referenzableitung

- Durch das Einblenden einer Referenzableitung kann der Benutzer einen Vergleich machen.
- Über die "Schnellnavigation" kann der Benutzer komfortabel zwischen den Referenzbildern wechseln.

iPhone Applikation iMedLearn

Version: 0.4

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

Änderungsgeschichte

Datum	Version	Änderung
5.10.2009	0.1	Erstellung des Dokuments
8.10.2009	0.2	Review & Ergänzungen
11.10.2009	0.3	SSDs ergänzt
12.10.2009	0.4	State Diagramm hinzugefügt

Referenzen

- [documentation/Analyse/paper-prototype-v2.pdf](#)
- [documentation/Analyse/anforderungsspezifikation.pdf](#)

Contents

1	Einleitung	2
2	Personas	3
2.1	Medizinstudent (Primäre Persona)	3
2.2	Arzt	4
3	Use Cases	5
3.1	Brief Use Cases	5
3.2	Use Case Diagramm	6
4	Domain Modell	7
4.1	Strukturdiagramm	7
4.2	Konzeptbeschreibung	7
5	System Sequenzdiagramme	8
6	Contracts	9
6.1	Slide laden / anzeigen	11
6.2	Slide verlassen / wechseln	11
6.3	Messung übernehmen	11
7	State Diagram	11
8	Logische Schichten	11
9	Datenanalyse	13
10	Deployment	15
11	Graphical User Interface	15
11.1	Überblick	15
11.2	Paper Prototype	15
11.3	User Environment Model	15

1 Einleitung

Dieses Dokument dient der Domain-Analyse der zu entwickelnden iPhone Applikation, und enthält die erarbeiteten Artefakte der OOA Phase.

In den nachfolgenden Punkten wird spezifisch auf die einzelnen Analyse-Artefakte der Domain eingegangen. Zuerst werden die Personas und die Use Cases beschrieben. Danach wird auf das Domain Model welches die Problem-domäne abbildet, und dessen Konzepte eingegangen. Dann folgen die Sequenzdiagramme für Abläufe im System als „Blackbox“ und State-Diagramme für die Zustände der Applikation. Zudem werden Contracts für die wichtigsten Systemfunktionen aufgezeigt. Anschliessend folgt noch die Beschreibung des

Deploymentvorganges der zu entwickelnden Applikation und eine Analyse der Datenstruktur. Zum Schluss wird noch das Benutzerinterface analysiert.

2 Personas

2.1 Medizinstudent (Primäre Persona)



Figure 1: Yves Wiedemeier

Yves Wiedemeier ist 24 Jahre alt und studiert zur Zeit Medizin an der Universität in Bern. Er ist technisch auf dem neusten Stand. Er ist oft unterwegs und nutzt sein Mobiltelefon um Wartezeiten zu überbrücken indem er sich die aktuellen Unterrichtsmaterialien herunterlädt und studiert. Zu Hause arbeitet er oft an seinem Computer und nutzt die vielfältigen E-Learning Angebote, welche es im Internet zu finden gibt. Seine Lehrbücher nimmt er nur mit, wenn er diese zum arbeiten an der Universität braucht.

2.2 Arzt



Figure 2: André Pauli

Herr André Pauli ist seit 20 Jahren praktizierender Arzt. Durch stetige Fortbildung hat er mittlerweile ein sehr grosses Fachwissen erlangt. Auch im Bereich Informatik ist er mit der Zeit gegangen und nutzt das Internet oft für Recherchen oder um sein medizinisches Wissen aus früherer Zeit wieder aufzufrischen.

3 Use Cases

3.1 Brief Use Cases

Nr.	Use Case	Beschreibung
1	Text / Bild Slide begutachten	Der Benutzer liest die Text / Bild Artikel und eignet sich so die nötigen Vorkenntniss an um die Aufgaben lösen zu können.
2	EKG Aufgabenstellung lesen	Der Benutzer liest die Aufgabenstellung um herauszufinden auf was er bei der anschliessenden Befundung achten muss.
3	EKG Slide befunden	Das EKG wird befundet indem der Benutzer sich die notwendigen Informationen mit Hilfe der vorhandenen Werkzeuge beschafft.
4	Referenzbilder begutachten	Der Benutzer vergleicht das vorliegende EKG mit dem Referenzbild, um auf Grund der Abweichungen Aufschluss über das Patientenbild zu erhalten. Dies ist nützlich für die Befundung.
5	EKG Werte messen	Die EKG Abschnitte werden mit Hilfe von verschiedenen Messwerkzeugen vermessen. Die gemessenen werte können dann im Befund verwendet werden.
6	Befund ausfüllen	Der Befund wird gemäss der Aufgabenstellung ausgefüllt.
7	Befund auswerten	Der Benutzer vergleicht die Angaben aus seinem Befund mit der Musterlösung.

3.2 Use Case Diagramm

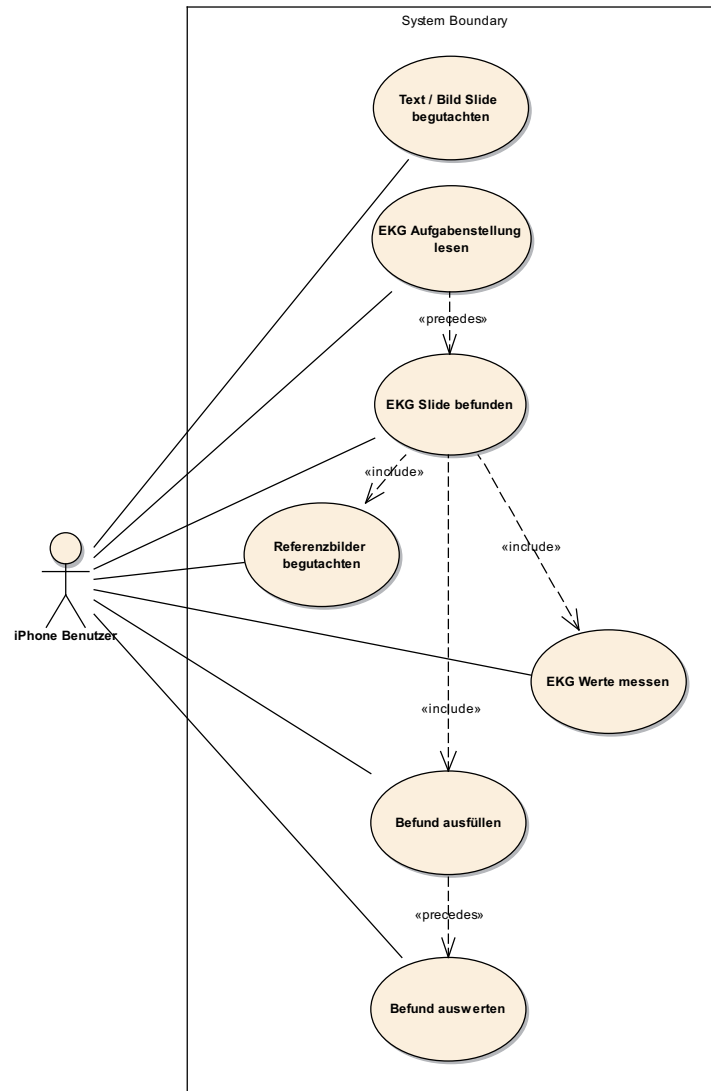


Figure 3: Use Case Übersicht

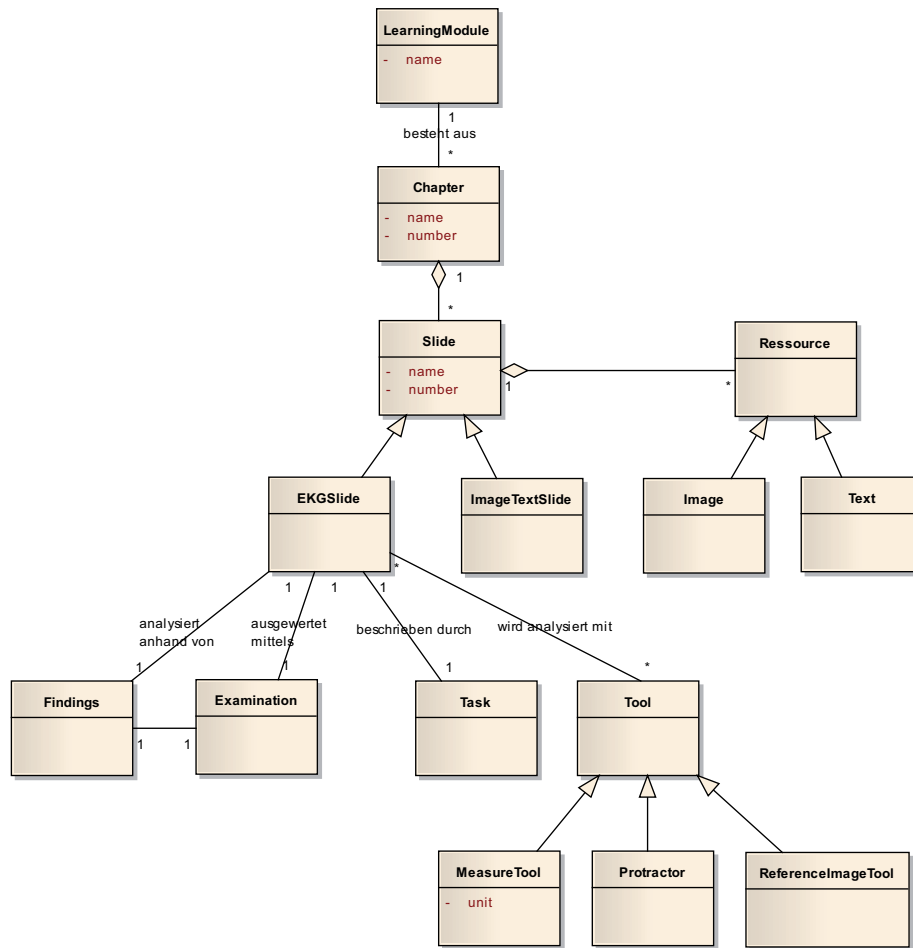


Figure 4: Strukturdiagramm Domain Model

4 Domain Modell

4.1 Strukturdiagramm

4.2 Konzeptbeschreibung

Nachfolgend werden die Konzepte / Entitäten des Domain-Modells erörtert.

- Learning Module: Diese Einheit dient der Abbildung eines Themengebietes wie z.B. Elektrokardiogramm und entspricht in unserem Fall der Applikation selber.
- Chapter: Chapters definieren die Menüstruktur und ermöglichen wie es der Name schon sagt, die Lerninhalte inKapitel zu gliedern. Ein Chapter

kann wiederum selber Chapters beinhalten (Composite¹).

- Slide:
 - EKG Slide: Ein EKG Slide entspricht einer Übungsaufgabe und stellt eine EKG Grafik dar. Es verwaltet die Aufgabenstellung, Befundung und Auswertung dieser Aufgabe. Ausserdem verwaltet es die Messwerkzeuge die verwendet werden können um die Aufgabe zu lösen.
 - Image Slide: Ein Image Slide kann Bilder und Text darstellen. Es stellt die für die Lösung der EKG Befundungs-Aufgaben nötige Theorie zur Verfügung.
- Ressource:
 - Image: Ein Slide (siehe oben) kann mehrere Ressourcen beinhalten, wovon die Bilder durch die Image Entität repräsentiert werden.
 - Text: Entspricht dem formatierten Text welcher ein Slide beinhalten kann.
- Task: Der Task definiert die Aufgabenstellung, auf Grund der dann ein Befund erstellt werden kann.
- Findings: Die Findings Entität verwaltet die persönlichen Lösungen zu den Aufgaben welche man vorzu erarbeitet.
- Examination: Die Examination enthält die Musterlösungen zu den Aufgaben. Diese können mit dem Befund des Benutzers verglichen werden.
- Tool:
 - MeasureTool: Die Measure Tools (Messwerkzeuge) dienen dazu die Messwerte zu erarbeiten und somit den Befund ausfüllen zu können.
 - Protractor: Der Protractor (Winkelmesser) dient dazu den Lagetyp und die Übergangzone zu Bestimmen. Diese Werte können dann im Befund übernommen werden.
 - ReferenceImageTool: Das Reference Image Tool stellt die Referenzbilder für den EKG Slide zur Verfügung. So kann können die Ableitungen des EKG's mit den Referenzableitungen verglichen werden.

5 System Sequenzdiagramme

Es werden nachfolgende die System Sequenzdiagramme für die beiden Use Cases «Text / Bild Slide begutachten» und «EKG Werte messen» dargestellt.

¹http://en.wikipedia.org/wiki/Composite_pattern

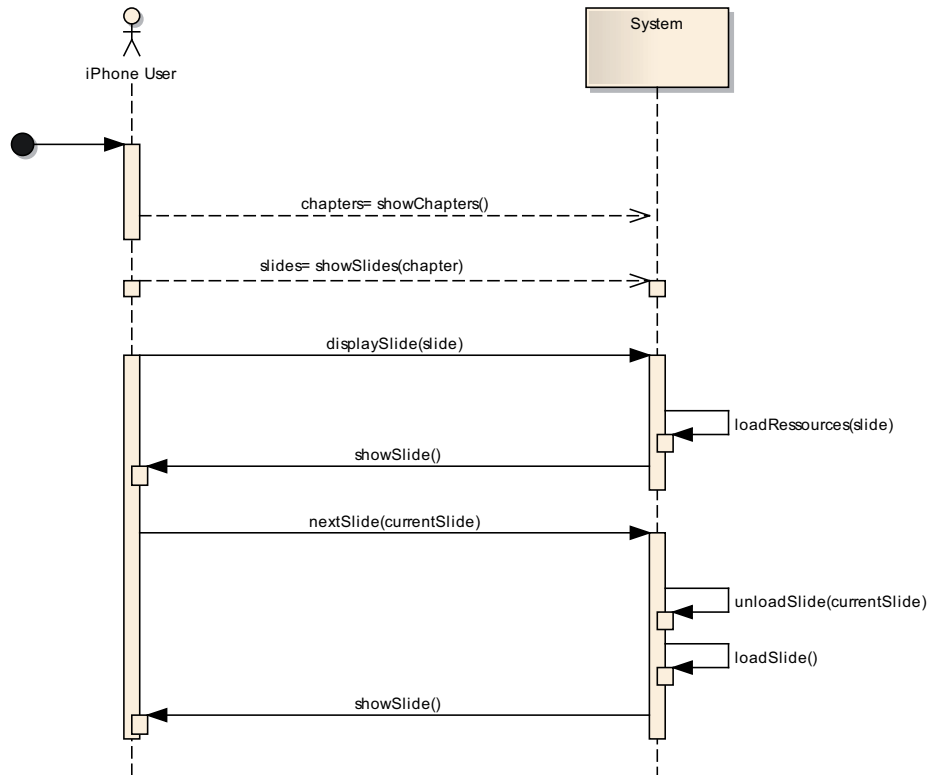


Figure 5: Sequenzdiagramm “Text / Bild Slide begutachten”

Slide Load SSD: Auf diesem Diagramm wird gezeigt wie der Anzeigevorgang eines Slides funktioniert. Zusätzlich zur Use Case Beschreibung vom Abschnitt «Use Cases» wird auf diesem Diagramm zum nächsten Slide gewechselt.

Measuretool SSD: Auf diesem Diagramm wird der Messvorgang am Beispiel QRS dargestellt. Das QRS Messwerkzeug wird ausgewählt, der QRS Wert gemessen und anschliessend in den Befund übertragen.

6 Contracts

Contracts für Systemfunktionen um den genauen vor-/nachher Stand des Systems aufzuzeigen.

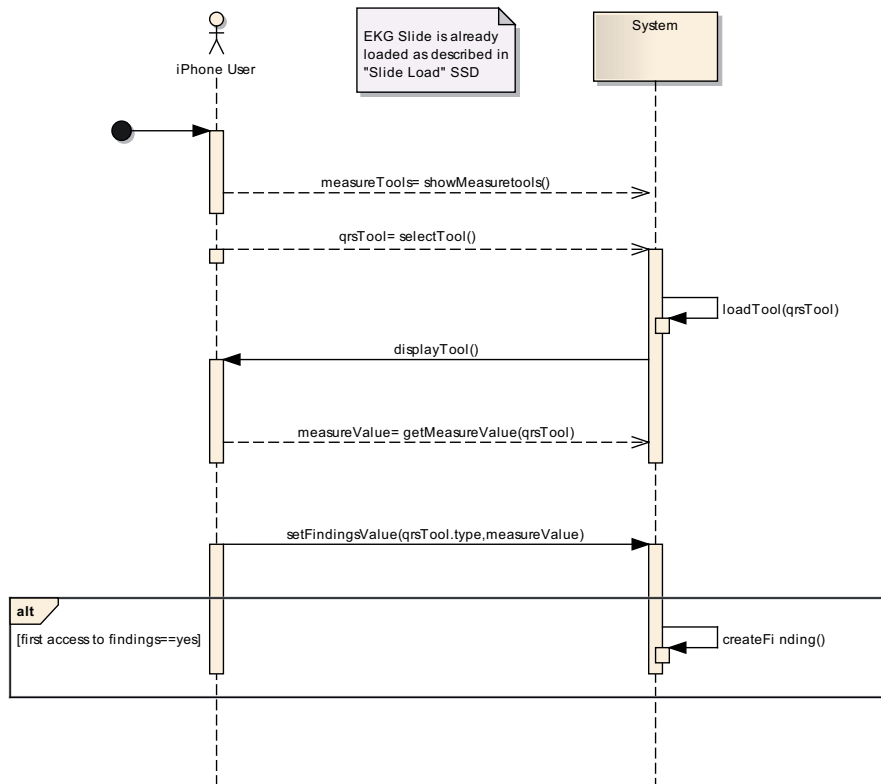


Figure 6: Sequenzdiagramm "Slide Load"

6.1 Slide laden / anzeigen

Operation	showSlide()
Reference	UC1 Text / Bild Slide begutachten
Preconditions	Benutzer hat Slide ausgewählt
	zusätzliche Ressourcen für den gewählten Slide wurden noch nicht geladen
	Slide selber ist vorausgeladen
Postconditions	zusätzliche Ressourcen für den gewählten Slide wurden nachgeladen
	Slide wird angezeigt

6.2 Slide verlassen / wechseln

Operation	leaveSlide()
Reference	UC1 Text / Bild Slide begutachten
Preconditions	Benutzer hat Slide ausgewählt und Slide wird angezeigt
	zusätzliche Ressourcen für den gewählten Slide wurden geladen
	Das Benutzerinterface wurde für den Slide angepasst
Postconditions	Benutzer hat Slide verlassen
	die zusätzlich geladenen Ressourcen wurden wieder freigegeben
	Status des Benutzerinterfaces wurde wiederhergestellt

6.3 Messung übernehmen

Operation	finishMeasurement()
Reference	UC5 EKG Werte messen, UC6 Befund ausfüllen
Preconditions	Benutzer befindet sich im Messmodus
	Benutzer hat Messwert gemessen
Postconditions	Messwert wurde in Befund übernommen
	Messmodus wurde verlassen

7 State Diagram

Logischer Statusablauf des Mess- und Befundungsvorgangs bei einem EkgSlide. Das Diagramm zeigt auf, dass die Mess- und Befundungsvorgänge modal stattfinden, sodass dem Benutzer immer klar ist was er gerade tun kann und vor allem tun muss.

8 Logische Schichten

Logische Schichtendarstellung für das GUI, die Business Logik und die Datenhaltung im Zusammenspiel mit dem Cocoa Touch Framework, welches auf dem iPhone zur Verfügung steht.

GUI Schicht: Diese Schicht ist zuständig für die Darstellung des GUIs und das Tracking der Benutzereingaben.

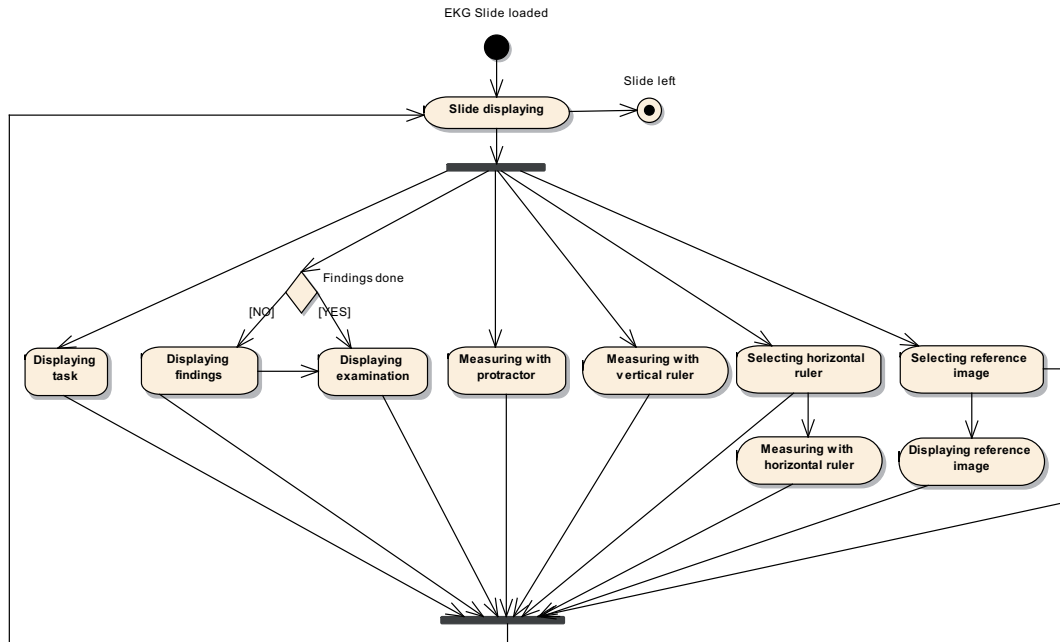


Figure 7: State Diagramm "EkgSlide"

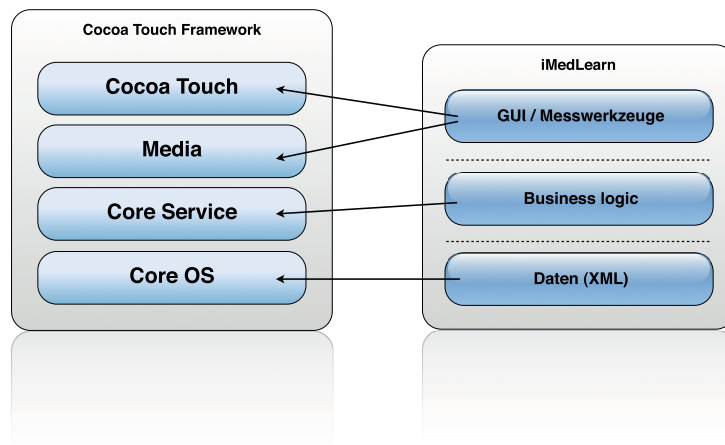


Figure 8: Logische Schichten

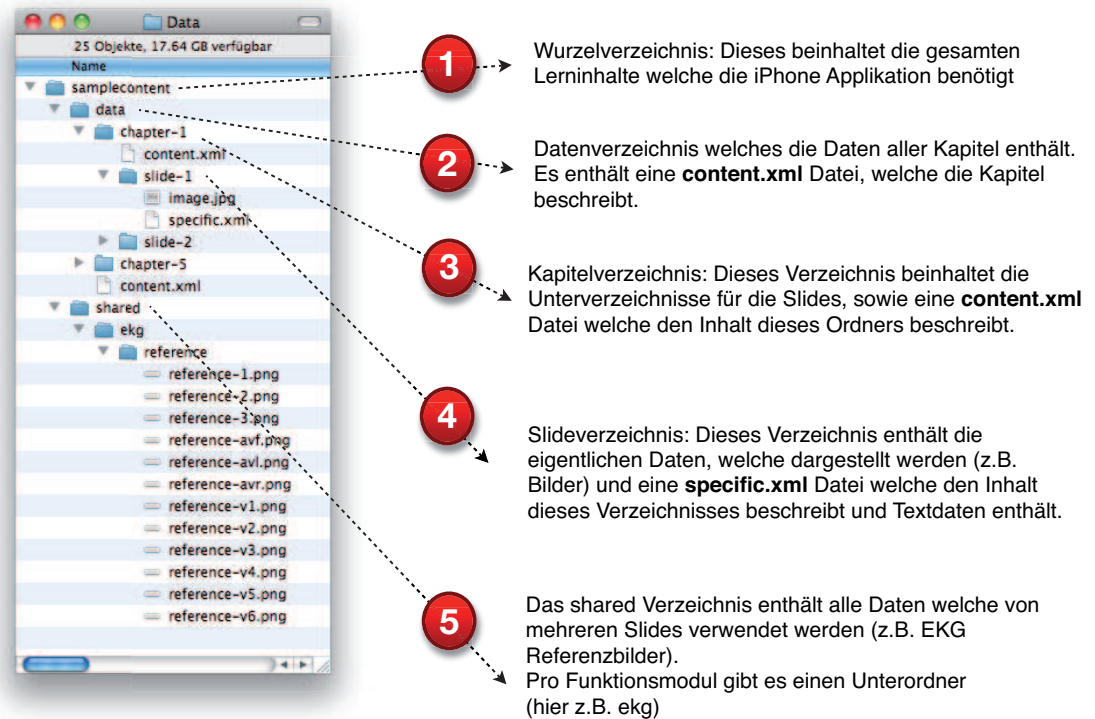


Figure 9: Beispiel Datenstruktur

Business logic: Diese Schicht enthält die Businesslogik, die unter anderem zuständig ist für die Auswertung der EKGs und die verschiedenen Berechnungen.

Datenhaltungsschicht: Diese Schicht ist zuständig um die Daten welche von der Datenbankschnittstelle (siehe Anforderungsspezifikationen) zur Verfügung gestellt werden einzulesen.

9 Datenanalyse

Um die Schnittstelle zu definieren wurde eine Beispiel-Ordnerstruktur erstellt, welche die relevanten XML Dateien und Bilder enthält.

Beispiel für Datei- und Ordnerstruktur

Listing 1: Auszug aus samplecontent/data/content.xml:

```
<content>
<items>
<item>
<name>chapter-1</name>
<title>Entstehung des EKGs</title>
<type>navigation</type>
```

```

</item>
<item>
  <name>chapter-2</name>
  <title>Beschreibung des EKGs</title>
  <type>navigation</type>
</item>
</items>
</content>

```

Listing 2: Auszug aus samplecontent/data/chapter-1/content.xml:

```

<content>
  <items>
    <item>
      <name>slide-1</name>
      <title><![CDATA[Historischer Überblick]]></title>
      <type>imageslide</type>
    </item>
    <item>
      <name>slide-2</name>
      <title><![CDATA[Physikalische Grundlagen]]></title>
      <type>imageslide</type>
    </item>
  </items>
</content>

```

Listing 3: Auszug aus samplecontent/data/chapter-1/slide/specific.xml:

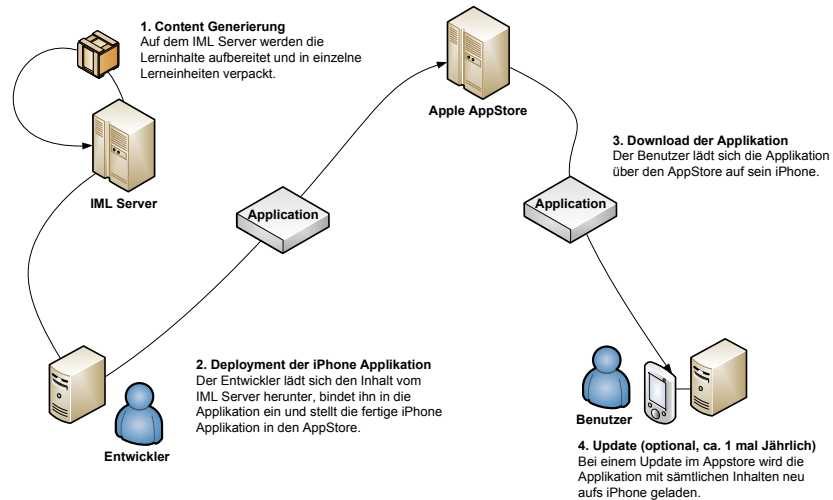
```

<slide>
  <image>image.jpg</image>
  <content><![CDATA[Der Brite Augustus Desire Waller zeichnete 1893 das
erste EKG am Menschen auf.]]></content>
</slide>

```

10 Deployment

Deployment Diagram



11 Graphical User Interface

11.1 Überblick

Die GUI-Analyse zeigt den angestrebten Aufbau der Benutzeroberfläche mit logischem Ablauf. Die genaue Definition erfolgt während der Construction in einem Designphase. Es wird hier lediglich ein struktureller und logischer Aufbau dargestellt.

11.2 Paper Prototype

Siehe paper-prototypes-v2.pdf

11.3 User Environment Model

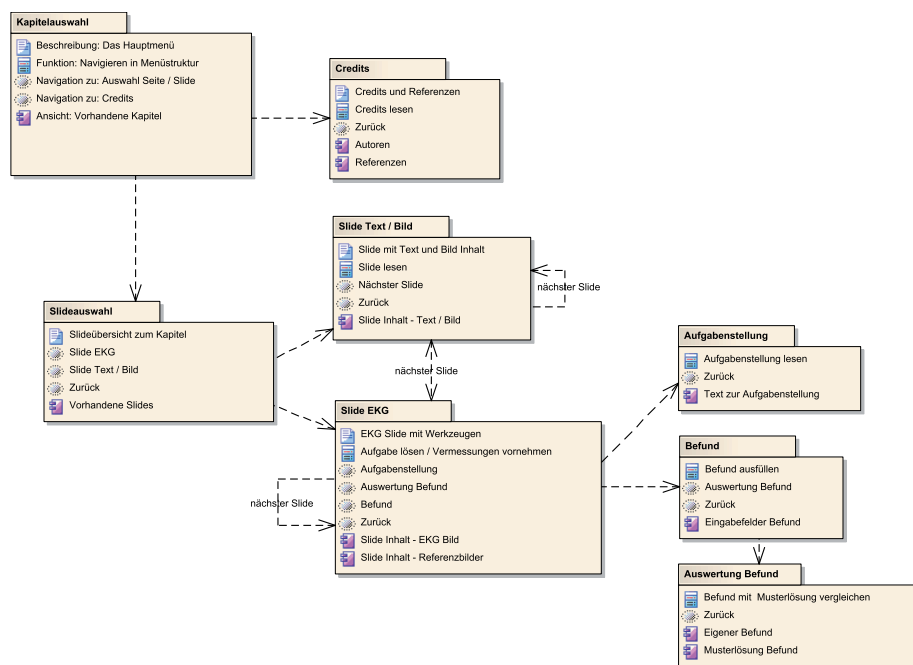


Figure 10: User Environment Model

Datenbankerweiterung iPhone Applikation iMedLearn

December 18, 2009

Version: 1.0

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
17.11.2009	0.1	Erstellung	SH,CR
24.11.2009	1.0	BOOL nach TINYINT	CR

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Ausgangslage	2
3	Vorschlag	2
4	Implementierung	2

2 Ausgangslage

Im Rahmen der Entwicklung einer iPhone Applikation sollen Lerninhalte aus der Datenbank des IML gelesen und für das iPhone aufbereitet werden. Da sich nicht alle Slides / Chapters für das iPhone eignen, macht es Sinn eine Differenzierung direkt auf der Datenbank vorzunehmen.

3 Vorschlag

Auf Ebene der Slides und der Chapters wird je ein zusätzliches Feld eingefügt, welches die iPhone Kompatibilität anzeigt. Der Vorschlag ist absichtlich so gewählt, dass er nur einen minimalen Eingriff in die Datenbankstruktur vorsieht. Hier der Vorschlag als SQL Statement:

- ALTER TABLE 'iml'. 'chapters' ADD COLUMN 'iphonecompatible' TINYINT(1) NOT NULL DEFAULT 1 AFTER 'filename';
- ALTER TABLE 'iml'. 'slides' ADD COLUMN 'iphonecompatible' TINYINT(1) NOT NULL DEFAULT 1 AFTER 'usemap';

4 Implementierung

Hat das Feld **iphonecompatible** den Wert **TRUE** (bzw. "1"), wird das Chapter / Slide vom Paketierungsscript in die iPhone Applikation übernommen. Ansonsten wird das Chapter / Slide ignoriert und ist anschliessend nicht auf dem iPhone verfügbar.

Die Navigation wird Top-Down ausgelesen. Das heisst wenn ein Kapitel als nicht iPhone kompatibel markiert ist, wird es ignoriert und nicht weiter verfolgt. Alle dem Kapitel untergeordnete Slides werden folglich ebenfalls ignoriert.

System Architecture Document

iPhone Applikation iMedLearn

December 18, 2009

Version: 0.7

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
17.11.2009	0.1	Erstellung	SH,CR
4.12.2009	0.2	Kapitel Datenexport erstellt	CR
		Kapitel Qualitätssicherung, Kapitel Wartbarkeit erstellt	SH
5.12.2009	0.3	Kapitel Datenexport erweitert	CR
		Kapitel Programmierung erweitert	SH
15.12.2009	0.4	Kapitel Datenexport fertiggestellt, Grafikkonzepte erweitert	CR
16.12.2009	0.5	Architektonische Darstellungen eingefügt	CR
16.12.2009	0.6	Programmierung und CodeStyle ergänzt	SH
16.12.2009	0.7	Qualitätssicherung und Wartbarkeit ergänzt	SH

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Referenzen	4
3	Einführung	4
3.1	Zweck	4
3.2	Begriffe	4
4	Grafikkonzepte	4
4.1	Ziele	4
4.2	Navigation	5
4.3	ImageSlide	6
4.3.1	Hauptansicht	6
4.3.2	Vollbild	6
4.4	EkgSlide	7
4.4.1	Zooming, Paning & die Werkzeugleiste	7
4.4.2	Aufgabenstellung	8
4.4.3	Befund	9
4.4.4	Auswertung	10
4.4.5	Winkelmesser	11
4.4.6	Vermessungswerkzeuge	12
4.4.7	Referenzbilder	13
4.5	Konzeptionelle Entscheidungen	13
5	Architektur	14
5.1	Ziele	14
5.2	Übersicht	14
5.3	Schichtenmodell	15

5.4	Domain	15
5.5	ViewControllers	16
5.6	Table View Elemente	17
5.7	Strategien und Konzepte	18
5.7.1	Composite	19
5.7.2	Lazy Acquisition	19
5.7.3	Before / After	20
5.7.4	Eager Acquisition	22
5.7.5	Factory / Factory method	22
5.7.6	Pooling	22
5.7.7	MVC	24
6	Programmierung	25
6.1	Programmiersprachen & Frameworks	25
6.2	Kompatibilität	25
6.3	Libraries	25
6.4	Code style	26
6.5	Debugging	29
7	Datenexport & Aufbereitung	29
7.1	Problemstellung	29
7.2	Realisierung	29
7.3	Export & Parsing	30
7.3.1	Realisierung	30
7.3.2	Konfiguration	30
7.3.3	Skriptaufruf & Ablauf	31
7.3.4	Musterformatierung "EKG Befund"	31
7.3.5	Besonderheiten bei der Formatierung	31
7.3.6	Validierung & Automatische Korrekturen	32
7.3.7	Reporting	32
7.3.8	Download	34
8	Schnittstellendefinition	34
8.1	Dateistruktur	34
8.2	Beispielstruktur	34
8.3	Schemata	35
8.3.1	Generelles Schema (content.xml)	35
8.3.2	Text / Bild spezifisches Schema (specific.xml)	36
8.3.3	EKG spezifisches Schema (specific.xml)	36
9	Qualitätssicherung	38
9.1	Unit testing	38
9.2	GHUnit	39
9.3	Testprotokolle	40
9.4	Unit Testing setup	40
9.5	Static Analyzer	41

9.6	Instruments	41
10	Wartbarkeit	42
10.1	Ressourcenverwaltung	42
10.1.1	Deployer	42
10.1.2	HTML Inhalte	44
10.1.3	String Ressourcen	45
10.1.4	Konfiguration	46

2 Referenzen

- [documentation/Analyse/domainanalyse.pdf](#)
- [documentation/Tests/Usability Test - A.Li.pdf](#)
- [documentation/Tests/Usability Test - J.Hunkeler.pdf](#)
- [documentation/Tests/Usability Test - K.Li.pdf](#)
- [documentation/Tests/System Tests.pdf](#)
- [documentation/Glossar/glossar.pdf](#)
- Patterns for Fault Tolerant Software - Robert S. Hanmer (ISBN-13: 978-0-470-31979-6)

3 Einführung

3.1 Zweck

Dieses Dokument soll zukünftigen Entwicklern und Projektmitarbeitern den Einstieg erleichtern sowie Zusammenhänge innerhalb der Architektur aufzeigen.

3.2 Begriffe

Die Begrifflichkeiten sind im Dokument “Glossar” festgehalten.

4 Grafikkonzepte

4.1 Ziele

Grundsätzlich sollen in der Benutzeroberfläche wann immer möglich iPhone Standardcontrols verwendet werden. Damit soll erreicht werden, dass der Benutzer sich schnell zurechtfindet.

Solches iPhone Standardverhalten soll auch bei der Navigation und dem Handling (Fingergesturen auf dem Multi-Touch-Display) anzutreffen sein.

4.2 Navigation



Abbildung 1: Navigation durch ein Kapitel

Für die Navigation durch die einzelnen Kapitel wurde eine Standardtabelle verwendet. Durch die verschiedenen Icons wird dem Benutzer die Art des Inhalts bzw. das Funktionsmodul signalisiert.

Um auf die nächst höhere Navigationsebene zu gelangen wird die Leiste am oberen Rand benutzt. Dieses Konzept ist auch in den einzelnen Slides konsistent realisiert.

4.3 ImageSlide

4.3.1 Hauptansicht



Abbildung 2: Hauptansicht ImageSlide

Auf jedem ImageSlide wird eine hochauflösende Grafik zusammen mit HTML Inhalten präsentiert. Die Grafik kann dabei beliebig skaliert werden, ohne dass sich dabei die Inhaltsansicht in der Grösse verändert. Diese bleibt immer der Bildschirmbreite angepasst und ist so für den Benutzer optimal navigierbar.

4.3.2 Vollbild



Abbildung 3: Bild im Vollbildmodus

Falls für die detaillierte Betrachtung der Grafik der ganze Bildschirm benötigt wird, lässt sich das Bild durch doppeltes antippen im Vollbildmodus betrachten. Hier ist sowohl eine horizontale als auch eine vertikale Ansicht möglich. Die Umschaltung erfolgt über das beim iPhone übliche Drehen des Gerätes. Die Navigation zurück in die Hauptansicht ist auch hier konsistent mit den anderen Ansichten realisiert.

4.4 EkgSlide

4.4.1 Zooming, Panning & die Werkzeugleiste

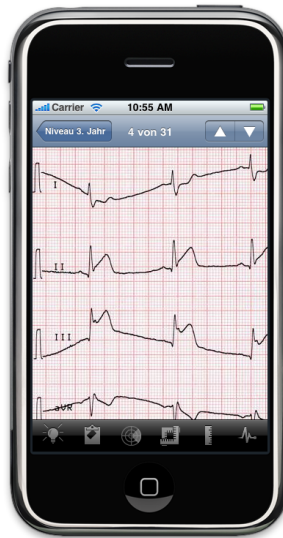


Abbildung 4: Starbildschirm EkgSlide mit Werkzeugleiste

Die Grundlage des EkgSlide Bildschirms bildet das Elektrokardiogramm. Die Grafik mit dem Millimeterraster lässt sich wie vom iPhone gewohnt skalieren und verschieben. Mit Hilfe der Werkzeugleiste am unteren Bildschirmrand lassen sich die verschiedenen Werkzeuge auswählen.

4.4.2 Aufgabenstellung

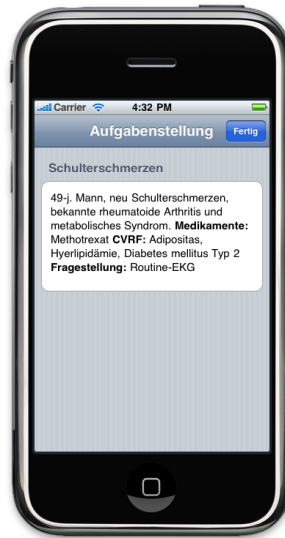


Abbildung 5: Aufgabenstellung

Nach der Auswahl in der Werkzeugleiste wird die Aufgabenstellung als modaler Dialog präsentiert. Sie ist zusammen mit der Befundung und der Auswertung in einheitlichem Stil gehalten.

4.4.3 Befund



Abbildung 6: Eingabemaske für den Befund

In die Felder des Befunds kann direkt hineingeschrieben werden. In den Textfeldern sind mehrzeilige Eingaben möglich. Bei numerischen Feldern werden fehlerhafte Eingaben abgefangen, indem Buchstaben und Sonderzeichen gar nicht erst eingegeben werden können. Wie die Aufgabenstellung lässt sich der modale Dialog über “Fertig” wieder ausblenden.

4.4.4 Auswertung



Abbildung 7: Auswertung des Befunds

Auf dem Auswertungsbildschirm wird der eingegebene Befund mit der Musterlösung verglichen. Textuelle Inhalte können vom Benutzer selbst verglichen werden. Sowohl alle numerische Eingaben als auch der Lagetyp und die Übergangszone werden ausgewertet und farblich als richtig (grün) oder falsch (rot) gekennzeichnet.

4.4.5 Winkelmesser

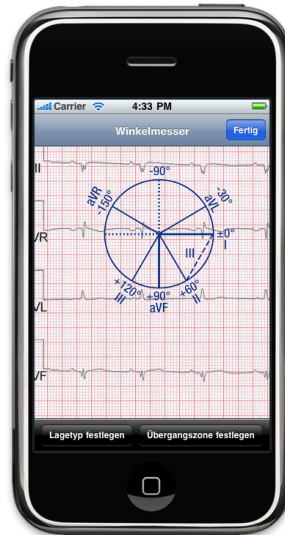


Abbildung 8: Winkelmesser

Der Winkelmesser erscheint nach entsprechender Auswahl in der Werkzeugleiste auf dem Elektrokardiogramm. Durch ziehen lässt er sich auf dem Bildschirm verschieben. Das Elektrokardiogramm lässt sich weiterhin vergrößern und verschieben. Über die Leiste am unteren Bildschirmrand kann der Benutzer seine Erkenntnisse betreffend Lagetyp und Übergangszone über ein komfortables Auswahlfenster direkt in den Befund eintragen.

4.4.6 Vermessungswerkzeuge

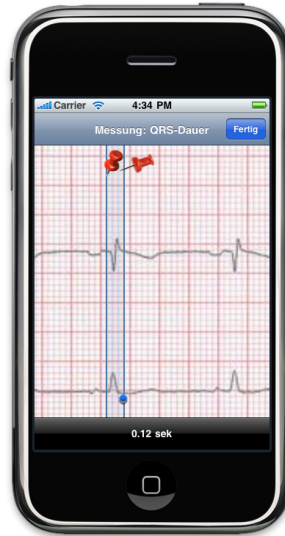


Abbildung 9: Vertikales Vermessungswerkzeug

Durch die Auswahl der zu vermessenden Eigenschaft werden auf dem Elektrokardiogramm zwei Messlinien sichtbar. Das aktuelle Messresultat wird am unteren Bildschirmrand angezeigt und entspricht dem Abstand der Messlinien. Das Elektrokardiogramm kann weiterhin skaliert und verschoben werden. Unabhängig davon können die Messlinien mit dem blauen Griff justiert werden. Über den an jeder Messlinie angebrachten Pin wird die Linie auf dem Elektrokardiogramm fixiert.

Mit der Auswahl von "Fertig" wird der gemessene Wert in den Befund übernommen. Bei erneuter Vermessung derselben Eigenschaft, werden die vorherigen Positionen der Messlinien wiederhergestellt.

4.4.7 Referenzbilder

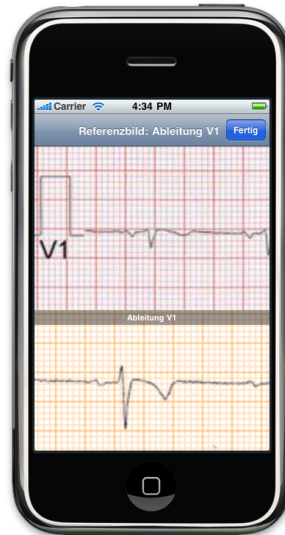


Abbildung 10: Referenzbildansicht

Referenzbilder dienen zum Vergleich eines normalen Herzschlages mit demjenigen des Elektrokardiogramms. Da ein Vergleich nur Sinn macht, wenn die Skalierung die gleiche ist, wirkt sich eine Grössenänderung jeweils auf beide Bereiche aus.

Horizontal können die beiden Ansichten separat gescrollt werden.

4.5 Konzeptionelle Entscheidungen

Die Applikation konnte zum allergrössten Teil gemäss den Paper Prototypes umgesetzt werden.

In einem ersten Grafikkonzept des EkgSlide waren, wie z.B. auf "Paper Prototype 6.3 Winkelmesser" zu sehen, ein vertikaler und ein horizontaler Massstab am linken und oberen Bildschirmrand geplant.

Da es sehr schwierig ist auf dem kleinen Bildschirm von Auge Massstablinien abzuzählen, wurde diese Funktion nicht implementiert bzw. durch die Funktionen "nur Zeit messen" im horizontalen Messwerkzeug und durch das vertikale Messwerkzeug ersetzt.

5 Architektur

Die Architektonische Darstellung soll einen Überblick über die gesamte Implementation geben.

5.1 Ziele

Die Inhalte des IML E-Learning Portal decken sehr viele Themengebiete ab. Die iPhone Applikation kann sämtliche Lerninhalte zum Thema Elektrokardiogramm.

Ziel der Architektur ist es, das Erweitern der Applikation hinsichtlich weiterer Lerninhalte möglichst einfach zu gestalten.

5.2 Übersicht

Folgende Übersicht zeigt das Zusammenspiel der Logischen Komponenten der Applikation. In der Entwicklungsumgebung sind diese in Form von GroupFolders vorhanden.

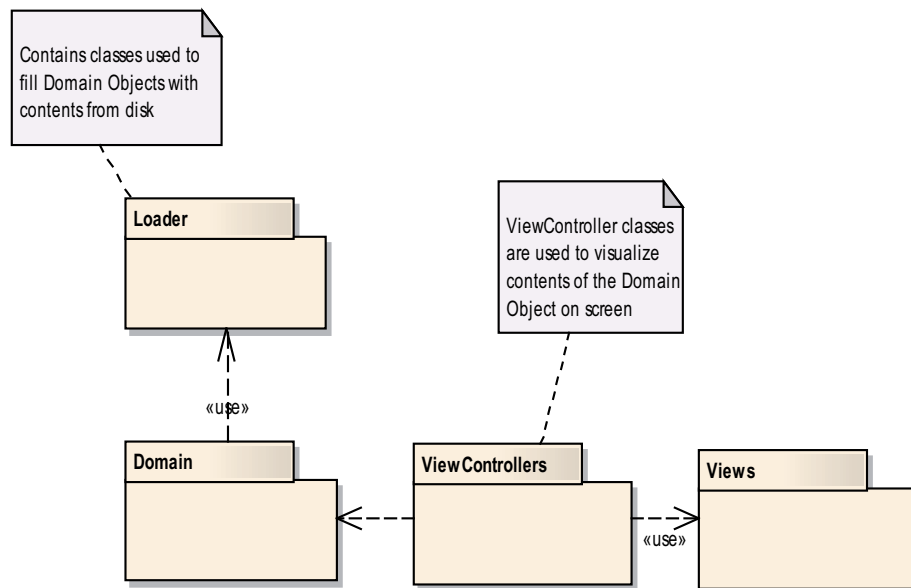


Abbildung 11: Architekturübersicht

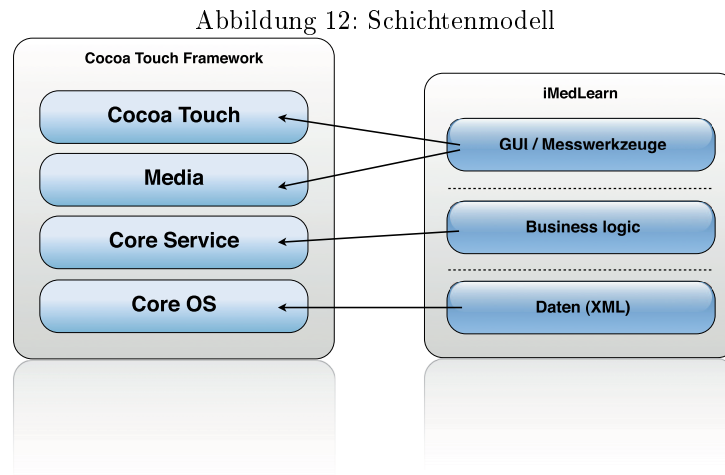
Die Domainobjekte laden ihre Attribute mit Hilfe eines Loaders vom Dateisystem. Für die Darstellung werden die Domainobjekte den ViewControllern bei

der Instanzierung übergeben. Einzelne Unterkomponenten der ViewController sind in Views ausgelagert. Diese können in mehreren ViewControllern wiederverwendet werden.

5.3 Schichtenmodell

Das Eventhandling sämtlicher GUI Komponenten wird vom Cocoa Touch Layer übernommen. Die Grafikverarbeitung und Darstellung wird vom Media Layer geregelt. Die Businesslogik wird mittels Core Service Layer verarbeitet. Hauptsächlich werden dafür die Core Foundation Klassen verwendet. Die Daten werden über Funktionen des Core OS Layers eingelesen.

Diese Schichten des Cocoa Touch Frameworks lassen sich auf unsere Applikation gemäss folgender Grafik abbilden:



5.4 Domain

Bei der Instanzierung der Subklassen von NavigationItem werden nur die Attribute von NavigationItem abgefüllt. Die Subklassenspezifischen Attribute werden beim ersten Zugriff mittels eines Loaders vom Dateisystem gelesen.

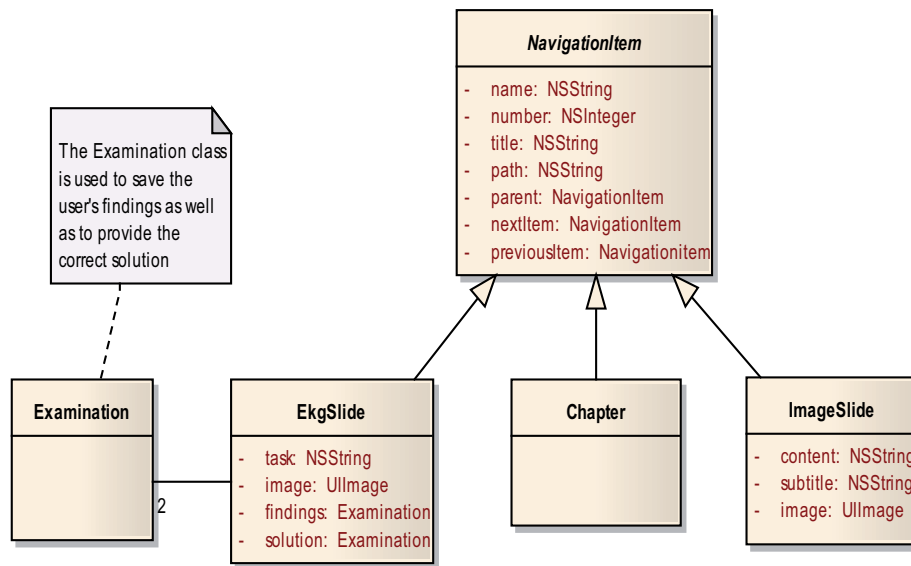


Abbildung 13: Klassenübersicht Domain

5.5 ViewControllers

Die View Controller Klassen implementieren die Visualisierung der Domaindaten auf dem Bildschirm. Das darzustellende Domainobjekt wird dem ViewController jeweils bei der Instanzierung übergeben.

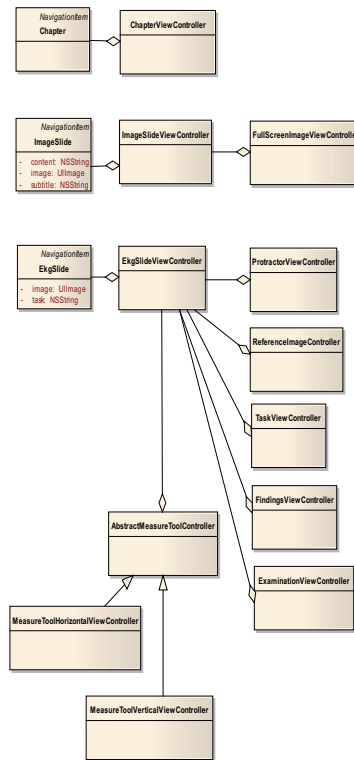


Abbildung 14: Klassenübersicht ViewControllers

ChapterViewController Die Navigation wird mit Hilfe des ChapterView-Controllers auf den Bildschirm gebracht.

ImageViewController Zeigt Slides des Typs ImageSlide an. Dies umfasst die Darstellung eines Bildes sowie Lerninhalte in Form von HTML. Das Bild kann zusätzlich in Vollbild dargestellt werden. Dazu wird der *FullScreenImageViewController* geladen.

EkgSlideViewController Alle interaktiven und zur Darstellung benötigten Komponenten zur Visualisierung eines EkgSlides werden im EkgSlide-ViewController abgehandelt. Die einzelnen Werkzeuge sind in weitere View-Controller ausgelagert.

5.6 Table View Elemente

Die vom iPhone SDK zur Verfügung gestellte UITableView nutzt zur effizienten Speichernutzung ein Pooling Pattern. Bei der Darstellung werden bestehende TableViewCell Objekte wiederverwendet.

Um dieses Konzept optimal zu Nutzen werden zunächst nur Instanzen der Subklassen von TableViewCellContent. Diese repräsentieren die Zellinhalte und unterstützen die von der UITableView verlangte Berechnung der Zellenhöhe.

Die eigentlichen Zellobjekt dann nur bei Bedarf neu instanziiert und wenn möglich wiederverwendet.

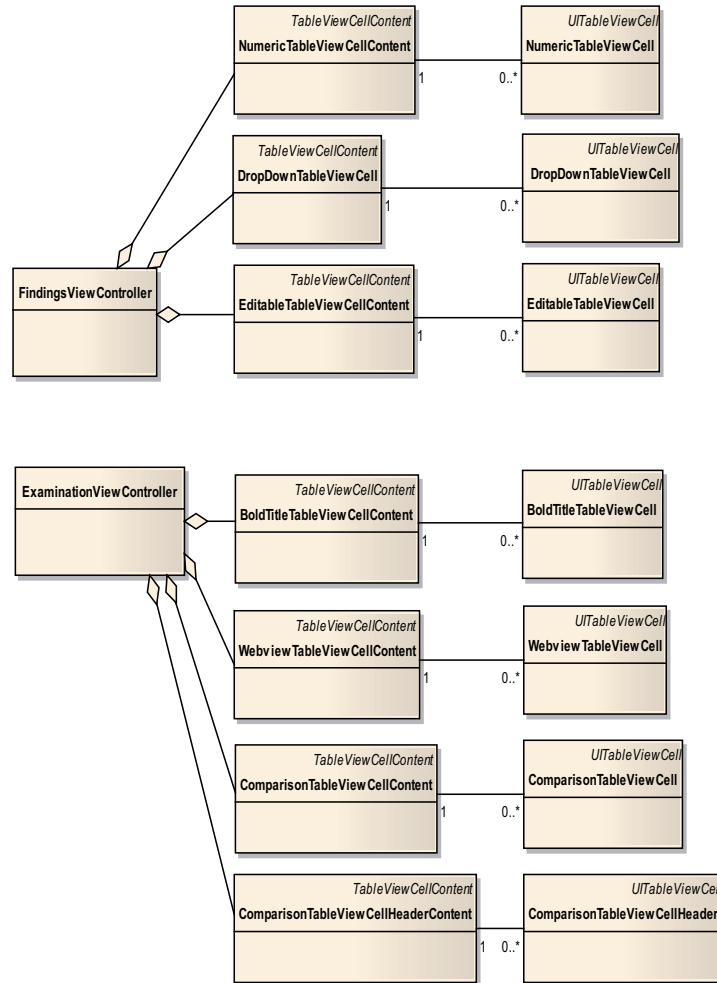


Abbildung 15: Klassenübersicht Tabellenkomponenten

5.7 Strategien und Konzepte

Folgende Software Patterns sind in der iMedLearn Applikation implementiert:

5.7.1 Composite

Um die Menüstruktur generisch zu halten und beliebig verschachteln zu können, wurde diese mittels Composite Pattern umgesetzt.

Es handelt sich hierbei um eine vereinfachte Variante dieses Patterns. So sind Knoten und Blättern vom selben Typ. Ein Knoten wird dann zu einem Blatt, wenn der Knoten keine weiteren Kindknoten mehr hat. In unserem fall sind die Menüeinträge auf der tiefsten Ebene die Blätter.

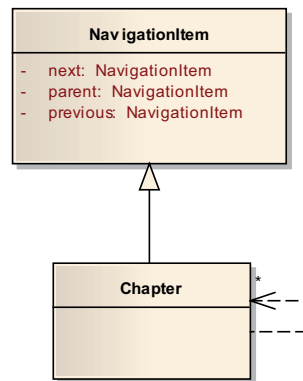


Abbildung 16: Einsatz des Composite Pattern in der Navigation

5.7.2 Lazy Acquisition

Da der Arbeitsspeicher welcher einer iPhone Applikation zur Verfügung steht eher gering ist, muss darauf geachtet werden, dass Ressourcen nicht unnötig Speicher belegen, wenn sie gar nicht verwendet werden. Deshalb wurde für den Zugriff auf die Ressource Properties der Slide Klassen ein lazy loading implementiert welches transparent beim Zugriff erfolgt. Auf der dem folgenden Sequenzdiagramm wird das lazy loading anhand des Ladevorgangs für die ImageSlides visualisiert.

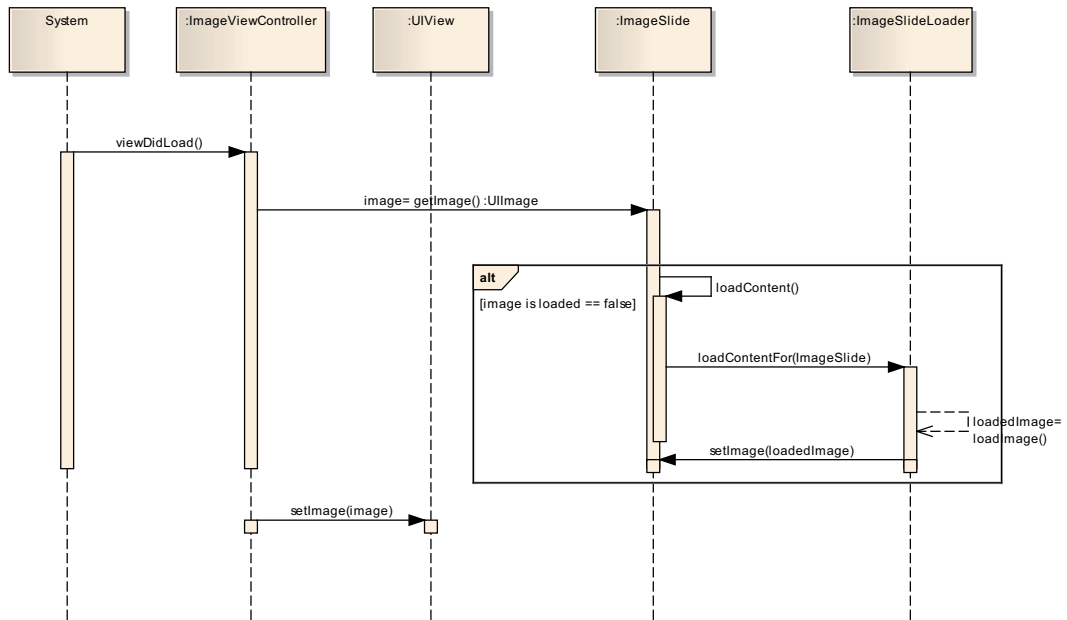


Abbildung 17: Lazy Acquisition im ImageSlide Objekt

5.7.3 Before / After

Es soll sichergestellt werden, dass Ressourcen z.B. beim Verlassen eines Messmodus wieder freigegeben werden. Da aber die meisten Messgeräte erst aus dem Speicher entfernt werden können, wenn deren Ausblend-Animation fertig gelaufen ist, musste ein Callback mechanismus verwendet werden. Dieser funktioniert mittels Protokoll, welches das delegate implementieren muss.

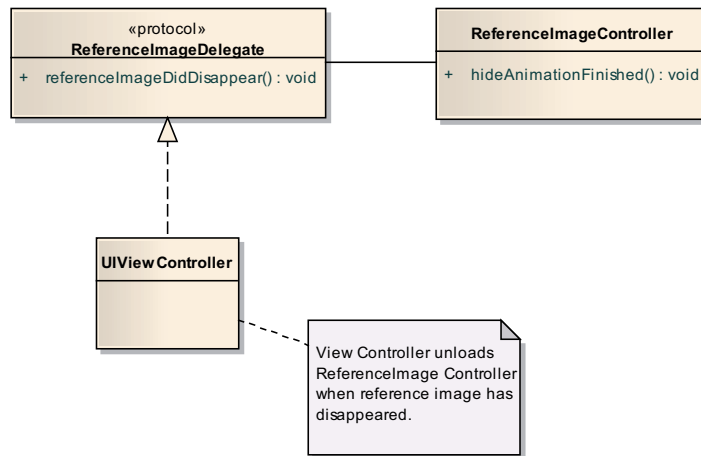


Abbildung 18: Before / After Pattern bei der Ressourcenverwaltung

Auf folgendem Sequenzdiagramm ist zu sehen wie das delegate Objekt (UIViewController) notifiziert wird, nachdem die Ausblend-Animation eines Referenzbildes (siehe domainanalyse.pdf) beendet ist. Sobald das Objekt diese Nachricht erhalten hat, kann es den Referenzbild Controller aus dem Speicher entfernen.

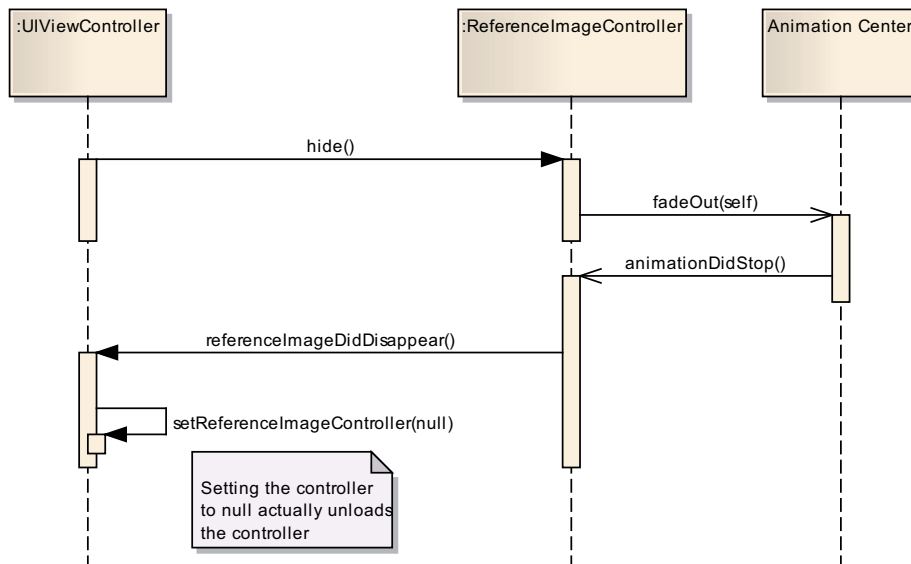


Abbildung 19: Ressourcenfreigabe nach Animationsende

5.7.4 Eager Acquisition

Im Hauptmenü der Applikation werden jeweils die Kapitel mit der Anzahl Seiten / Slides angezeigt welche dieses Kapitel beinhaltet. Um dies schon beim Starten anzeigen zu können werden die Kindknoten dieser Kapitel eager geladen. Der Zugriff erfolgt zwar lazy, jedoch schon beim Startup. Die Ressourcen der Kindknoten hingegen werden erst beim Zugriff nachgeladen (siehe Lazy Acquisition).

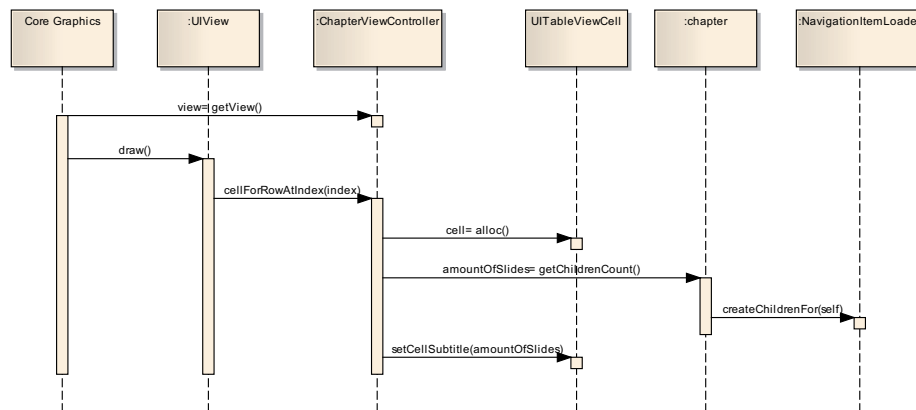


Abbildung 20: Eager Acquisition bei der Navigationsinstanzierung

5.7.5 Factory / Factory method

Um die ViewController für die Slides zu erstellen wird die NavigationViewController Factory beauftragt entsprechend dem übergebenen Slide Objekt ein ViewController zu erstellen. Diese Klasse ist der Information Expert für die NavigationViewController.

Um die Zellen der table views zu erstellen wird jeweils eine Factor Methode damit beauftragt. So bleibt die Erstellung dieser Objekte gekapselt an einem Ort.

5.7.6 Pooling

Horizontale Messwerkzeug gibt es einen Modus um sich den gemessenen Abstand wiederholt anzeigen zu lassen. Beim verändern des Messwertes werden diese Wiederhollinien verschoben und bei bedarf neu erzeugt, da manchmal mehr manchmal weniger Linien angezeigt werden. Um die Verwaltung dieser Grafikressourcen (Linien) effizient zu gestalten, wurde beim zuständigen View

Controller das wurde das Pooling Pattern verwendet. Wenn beim verändern des Messwertes weniger Linien angezeigt werden müssen, werden diese zurück in den Pool gelegt. Wenn beim erneuten verändern des Messwertes wieder mehr Linien angezeigt werden sollen, müssen diese nicht neu alloziert werden, sonder können aus dem Pool genommen werden. Dies bringt einen deutlichen Performance gewinn.

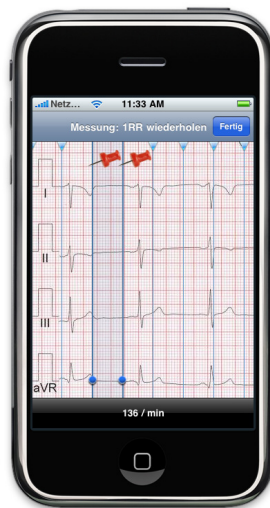


Abbildung 21: Pooling Pattern bei den Wiederholungslinien

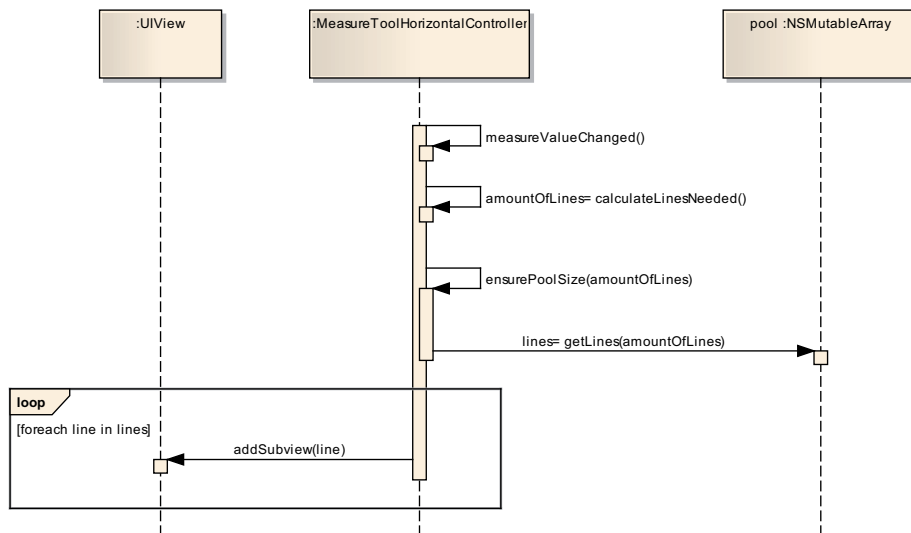


Abbildung 22: Hinzufügen der Linien zur View

5.7.7 MVC

Da die GUI Komponenten des Cocoa Frameworks und der Interface Builder komplett auf MVC aufbauen, wurde die Schnittstelle vom Benutzerinterface zur Businesslogik ebenfalls mit MVC implementiert.



Abbildung 23: Model View Controller

6 Programmierung

6.1 Programmiersprachen & Frameworks

Die iPhone Applikation wurde komplett in Objective-C geschrieben, da Objective-C die native Programmiersprache für das iPhone ist. Die Datenbankschnittstelle (siehe Kapitel Datenexport & Aufbereitung) wurde komplett in PHP geschrieben, da die Webplattform vom IML ebenfalls in PHP geschrieben ist und sie sich so am besten integrieren lässt.

6.2 Kompatibilität

iMedLearn wurde für das iPhone OS in der Version 3.0 entwickelt und getestet. Ältere Versionen werden nicht unterstützt.

6.3 Libraries

Vom Cocoa Touch Framework wird standardmässig nur ein SAX Parser zur Verfügung gestellt. Da eine DOM basierte Lösung aber einen wesentlich einfacheren Zugriff auf XML Knoten ermöglicht, wurde die Kiss XML¹ Library eingesetzt. Diese verwendet intern die libxml2² Library. Erstaunlicherweise scheint diese schneller zu sein als der zur Verfügung gestellt NSXMLParser wie folgende Messresultate zeigen:

¹<http://code.google.com/p/kissxml/>

²<http://de.wikipedia.org/wiki/Libxml2>

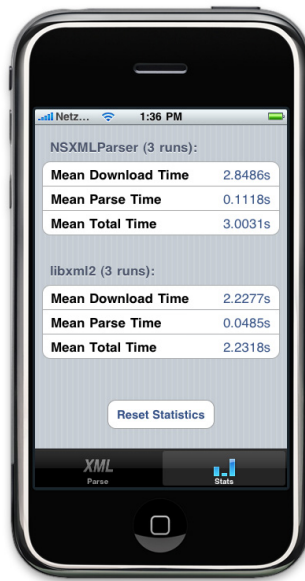


Abbildung 24: Performancevergleich der XML Libraries

6.4 Code style

Da Objective-C eine Programmiersprache ist, welche dynamic Binding verwendet, ein komplexes Memory Management hat und weder private Methoden, noch Konstruktoren/Destruktoren besitzt, basiert vieles auf Conventions. Diese empfiehlt es sich, wenn möglich einzuhalten um mögliche Fehler in der Programmierung zu vermeiden. Wir verwenden folgende «best practices»:

- declare delegates accessors with «assign»
 - Da auf dem iPhone aus Performance Gründen kein Garbage collector zur Verfügung steht, müssen die Referenzzähler selber nachgeführt werden. Wenn die Referenzzähler bei Null angelangt sind werden die Ressourcen automatisch freigegeben. Um nun Abhängigkeitszyklen zu vermeiden sollen die Delegates immer mittels assign deklariert werden, da so der Referenzzähler dieses Delegates nicht erhöht wird und kein Zyklus entstehen kann.
- Use property accessors / Dot syntax
 - Um Member Variablen zu setzen soll nach Möglichkeit ein property accessor³ verwendet werden. Dieser stellt automatisch einen sicheren

³<http://developer.apple.com/mac/library/documentation/cocoa/Conceptual/ObjectiveC/Articles/ocProperties.html>

Mechanismus für die Zuweisung von Member Variablen sicher, sodass die Referenzzähler keine Fehler aufweisen. Bsp.:

```
@property (nonatomic, retain) NSObject* myobj;
```

Ein weiterer Vorteil liegt darin, dass man bei der designated deallocator Methode den Accessor brauchen kann um ein Objekt freizugeben und gleichzeitig den Pointer auf Null zu setzen. Ein Beispielaufruf dafür ist:

```
self.chapter = nil;
```

- Caller retains
 - Wenn eine Factory methode verwendet wird, welche ein neu erstelltes Objekt zurück gibt, soll diese Methode auf dieses Objekt ein «autorelease» aufrufen welches den Referenzzähler für die Dauer des Callerscopes erhöht. Wenn der Aufrufer der Methode das Objekt weiter verwendet muss er ein «retain» darauf aufrufen, oder einen entsprechenden property accessor verwenden.
- Use accessors in init method
 - Die designated initializer Methode «init» ist ein Ort wo meistens Member Variablen zugewiesen werden, welche als Parameter übergeben werden. Gemäss dem oben erwähnten Prinzip «caller retains» müssen die Referenzzähler dieser Parameter mittels retain hochgezählt werden. Damit man das nicht vergisst, empfiehlt es sich die Member Variablen mittels Accessor zu setzen. So werden die Referenzzähler automatisch hochgezählt. (Vorraussetzung dafür ist, dass das Property mit dem Schlüsselwort retain deklariert wurde.) Eine Initializer Methode könnte folgendermassen aussehen:

```
-(id)initWithObject:(NSObject*)myparam{  
    self.mymember = myparam; //accessor retains  
}
```

- dealloc releases all instance variables using accessors with nil value
 - Die dealloc Methode soll alle Membervariablen freigeben. Falls ein Accessor für die Membervariable existiert soll mit dessen Hilfe die Variable auf «nil» gesetzt werden um so diese freizugeben und den Pointer zurückzusetzen. Wenn kein Accessor existiert kann auch «release» verwendet werden.
- when using custom setters, first retain new value, release old value, then assign new to old

- Dieses Verfahren soll verwendet werden, wenn nicht der Default Setter verwendet werden kann, welcher vom Property Mechanismus zur Verfügung gestellt wird. Dieser Ablauf stellt sicher, dass es keinen ungültigen Zugriff gibt, wenn eine Variable zufälligerweise sicher selber zugewiesen wird. Wenn man in letzterem Fall den Referenzzähler der alten Variablen zuerst verringern würde, würde die Ressource freigegeben und somit ein ungültiger Zugriff erfolgen.
- if you create an object with alloc or copy, send it a release or autorelease message at the end of the function. If you create an object any other way, do nothing.
 - Diese Regel gilt nicht bei Initialisierungsmethoden, welche Membervariablen zuweisen. Diese sollten mittels dealloc freigegeben werden.
- call super init at the beginning of init method
 - Diese Regel stellt sicher, dass das Super-Objekt beim Durchlauf der init Methode bereits erstellt und initialisiert ist.
- dont use autorelease if not necessary
 - Da das iPhone ein Mobile Device mit beschränkten Ressourcen ist, sollte auf den Autorelease-Mechanismus (nicht zu verwechseln mit dem Garbage Collector) wenn möglich verzichtet werden, da dieser einen unnötigen Overhead erzeugt. Stattdessen sollte ein manuelles «release» verwendet werden.
- use properties / synthesize even with custom setters and getters
 - Auch wenn eigene Accessor Methoden geschrieben werden, sollte der property Mechanismus verwendet werden, um den Zugriff Einheitlich zu halten.
- don't check for null pointer access
 - Da in Objective-C beim Zugriff auf einen Null Pointer nichts geschieht, besteht auch kein Bedarf diesen Zugriff mit einer Bedingungsabfrage zu sichern.
- when using collections such as NSArray and NSDictionary make sure you release an allocated object after adding it to the collection
 - Da die Cocoa Container Klassen die Referenzzählung selber übernehmen, müssen die Referenzzähler der Objekte nach der Übergabe an solch einen Container verringert werden.

6.5 Debugging

Das Debugging gestaltet sich bei einer auf C basierenden Sprache welche dynamic Binding verwendet nicht ganz einfach. Da auf dem iPhone kein Garbage Collector verwendet werden kann muss das Memory Management manuell programmiert werden. Es kommt öfter vor dass sich dort Fehler einschleichen. Und nicht selten kommt es vor, dass sich die Laufzeitumgebung mit der Meldung «EXEC BAD ACCESS» verabschiedet. Oftmals ist der Fehler im designated deallocator genannt «dealloc» zu suchen. Es ist in diesem Fall sehr nützlich Log Messages zu verwenden, um nicht durch jeden einzelnen verschachtelten dealloc Aufruf durchsteppen zu müssen.

In der Release Version möchte man aber diese Log Messages nicht mehr drin haben, da sie effektiv nur für das Debugging nützlich sind. Deshalb wurde eine eigene Logging Methode namens «DebugLog()» geschrieben, welche beim Release Build durch den Präprozessor entfernt wird. So kann an der Applikation weitergearbeitet werden und die Log Nachrichten weiterverwendet werden, ohne bei jedem Release alle Log Aufrufe zu entfernen.

Beispiel Fehlermeldung bei Programmabsturz:

```
Exception Type: EXC_BAD_ACCESS
Crashed Thread: 0 Dispatch queue: com.apple.main-thread
```

7 Datenexport & Aufbereitung

7.1 Problemstellung

Die bestehende E-Learning Plattform des IML verwaltet die textuellen Lerninhalte in einer MySQL Datenbank. Weitere Inhalte wie z.B. Bilder sind auf dem Dateisystem des Webservers abgespeichert.

Um diese Inhalte auf dem iPhone nutzen zu können, müssen diese in eine für das iPhone gut lesbare Form gebracht werden. Ziel ist es, die Daten aus der Datenbank so einfach wie möglich aufzubereiten und in den aktuellen Entwicklungsstand der iPhone-Applikation zu integrieren.

7.2 Realisierung

Als gemeinsame Basis von iMedLearn und dem bestehenden Inhalt wurde eine XML basierte Dateistruktur entworfen. Diese Schnittstellendefinition umfasst eine Ordnerstruktur sowie diverse XML Schemata. Der eigentliche Datenexport ist mit einem PHP Programm gelöst. Die aufbereiteten Daten können nach dem Export mit einem speziell für diesen Zweck entwickelten Programm namens “Deployer” in die Entwicklungsumgebung XCode integriert werden.

Der Datenexport umfasst folgende Schritte:

1. Aufruf des PHP Programms für den Datenexport. Das Skript wird nun auf dem Server ausgeführt. Ein automatisch angezeigter Report gibt Auskunft

über die Schnittstellenkonformität und iPhone-Kompatibilität des Inhalts. Die aufbereiteten Daten stehen nun als ZIP zum Download bereit.

2. Download der Daten durch den Entwickler.
3. Der Entwickler ruft auf seinem Entwicklungsrechner die “Deployer” Applikation auf, welche das ZIP-File mit den neuen Inhalten in die Entwicklungsumgebung XCode integriert.

In den nachfolgenden Kapiteln wird genauer auf die einzelnen Komponenten dieses Exportprozesses eingegangen.

7.3 Export & Parsing

Die Daten werden mit Hilfe eines PHP Programms aus der Datenbank exportiert und aufbereitet. Da nicht alle Daten strukturiert in der Datenbank abgelegt sind, ist zum Teil zusätzlicher Parsingaufwand entstanden.

7.3.1 Realisierung

Die Implementierung umfasst den Datenexport und eine anschließende Prüfung der Inhalte mitsamt Reportgenerierung.

Eignen sich gewisse Lerninhalte nicht für das iPhone, können sie sowohl auf Kapitel- als auch auf Slideebene mit Hilfe eines Datenbankflags “**iphonecompatible**” für den Export deaktiviert werden. Diesbezüglich wurde im Rahmen dieser Arbeit mit dem IML eine Datenbankerweiterung besprochen. Die detaillierte Erweiterung ist im Dokument “Vorschlag Datenbankerweiterung” beschreiben.

7.3.2 Konfiguration

Bevor der Export starten kann, muss die Umgebung zuerst konfiguriert werden. Dies passiert durch die Anpassung der Datei “**config.php**”. Die einzelnen Konfigurationparameter sind in der Datei selbst beschrieben. Es gilt bei der Konfiguration folgendes zu beachten:

- Das Skript benötigt zum Generieren der Inhalte Schreibrechte im konfigurierten Verzeichnis **output**. Es wird empfohlen, die Rechte für diesen Ordner mit **chmod** auf **777** zu setzen.
- Während dem Export werden sämtliche Bilder aus der Umgebung zusammengezogen. Über den Parameter **\$sourcedir** wird das Hauptverzeichnis der IML Website angegeben. Dabei kann auch ein Symlink zum Hauptverzeichnis verwendet werden.

7.3.3 Skriptaufruf & Ablauf

Der Exportvorgang kann über die Startseite (“index.php”) gestartet werden. Dort ist auch ein Downloadlink für die zuvor aufbereiteten Daten verfügbar.

Das Exportieren umfasst folgende Schritte:

1. Auslesen aller Kapitel des Lernmoduls “Elektrokardiogramm” inklusive der Unterkapitel und der enthaltenen Slides aus der Datenbank.
2. Die Slides werden anhand des Feldes “**typeid**” aus der Tabelle “**slides**” als ImageSlide oder EKGSslide klassifiziert.
 - (a) Handelt es sich um ein EKGSslide, wird das Feld “**description**” analysiert und zur späteren Bearbeitung in einzelne Werte zerlegt. Am besten gelingt dies, wenn das Feld gemäss Musterformatierung (siehe “Musterformatierung”) aufgebaut ist.
3. Sämtliche aus der Datenbank ausgelesene Werte werden nun überprüft und wenn nötig für das iPhone aufbereitet (siehe “Validierung & Automatische Korrekturen”)

7.3.4 Musterformatierung “EKG Befund”

Das Feld **description** sollte nach Möglichkeit immer gleich aufgebaut sein.

Da der eingesetzte Parser nicht alle möglichen Situationen behandeln kann, wird hier eine Musterformatierung vorgeschlagen, welche mit Bestimmtheit erkannt wird:

```
<b>Aufnahmetechnik:</b> Ableitgeschw. 25 mm/s, Eichung 1cm = 1mV, Ableitungen  
korrekt angelegt.  
<b>P:</b> 0.10 sek  
<b>PQ:</b> 0.16 sek  
<b>QRS:</b> 0.09 sek  
<b>QTm:</b> 0.38 sek  
<b>QTc:</b> 0.41 sek  
<b>HF:</b> 70/min  
<b>R-Frequenz:</b> 60/min  
<b>P-Frequenz:</b> 90/min  
<b>Lagetyp:</b> Linkslage  
<b>Übergangszone:</b> V2  
<b>P-Welle:</b> Am Ende der T-Welle des 1. QRS liegt eine P-Welle  
<b>QRS-Komplex:</b> o.B.  
<b>ST-Strecke:</b> o.B.  
<b>T-Welle / U-Welle:</b> o.B.  
<b>Besonderheiten:</b>  
<b>Rhythmus:</b> unregelmässiger SR  
<b>Gesamtbeurteilung:</b> Nc SR, arrhythmisch, AV-Blockierte atriale  
Extrasystole <b>Kommentar: fehlende Überleitung einer atrialen  
Extrasystole</b>
```

7.3.5 Besonderheiten bei der Formatierung

HF oder R-Frequenz & P-Frequenz Im Normalfall sollte nur eine Herzfrequenz (“HF”) angegeben werden. Sind “R-Frequenz” und “P-Frequenz” verschieden, müssen *anstatt* der Herzfrequenz (“HF”) die “R-Frequenz”

und die “P-Frequenz” angegeben werden. Im EKG spezifischen Schema existieren dann nur noch die letzteren beiden Werte. Die “Herzfrequenz” wird dort zur “R-Frequenz”.

Gesamtbeurteilung & Kommentar Da die Kommentare sehr unterschiedlich und teilweise mit Formatierung versehen sind, werden diese ebenfalls unter “Gesamtbeurteilung” zusammengefasst. “Gesamtbeurteilung” ist das einzige Feld, in welchem zusätzliche HTML Formatierung erlaubt ist. Als “Gesamtbeurteilung” wird der Inhalt *vom Stichwort “Gesamtbeurteilung” bis zum Ende des Textes* betrachtet. Im Musterbeispiel entspricht dies dem gesamten Text von “Nc SR...” bis “...Extrasystole”.

7.3.6 Validierung & Automatische Korrekturen

Während des Datenexports werden die Ergebnisse des Parsing Vorgangs auf Konformität mit den jeweiligen XML Schemas geprüft. Kleine Ungenauigkeiten und iPhone-spezifische Details werden automatisch korrigiert. Dies umfasst folgende Punkte:

Hyperlinks werden automatisch aus den Inhalten herausgefiltert.

Dezimalzahlen & Messwerte werden in Gleitkommazahlen umgewandelt. Einheiten wie “pro sek” werden rausgefiltert. Wo ein Komma anstatt des Punktes verwendet wurde, wird dies korrigiert. Als Beispiel: ‘0,08 sek’ wird zu ‘0.08’

Optionsfelder werden falls möglich korrigiert. Da die Werte genau stimmen müssen, sind selbst kleine Abweichungen nicht akzeptabel. Beispielkorrektur: ‘V2-V3’ wird zu ‘V2 / V3’

7.3.7 Reporting

Beim Exportvorgang wird automatisch ein HTML Report generiert. Fehlerhafte und automatisch korrigierte Daten werden dabei farblich hervorgehoben.

Es wird folgende Farbcodierung verwendet:

Valid (grün) Der Inhalt ist für das iPhone geeignet. Die Werte aus der Datenbank entsprechen dem erwarteten Format.

Corrected (gelb) Die Datenbankwerte wurden automatisch korrigiert und für das iPhone kompatibel gemacht. Eine Anpassung der Datenbankwerte ist möglicherweise sinnvoll und sollte geprüft werden.

Invalid (rot) Das Feld ist für die iPhone Applikation in dieser Form nicht verwertbar. Der Inhalt muss in der Datenbank korrigiert werden.

Es empfiehlt sich möglichst alle Inhalte schon in der Datenbank in eine konsistente Form zu bringen.

IML Learning Content Packaging - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/packaging/?nav=generate

IML Learning Content Packaging

EKGSlide Parse & Validation Results

Field	Value
Id	3929
Name	slide-5
Title	Bypassoperation
Type	ekgslide
Original Content	<p>Aufnahmetechnik: Ableitgeschw. 25 mm/s, Eichung 1cm = 1mV, Ableitungen korrekt angelegt, Zitter-Artefakte.</p> <p>P: 0,10 sek</p> <p>PQ: 0,24 sek</p> <p>QRS: 0,10 sek</p> <p>QT: 0,4 sek</p> <p>QTc: 0,42 sek</p> <p>HF: 66/min</p> <p>Lagetyp: LL</p> <p>Übergangszone: V3</p> <p>P-Welle: o.B.</p> <p>QRS-Komplex: SV1 (13mm) + RV5 (22mm) -> 35mm</p> <p>ST-Strecke: leicht deszendierende ST-Senkung von 0,1 mV in V5, V6</p> <p>T-Welle / U-Welle: o.B.</p> <p>Besonderheiten: -</p> <p>Rhythmus: SR</p> <p>Gesamtbeurteilung: Nc SR, LL, AV-Block I. EKG Kriterien für linksventrikuläre Hypertrophie erfüllt.</p>
Task corrected	<p>Hyperlinks removed for iPhone compatibility!</p> <p>72-j. Mann mit St.n. aorto-koronarer Bypassoperation und Aortenklappenersatz vor Jahren. Medikamente: Atenolol, Statin.</p> <p>CvRF: Art. Hypertonie, Hypercholesterinämie, Nikotin 60 py, Diabetes mellitus Typ 2. Fragestellung: Routine-EKG</p>
Technique	Ableitgeschw. 25 mm/s, Eichung 1cm = 1mV, Ableitungen korrekt angelegt, Zitter-Artefakte.
P	'0,10 sek' was converted to '0.10'
PQ	'0,24 sek' was converted to '0.24'

Find: corrected Previous Next Highlight all Match case

Done

Abbildung 25: Auszug aus einem Exportreport

Tip Um Fehler und Warnungen im Report schneller zu lokalisieren, kann im Browser nach den Begriffen “corrected” bzw. “invalid” gesucht werden.

Jedes XML Dokument wird bei der Generierung auf Konformität mit dem jeweiligen XML Schema geprüft. Das Resultat dieser Prüfung ist im Report ebenfalls ersichtlich.

Wichtig: Unabhängig vom Resultat der Validierung werden die Inhalte als ZIP Archiv gepackt und zum Download bereitgestellt. So lassen sich bereits kompatible Inhalte testen, währenddem andere Inhalte noch überarbeitet werden.

7.3.8 Download

Nach erfolgreicher Generierung steht der gesamte Inhalt in Form eines ZIP Archives als Download bereit. Mit Hilfe des “Deployer”s kann dieses Archiv nun in die XCode Umgebung integriert werden.

8 Schnittstellendefinition

Die Schnittstelle gibt eine Ordnerstruktur sowie einige XML Schemata vor. Diese Ordnerstruktur entspricht nach dem Deploymentvorgang auch der Struktur der Daten auf dem iPhone.

8.1 Dateistruktur

Auf oberster Ebene befinden sich die zwei Verzeichnisse “shared” und “data”.

shared beinhaltet für jedes Funktionsmodul ein weiterer Unterordner, in welchem immer wieder benötigte und von der Datenbank unabhängige Daten abgelegt sind. Im Fall des Elektrokardiogramms sind das z.B. die Referenzbilder.

data beinhaltet die eigentlichen Lerninhalte. In jedem Ordner und in jedem Unterordner muss eine XML Datei abgelegt sein, welche die Inhalte des Ordners beschreibt. Hat der Ordner weitere Unterordner (=Unterkapitel), beschreibt diese Datei die Navigationsstruktur und folgt dem generellen Schema (“content.xml”). Ansonsten beschreibt Sie effektiv Lerninhalte und folgt einem spezifischen Schema (“specific.xml”).

8.2 Beispielstruktur

```
+ shared
  + ekg
    - reference-v1.png
    - reference-v2.png
+ data
  - content.xml
  + chapter-1
    - content.xml
    + slide-1
```

```

- specific.xml
- sheme_a.jpg
- image.jpg
+ slide -2
- specific.xml
+chapter -2
etc..

```

8.3 Schemata

Die XML Dokumente mit den vordefinierten Namen “content.xml” und “specific.xml” müssen den im folgenden Absatz definierten Schemata entsprechen.

8.3.1 Generelles Schema (content.xml)

Das generelle Schema wird im ersten gelesenen File (“content.xml” im Ordner “data”) erwartet und beschreibt die Navigation. Zusätzlich wird es immer dann verwendet, wenn in einer übergeordneten Datei das Feld **type** in einem **item** auf den Wert **navigation** gesetzt ist und so auf eine weiter Navigationsebene referenziert wird. Das generelle Navigationsschema sieht wie folgt aus:

Listing 1: content.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/
XMLSchema">
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="items" minOccurs="1" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="item" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="xs:string"/>
                    <xs:element name="title" type="xs:string"/>
                    <xs:element name="type">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:enumeration value="imageslide"/>
                          <xs:enumeration value="ekgslide"/>
                          <xs:enumeration value="navigation"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

name repräsentiert den internen Namen der nächsten Ebene und entspricht zugleich dem relativen Ordernamen, in welchem abhängig vom **type** ein “content.xml” bzw. ein “specific.xml” erwartet wird.

title Der hier angegebene Titel wird auf dem iPhone in der Kapitelnavigation angezeigt. Er darf keine HTML Formatierung enthalten.

type weist auf Typ des Kapitels / Slides hin. Abhängig davon wird auf der nächsten Ebene ein anderer Dateiname für das XML und ein anderes Schema verwendet.

8.3.2 Text / Bild spezifisches Schema (specific.xml)

Wird im generellen Schema der **type** auf **imageslide** gesetzt, wird im entsprechenden Ordner ein "specific.xml" mit dem spezifischen Text/Bild Schema erwartet. Folgendes XML Schema beschreibt dessen Inhalte:

Listing 2: imageslide.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/
  XMLSchema">
  <xs:element name="slide">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="image" type="xs:string"/>
        <xs:element name="subtitle" type="xs:string"/>
        <xs:element name="content" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

image repräsentiert den Dateinamen und Pfad zu einer Bilddatei relativ zum aktuellen Verzeichnis. Das Bild darf eine beliebige Grösse haben und sollte vom Typ GIF, JPEG oder PNG sein.

subtitle Ein Untertitel ohne Formatierung.

content Die eigentlichen Lerninhalte in Form von Text. HTML Formatierung ist hier erlaubt, sollte aber natürlich als CDATA ins XML eingebettet werden.

8.3.3 EKG spezifisches Schema (specific.xml)

Wird im generellen Schema der **type** auf **ekgslide** gesetzt, wird im entsprechenden Ordner ein "specific.xml" mit dem EKG spezifischen Schema erwartet. Folgendes XML Schema beschreibt dessen Inhalte:

Listing 3: ekgslide.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/
  XMLSchema">
  <xs:simpleType name="decimal-or-empty">
    <xs:union memberTypes="xs:decimal empty-string" />
  </xs:simpleType>

  <xs:simpleType name="empty-string">
    <xs:restriction base="xs:string">
      <xs:enumeration value="" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

</xs:restriction>
</xs:simpleType>
<xs:element name="slide">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="image" type="xs:string"/>
      <xs:element name="task" type="xs:string"/>
      <xs:element name="examination">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="technique" type="xs:string"/>
            <xs:element name="p" type="decimal-or-empty"/>
            <xs:element name="pq" type="decimal-or-empty"/>
            <xs:element name="qrs" type="decimal-or-empty"/>
            <xs:element name="qtm" type="decimal-or-empty"/>
            <xs:element name="qtc" type="decimal-or-empty"/>
            <xs:element name="rf" type="decimal-or-empty"/>
            <xs:element name="pf" type="decimal-or-empty"/>
            <xs:element name="type">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value=""/>
                  <xs:enumeration value="ÃĀĒRL"/>
                  <xs:enumeration value="ÃĀĒRL - RL"/>
                  <xs:enumeration value="RL"/>
                  <xs:enumeration value="RL - SL"/>
                  <xs:enumeration value="SL"/>
                  <xs:enumeration value="SL - ML"/>
                  <xs:enumeration value="ML"/>
                  <xs:enumeration value="ML - LL"/>
                  <xs:enumeration value="LL"/>
                  <xs:enumeration value="LL - ÃĀĒLL"/>
                  <xs:enumeration value="ÃĀĒLL"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          <xs:element name="zone">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value=""/>
                <xs:enumeration value="&lt; V1"/>
                <xs:enumeration value="V1"/>
                <xs:enumeration value="V1 / V2"/>
                <xs:enumeration value="V2"/>
                <xs:enumeration value="V2 / V3"/>
                <xs:enumeration value="V3"/>
                <xs:enumeration value="V3 / V4"/>
                <xs:enumeration value="V4"/>
                <xs:enumeration value="V4 / V5"/>
                <xs:enumeration value="V5"/>
                <xs:enumeration value="V5 / V6"/>
                <xs:enumeration value="V6"/>
                <xs:enumeration value="&gt; V6"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="pwave" type="xs:string"/>
          <xs:element name="qrscomplex" type="xs:string"/>
          <xs:element name="st" type="xs:string"/>
          <xs:element name="twave" type="xs:string"/>
          <xs:element name="speciality" type="xs:string"/>
          <xs:element name="rhythm" type="xs:string"/>
          <xs:element name="conclusion" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```
</xs:complexType>  
</xs:element>  
</xs:schema>
```

image repräsentiert den Dateinamen und Pfad zu einer Bilddatei relativ zum aktuellen Verzeichnis. Das Bild darf eine beliebige Grösse haben und sollte vom Typ GIF, JPEG oder PNG sein.

task Die Aufgabenbeschreibung. Hier darf HTML Formatierung verwendet werden.

examination enthält die Musterlösung der Befundung. Ausser dem Feld **conclusion** darf keines der Felder HTML Formatierung enthalten. Neben normalen Textfeldern und numerischen Werten gibt es auch Felder mit einer vorgegebener Auswahl an Möglichkeiten.

9 Qualitätssicherung

9.1 Unit testing

Das Unit Testing unter XCode steckt noch in den Kinderschuhen. Das SenTesting Framework ist seit der XCode Version 3.0 nun fest integriert in die IDE. Dieses Framework ermöglicht Unit Tests in gewohnter Manier zu schreiben inklusive setUp und tearDown Hooks. Es hat jedoch einen entscheidenden Nachteil. Die Unit Tests werden zur Compile Zeit ausgeführt. Das bedeutet, dass diese nicht mit dem Debugger analysiert werden können. Wenn also bei der Durchführung der Unit Tests ein fehlerhafter Speicherzugriff stattfindet, wird das Assert Statement gar nie ausgewertet. Somit ist es dann sehr schwierig zu sagen wo genau der Fehler liegt. Auf folgender Abbildung ist die Auswertung (mittels SenTesting Framework) eines fehlgeschlagenen Tests zu sehen:

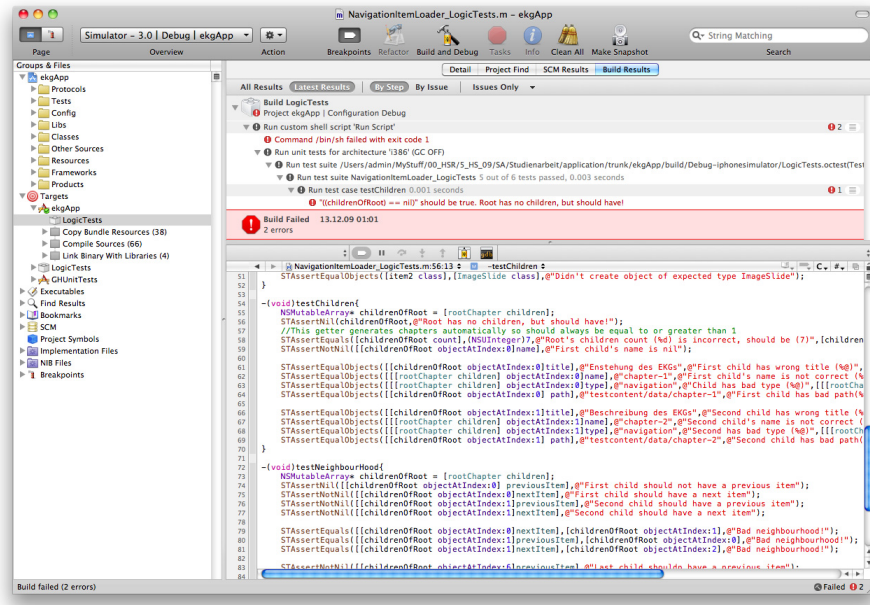


Abbildung 26: Fehlgeschlagener Unit Test in XCode

9.2 GHUnit

Um diesen Mangel der Testumgebung von XCode zu beheben, dass die Unit Tests nicht mit dem Debugger analysiert werden können, wurde ein weiteres Testing Framework von Gabriel Handford namens GHUnit verwendet. Dieses

Framework erlaubt es einem die Unit Tests einzeln anzuwählen und zur Laufzeit ablaufen zu lassen. Somit können die Unit Tests natürlich auch mit dem Debugger durchlaufen werden um die Fehler zu lokalisieren. Auf folgender Abbildung ist ein erfolgreicher Testablauf unter der Verwendung des GHUnit Frameworks zu sehen:



Abbildung 27: Unit Tests auf dem iPhone mit GHUnit

9.3 Testprotokolle

Die durchgeführten Tests sind in den Dokumenten “System Tests” und “Usability Tests” dokumentiert.

9.4 Unit Testing setup

Da das initiale Einrichten der Unit Testumgebung nicht sehr intuitiv ist seien hier nochmals die einzelnen Einrichtungsschritte für das SenTesting erwähnt:

1. Testbundle Target erstellen
2. Unit Testklasse erstellen (Deploy soll nur für das Testbundle erfolgen)
3. Tests automatisieren indem das Testbunde per drag and drop auf das App Target gezogen wird und somit jeweils beim compilieren ausgeführt wird.
4. Als Base SDK für Testbundle «Simulator» einstellen und «Build active architecture only» deaktivieren
5. Falls Libraries verwendet werden (z.B. XML Parser) müssen die Linker Flags ebenfalls angegeben werden
6. Alle Implementations und Ressource Dateien die vom Test verwendet werden, müssen für dieses Target für das Deploying markiert werden

7. Falls in den Tests Dateien eingelesen werden, kann auf diese nicht mittels MainBundle zugegriffen werden. Das Bundle muss mittels bundleForClass abgefragt werden.

Eine Installationsanleitung von GHUnit steht hier zur Verfügung: <http://github.com/gabriel/gh-unit/blob/master/README.md>

9.5 Static Analyzer

Ein weiteres Mittel, welches zur Verfügung steht ist der static Analyzer. Dieser hilft dabei Memory Leaks und weitere Fehler wie z.B. eine Dereferenzierung eines Null Pointers und weitere Fehler im Code schon zur Compilezeit zu erkennen. Er visualisiert sofern möglich den Fehlerpfad zum Root Failure (siehe Fault Tolerance Patterns von Robert S. Hanmer). Auf folgender Abbildung ist solch ein Fehler Pfad welcher vom statischen Analyzer erzeugt wird zu sehen.

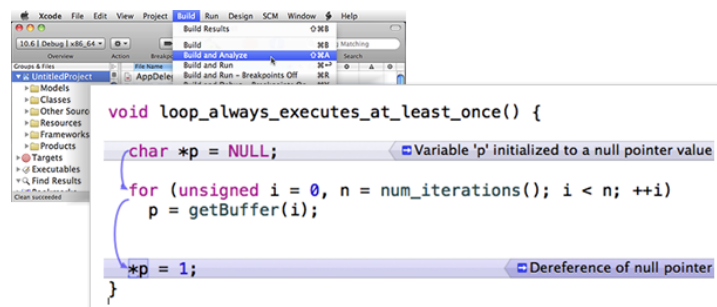


Abbildung 28: Static Analyzing in XCode

9.6 Instruments

Um Memory Leaks zur Laufzeit zu verfolgen steht das Programm «Instruments» von Apple zur Verfügung. Dieses zeichnet alle Memory Leaks auf die gefunden wurden, während man das Programm manuell testet. Das Ergebnis ist allerdings teilweise trügerisch, da auch einzelne Libraries des Cocoa Touch Frameworks noch Leaks drin haben. Es kann also sein dass auf dem Stacktrace eines Leaks eigene Klassen drin sind, obwohl die Ursache des Leaks in den Frameworkklassen liegt. Auf folgender Abbildung ist eine Aufzeichnung eines Leaks zu sehen, dessen Ursprung in der CoreGraphics Library ist:

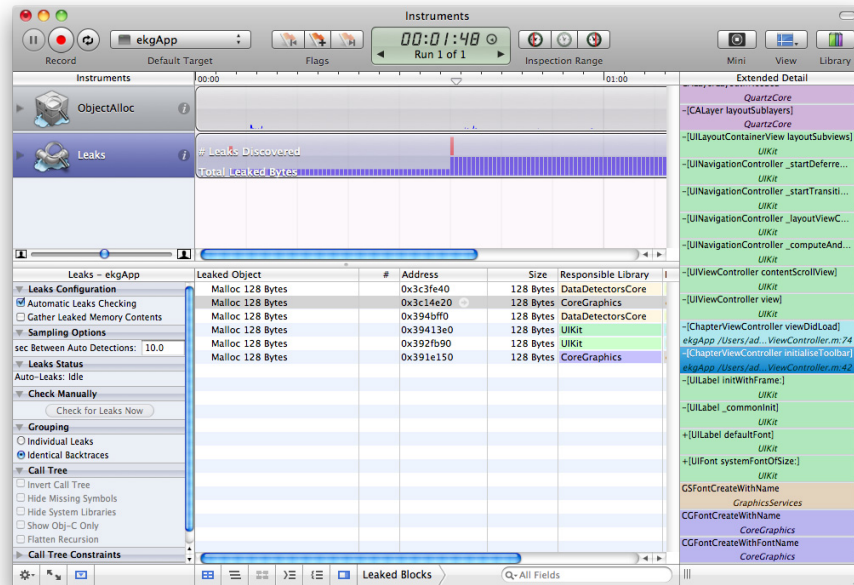


Abbildung 29: Memoryleak Analyse mit Instruments

10 Wartbarkeit

10.1 Ressourcenverwaltung

10.1.1 Deployer



Abbildung 30: Logo von Automator

Da sich der Inhalt der Applikation öfter ändern kann wurde ein Automator⁴ Programm geschrieben welches den Deploymentvorgang der Lerninhalte automatisiert. Das Deployer Diagramm besteht aus Shell Scripts, Apple Scripts und Automator Aktionen.

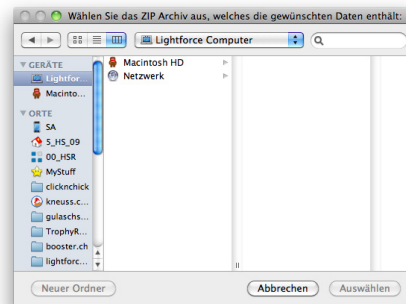


Abbildung 31: Aufforderung für die Angabe des ZIP Archivs

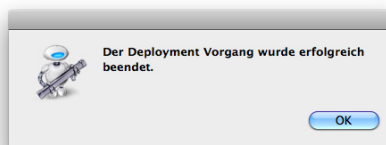


Abbildung 32: Notifikation nach erfolgreichem Deployment

Der Deployer führt folgende Schritte aus:

- Frage nach ZIP Datei fragen welche die neuen Lerninhalte enthält
- Löschen der alten Lerninhalte
- ZIP Datei mit den Lerninhalten entpacken
- Entpackte Dateien an richtigen Ort verschieben
- XCode Projekt Targets löschen (clean)

⁴<http://de.wikipedia.org/wiki/Automator>

- Neuer Build erzeugen

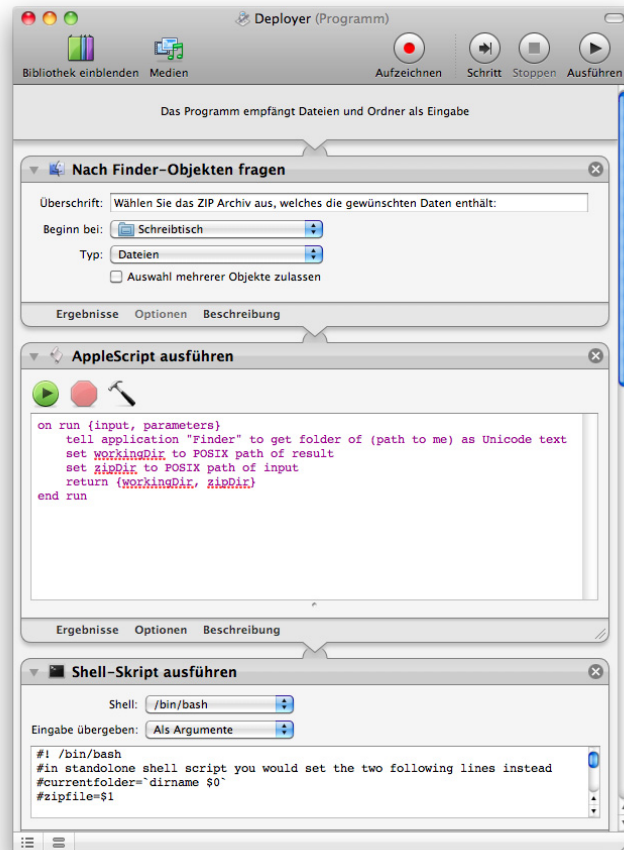


Abbildung 33: Auszug aus dem "Deployer"

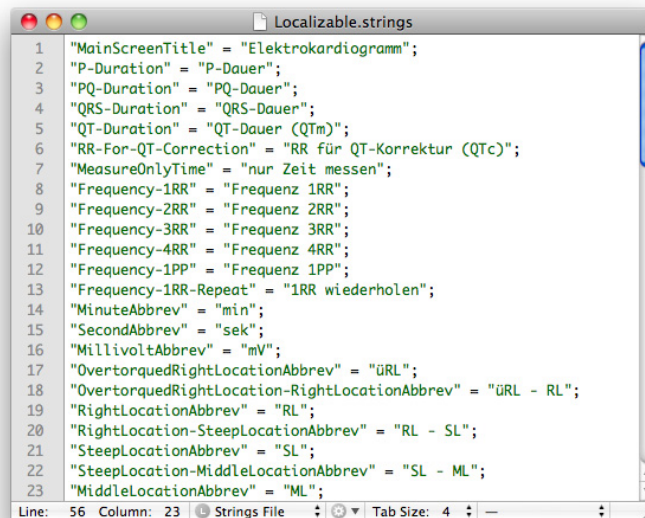
10.1.2 HTML Inhalte

Um Inhalt und Code vollständig zu trennen wurden neben den Lerninhalten und den Bezeichnern (Strings) auch die Impressumsseite extrahiert.

Diese befindet sich im Unterverzeichnis «about» und ist mit einem beliebigen HTML Editor bearbeitbar.

10.1.3 String Ressourcen

Sämtliche Strings welche Sprachbezogen sind wurden in eine sprachspezifische Localizable.strings Datei extrahiert. Die Strings welche sich in GUI Builder Dateien befinden werden lokalisiert indem die gesamten Interface Builder Dateien lokalisiert werden. So wird einerseits eine Internationalisierung ermöglicht sowie die Wartbarkeit der Applikation erhöht. Die Strings bzw. die Bezeichner können so an einer zentralen Stelle bearbeitet werden.



```
1 "MainScreenTitle" = "Elektrokardiogramm";
2 "P-Duration" = "P-Dauer";
3 "PQ-Duration" = "PQ-Dauer";
4 "QRS-Duration" = "QRS-Dauer";
5 "QT-Duration" = "QT-Dauer (QTm)";
6 "RR-For-QT-Correction" = "RR für QT-Korrektur (QTc)";
7 "MeasureOnlyTime" = "nur Zeit messen";
8 "Frequency-1RR" = "Frequenz 1RR";
9 "Frequency-2RR" = "Frequenz 2RR";
10 "Frequency-3RR" = "Frequenz 3RR";
11 "Frequency-4RR" = "Frequenz 4RR";
12 "Frequency-1PP" = "Frequenz 1PP";
13 "Frequency-1RR-Repeat" = "1RR wiederholen";
14 "MinuteAbbrev" = "min";
15 "SecondAbbrev" = "sek";
16 "MillivoltAbbrev" = "mV";
17 "OvertorquedRightLocationAbbrev" = "üRL";
18 "OvertorquedRightLocation-RightLocationAbbrev" = "üRL - RL";
19 "RightLocationAbbrev" = "RL";
20 "RightLocation-SteepLocationAbbrev" = "RL - SL";
21 "SteepLocationAbbrev" = "SL";
22 "SteepLocation-MiddleLocationAbbrev" = "SL - ML";
23 "MiddleLocationAbbrev" = "ML";
```

Line: 56 Column: 23 Strings File Tab Size: 4

Abbildung 34: Internationalisierung über Resourcedateien

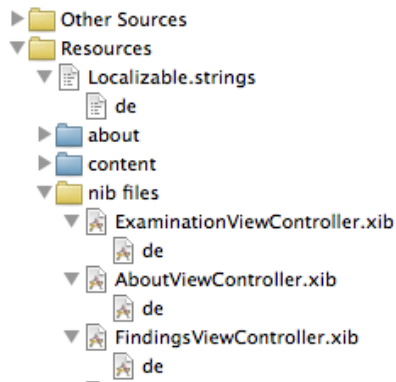


Abbildung 35: Lokalisierte Ordnerstruktur

10.1.4 Konfiguration

Die Datenpfade, Bezeichner und die Namen von gemeinsam genutzten Ressourcen sind in Konfigurationsdateien abgelegt. Auf folgender Abbildung ist ein Auszug solch einer Konfigurationsdatei zu sehen:

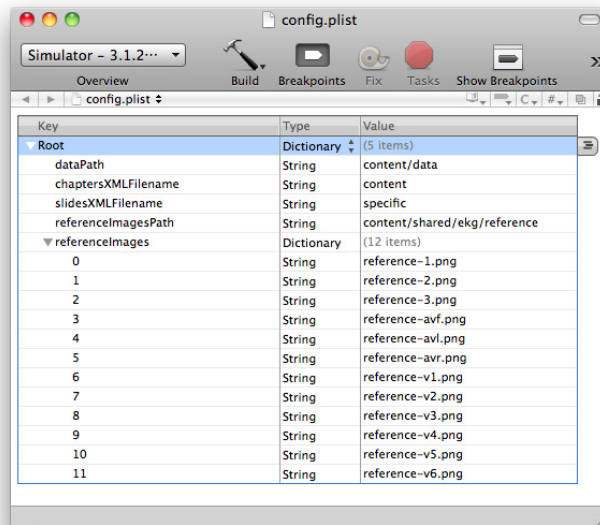


Abbildung 36: Konfiguration statischer Pfade

Der Anzeigename der Applikation und weitere Startup Parameter werden in der Datei «ekgApp-Info.plist» abgelegt, wie auf folgender Abbildung zu sehen ist:

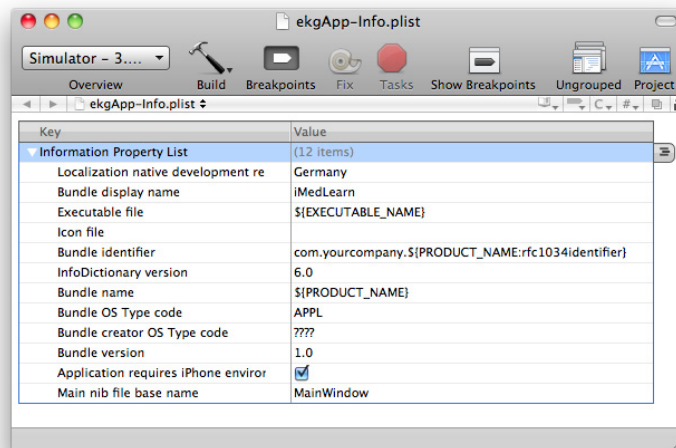


Abbildung 37: Produktkonfiguration

Nomenclature

CR Christian Ramseier

GUI Graphical User Interface:
Eine grafische Benutzeroberfläche ist eine Software-Komponente, die dem Benutzer eines Computers die Interaktion mit der Maschine über grafische Symbole erlaubt.

HTML Die Hypertext Markup Language (HTML, dt. Hypertext-Auszeichnungssprache), oft kurz als Hypertext bezeichnet, ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten.

libxml2 libxml2 ist eine Programmbibliothek zum Parsen von XML-Dokumenten, die von Daniel Veillard für das GNOME-Projekt entwickelt wurde.

SH Sebastian Hunkeler

System Tests

15. Dezember 2009

Getestete Betriebssysteme iPhone OS 3.0, iPhone OS 3.1.2

Getestete Hardware iPhone 3G, iPhone 3GS

1 Referenzen

- [documentation/Design/system-architecture-document.pdf](#)

2 Performance

Getestete Funktionalität	iPhone 3G	iPhone 3GS
Startup	2.8 Sekunden	1.3 Sekunden
Bilder laden	keine Verzögerung erkennbar	keine Verzögerung erkennbar
Webviews laden	Im Schnitt 1.0 Sekunden Verzögerung gemessen	keine Verzögerung erkennbar
TableView scrolling	flüssig mit kleinen Ruckel-Artefakten	flüssig
Messungen	flüssig	sehr flüssig
Animationen	flüssig	sehr flüssig

3 Betriebssystem Kompatibilität

Getestete Funktionalität	iPhone OS 2.0	iPhone OS 3.0	iPhone OS 3.1.2 (aktuellste Version; Stand: 15.12.2009)
Menü	nicht unterstützt gemäss Spezifikation	keine Fehler festgestellt	keine Fehler festgestellt
Image Slides	nicht unterstützt gemäss Spezifikation	keine Fehler festgestellt	keine Fehler festgestellt
EKG Slides	nicht unterstützt gemäss Spezifikation	keine Fehler festgestellt	keine Fehler festgestellt
Datenhaltung (XML)	nicht unterstützt gemäss Spezifikation	keine Fehler festgestellt	keine Fehler festgestellt

4 Stabilität

Getestete Funktionalität	iPhone 3G	iPhone 3GS
Startup	stabil	stabil
Bilder laden	stabil (getestet mit den aktuellen Inhalten d.h. Bilder mit max. 1120 x 600 px)	stabil (getestet mit den aktuellen Inhalten d.h. Bilder mit max. 1120 x 600 px)
Webviews laden	stabil	stabil
TableView scrolling	stabil	stabil
Messungen	sporadischs Fehlverhalten bei der Millivoltmessung Behoben mit der Programmversion 1.0	sporadischs Fehlverhalten bei der Millivoltmessung Behoben mit der Programmversion 1.0
Animationen	stabil	stabil

Usability Test

14. Dezember 2009

Testteilnehmer Amery Chen Li

Ausbildung keine (Kindergarten)

Alter 6 Jahre

Aufsichtsperson S. Hunkeler

1 Auftrag:

Eine EKG Messaufgabe Lösen. Konkret soll sich der Testteilnehmer selber informieren, wie man den QRS Messwert ausmisst und dies anhand der Aufgabe «Aufnahmetechnik 1» aus dem Kapitel «Niveau 3. Jahr» umsetzen.

2 Auszuführende Schritte:

2.1 «Abschnitte des EKG» im Kapitel «Das EKG schreiben» öffnen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Da die Testperson noch nicht lesen kann, wurde ihre gezeigt wo sich dieser Abschnitt befindet

2.2 Bild im Vollbildmodus anzeigen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Der gegebene Hinweis war: zwei mal auf Bild tippen

2.3 Vollbildmodus verlassen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			

2.4 «Aufnahmetechnik 1» QRS Wert ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Der Testperson wurde nochmals kurz erklärt welchen Bereich sie messen soll

2.5 Millivoltwert der Kurve ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Der Testperson wurde nochmals kurz erklärt welchen Bereich sie messen soll.

2.6 Gemessenen QRS Wert mit Lösung verifizieren

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Die Beschriftung des Buttons musste der Testperson vorgelesen werden. Die «Green Bar» wurde von der Testperson auf Anhieb verstanden.

3 Unterschrift

A M E R Y

Usability Test

13. Dezember 2009

Testteilnehmer Judith Hunkeler

Ausbildung Kauffrau

Alter 59 Jahre

Aufsichtsperson S. Hunkeler

1 Auftrag:

Eine EKG Messaufgabe Lösen. Konkret soll sich der Testteilnehmer selber informieren, wie man den QRS Messwert ausmisst und dies anhand der Aufgabe «Aufnahmetechnik 1» aus dem Kapitel «Niveau 3. Jahr» umsetzen.

2 Auszuführende Schritte:

2.1 «Abschnitte des EKG» im Kapitel «Das EKG schreiben» öffnen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			

2.2 Bild im Vollbildmodus anzeigen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Der gegebene Hinweis war: zwei mal auf Bild tippen

2.3 Vollbildmodus verlassen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Der gegebene Hinweis war: einmal mal auf Bild tippen um Navigationsleiste einzublenden

2.4 «Aufnahmetechnik 1» QRS Wert ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
	✓		Der Testperson wurde nochmals kurz erklärt welchen Bereich sie messen soll

2.5 Millivoltwert der Kurve ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Der Testperson wurde nochmals kurz erklärt welchen Bereich sie messen soll

2.6 Gemessenen QRS Wert mit Lösung verifizieren

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Die «Green Bar» wurde von der Testperson auf Anhieb verstanden.

Usability Test

14. Dezember 2009

Testteilnehmer Keely Li

Ausbildung Mediamatikerin

Alter 25 Jahre

Aufsichtsperson S. Hunkeler

1 Auftrag:

Eine EKG Messaufgabe Lösen. Konkret soll sich der Testteilnehmer selber informieren, wie man den QRS Messwert ausmisst und dies anhand der Aufgabe «Aufnahmetechnik 1» aus dem Kapitel «Niveau 3. Jahr» umsetzen.

2 Auszuführende Schritte:

2.1 «Abschnitte des EKG» im Kapitel «Das EKG schreiben» öffnen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Wurde auf Anhieb gefunden

2.2 Bild im Vollbildmodus anzeigen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			(Testperson hat schon geringe Erfahrung im Umgang mit dem iPhone)

2.3 Vollbildmodus verlassen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			

2.4 «Aufnahmetechnik 1» QRS Wert ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			

2.5 Millivoltwert der Kurve ausmessen

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Die Testperson musste kurz ausprobieren welches Werkzeug zu wählen ist.

2.6 Gemessenen QRS Wert mit Lösung verifizieren

Konnte ohne Hilfe umgesetzt werden	Konnte mittels Hinweis der Aufsichtsperson umgesetzt werden	Konnte nicht umgesetzt werden	Bemerkung
✓			Die «Green Bar» wurde von der Testperson auf Anhieb verstanden.

3 Unterschrift



Technischer Bericht Studienarbeit HS 09

18. Dezember 2009

Arbeitstitel iMedLearn - E-Learning Applikation für das iPhone

Studenten Sebastian Hunkeler, Christian Ramseier

1 Aufgabenstellung

Die Aufgabenstellung war so formuliert, dass uns ein grosser Freiraum für eigene Ideen liess. Die Vorgabe war es eine E-Learning Applikation für das iPhone zu erstellen. An den Kickoff Sitzungen wurde mit dem Industriepartner zusammen die Konkrete Idee für die Applikation entwickelt. Es sollte eine E-Learning Applikation sein die sich um das Thema EKG dreht.

2 Vorgehen

2.1 Analysen

Da es schwierig ist umfangreiche Inhalte auf einem kleinen Bildschirm, wie es das iPhone hat darzustellen, war für uns der Paper Prototyp sehr wichtig. Zuvor mussten wir aber einen Testinhalt für die zu entwickelnde Applikation zusammenstellen. Somit mussten wir zuerst die Datenbankstruktur des IMLs analysieren und einen Prototyp der Schnittstelle entwickeln, damit wir überhaupt eine detaillierte Analyse der Datenstruktur tätigen konnten, die wir für die Entwicklung der iPhone Applikation brauchten.

2.2 Architektur

Bei der Architektur galt es zu beachten, dass vom IML zu einem späteren Zeitpunkt möglicherweise weitere bestehende Lerninhalte für das iPhone verfügbar gemacht werden sollen. Darum galt es die Architektur für mögliche Erweiterungen so flexibel wie möglich zu halten.

3 Schlussfolgerung

Mit der entwickelten iPhone Programm iMedLearn steht eine Applikation zur Verfügung die einfach erweiterbar ist und die dank der Schnittstelle einfach mit neuen Inhalten versehen werden kann. Nachdem die IML Mitarbeiter die finalen Inhalte für die Applikation fertiggestellt haben, wird das iMedLearn Programm in den AppStore von Apple gestellt.

Erfahrungsbericht Studienarbeit HS 09

18. Dezember 2009

Arbeitstitel iMedLearn - E-Learning Applikation für das iPhone

Student Christian Ramseier

1 Erfahrungsbericht

Bei der Auswahl der Studienarbeit habe ich mich für ein Entwicklungsprojekt auf einem Embedded Device entschieden, da sich dort bezüglich Usability und Ressourcen ganz andere Problemstellungen ergeben. Mit dem iPhone und Objective C konnte ich diesbezüglich sicherlich sehr gute Erfahrungen mit auf den Weg nehmen.

Die Aufgabenstellung benötigte zudem Fachwissen aus dem medizinischen Bereich, was die theoretisch bereits bekannte Anforderungsanalyse sehr spannend und lehrreich machte. Besonders gut hat mir dabei die konstruktive Zusammenarbeit mit U. Woermann gefallen.

Die Einarbeitungszeit in Objective C war gemessen an der Projektdauer sicherlich sehr aufwendig und zeitintensiv. Dass dabei trotzdem alle Anforderungen an iMedLearn erreicht werden konnten ist sehr zufriedenstellend und die fertige iPhone Applikation macht Freude.

Ich denke die Teamarbeit mit Betreuer, Experte und Industriepartner waren sehr praxisnah und deshalb eine wertvolle Erfahrung für die Zukunft!

Erfahrungsbericht Studienarbeit HS 09

18. Dezember 2009

Arbeitstitel iMedLearn - E-Learning Applikation für das iPhone

Student Sebastian Hunkeler

1 Einleitung

Als erstes möchte ich kurz erwähnen, dass mir die Umsetzung dieses Projekts viel Freude bereitet hat. Wir hatten einen sehr motivierten Industriepartner bzw. Auftraggeber der uns stets wertvolle Inputs geliefert hat. Auch unser Betreuer Hans Rudin war sehr motiviert, hat das Projekt professionell gemanaged und hat sich für uns eingesetzt. Nicht zu vergessen ist unser technischer Betreuer Markus Fretz, welcher uns mit seinem Fachwissen stets zur Seite stand. Vielen Dank nochmals an die erwähnten Herren.

2 Generell

Sehr gut an der Studienarbeit generell finde ich, dass man grösstenteils selbstständig arbeiten darf, seine eigenen Ideen miteinbringen kann und mit einem Industriepartner zusammenarbeitet. Wir haben für das Projekt viel Zeit und oftmals auch Schlaf geopfert, aber ich finde es hat sich gelohnt. Ich bin mit dem Ergebnis sehr zufrieden und habe eine Arbeit abgeliefert auf die ich stolz bin. Alles in allem war es eine interessante Erfahrung und ich konnte einiges lernen was für die bevorstehende Bachelorarbeit sehr nützlich sein wird.

3 Gelerntes

Das wichtigste, was ich gelernt habe ist neben dem generellen Projektmanagement der Umgang mit dem iPhone Framework. Dies war auch einer der grössten Beweggründe diese Arbeit auszuwählen. Ich kenne mich nun sehr gut mit dem iPhone SDK aus und kann dieses Wissen in Zukunft weiterverwenden.

4 Besondere Erfahrungen

Besonders zu erwähnen ist meiner Meinung, dass es zum Teil schwierig ist den praktischen Aspekt dieser Arbeit mit dem theoretischen bzw. schulischen Aspekt zu vereinen. Damit meine ich, wenn während des Projektes neue Wünsche von Seiten des Kunden auftauchen, kann man diese in der Praxis natürlich nicht abschlagen. Da es aber eine schulische Arbeit ist, die zeitlich begrenzt ist, muss man dabei einen Kompromiss eingehen. Ich denke das ist uns sehr gut gelungen. Unser Kunde bzw. der Industriepartner IML, namentlich Ueli Woermann ist mit unserer Arbeit sehr zufrieden. Auch wir sind sehr zufrieden mit unserer abgelieferten Arbeit und sind froh dass wir alle Features die wir gerne implementieren wollten auch in der vorgegebenen Zeit implementieren konnten.

Auswertung Zeiterfassung iPhone Applikation iMedLearn

December 18, 2009

Version: 1.0

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
18.12.2009	1.0	Erstellung	SH,CR

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Auswertung Arbeitswochen	2
3	Auswertung Arbeitspakete	3

2 Auswertung Arbeitswochen

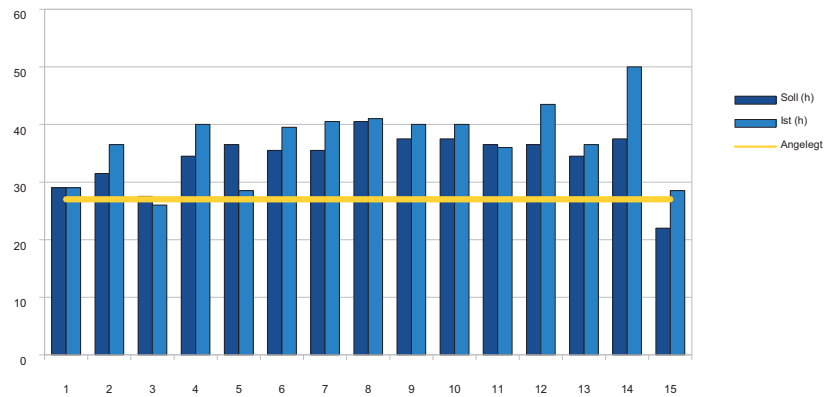


Figure 1: Ist-Soll-Vergleich Zeitaufwand Arbeitswochen

Die aufgewendete Zeit ist leicht über der für die Studienarbeit angelegte Zeit. Ansonsten stimmen sie recht gut mit den Planwerten überein.

3 Auswertung Arbeitspakete

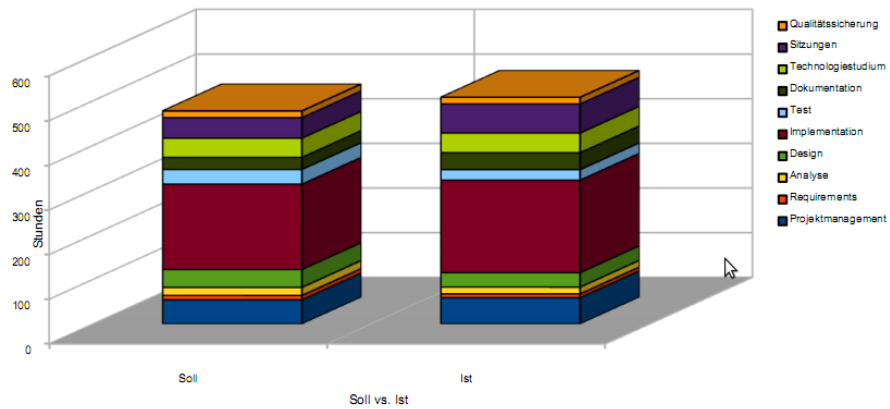
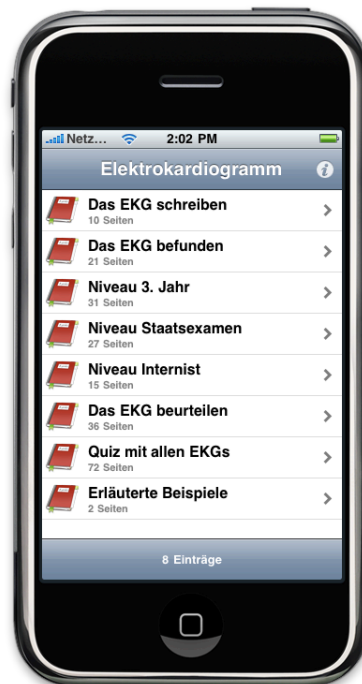


Abbildung 2: Ist-Soll-Vergleich Zeitaufwand Arbeitspakete

Der mit Abstand grösste Punkt bildet in dieser Auswertung die Implementation. Dieser Umstand ist sicherlich auf die Einarbeitungszeit in die Entwicklungsumgebung XCode und die Sprache Objective C zurückzuführen und wurde schon bei der Planung berücksichtigt.

iMedLearn

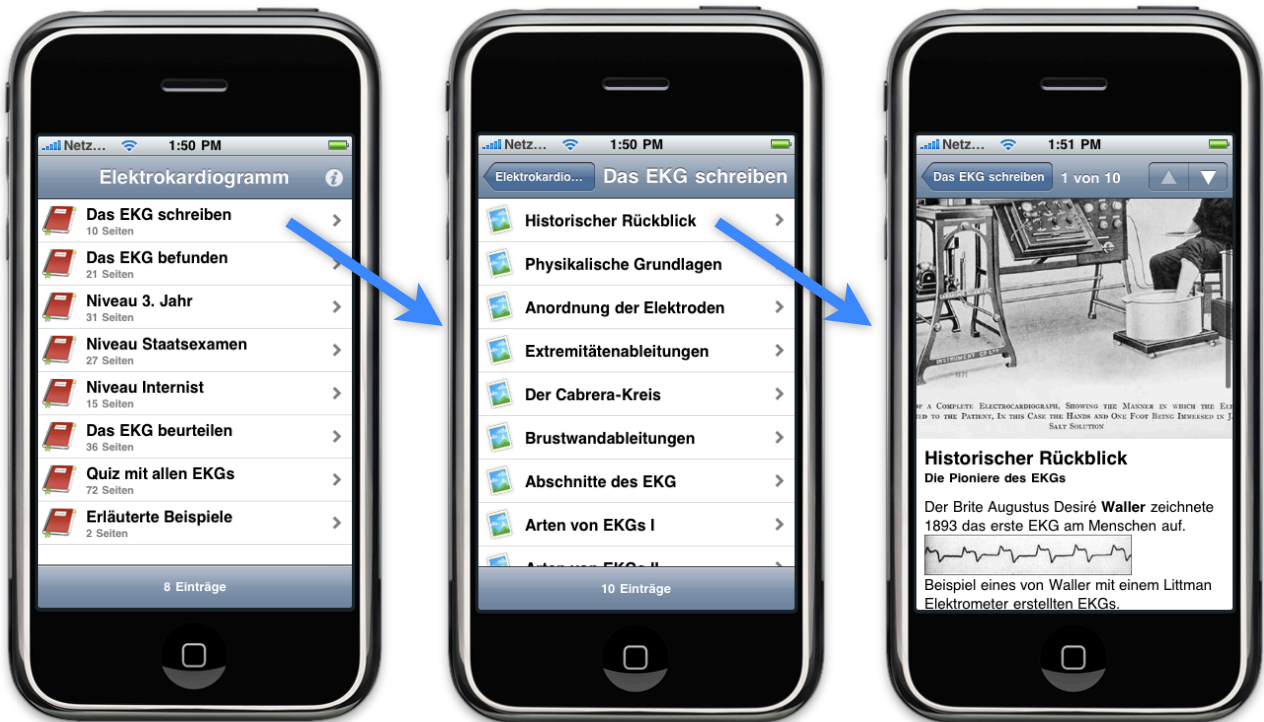
Bedienungsanleitung



INHALTSVERZEICHNIS

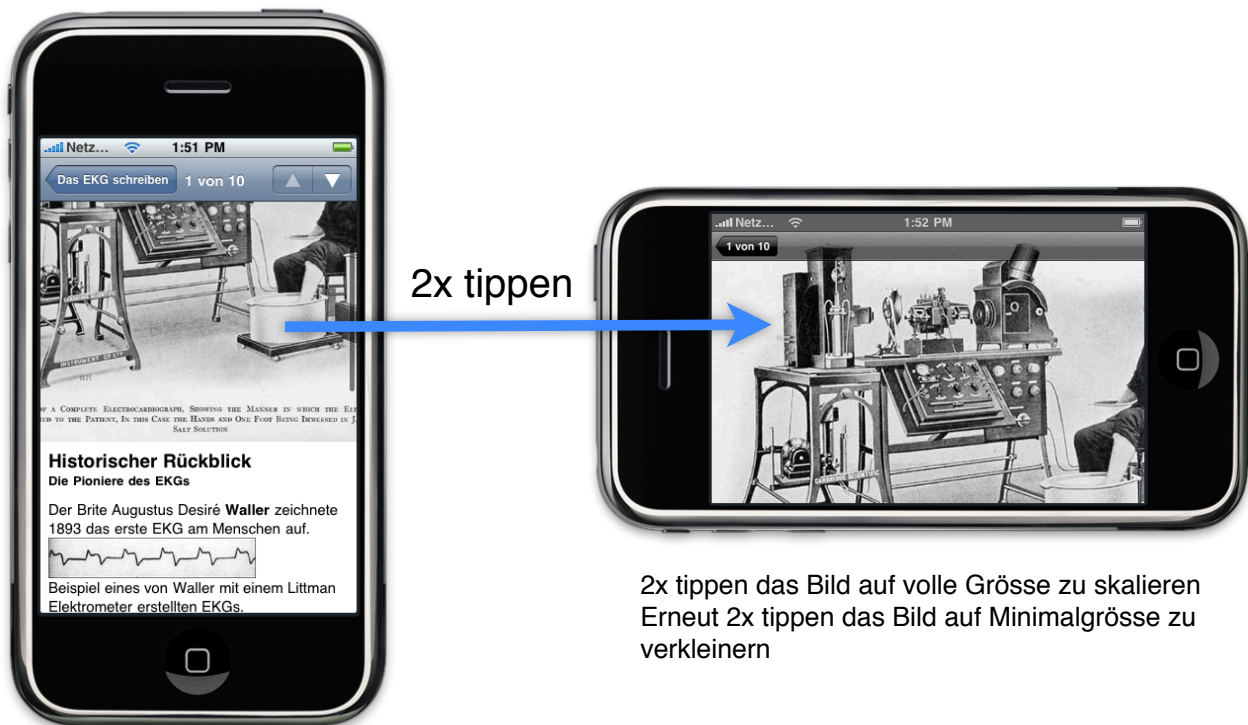
1.	Das Hauptmenü	3
2.	Image Slide	4
3.	EKG Slide	5
4.	Befund	6
5.	Winkelmesser	6
6.	Messwerkzeug	7
7.	Referenzbilder	8

8. DAS HAUPTMENÜ



Im Hauptmenü werden alle Kapitel angezeigt. Durch das Antippen eines Kapitels gelangt man zur nächsten Ansicht. Dort werden die Inhalte des Kapitels angezeigt. Durch das Anwählen eines Eintrags dieses Kapitels gelangt man zu dessen Ansicht. Auf den Abbildungen gezeigt ist der Ablauf, wie man zu einem Image Slide gelangt.

9. IMAGE SLIDE



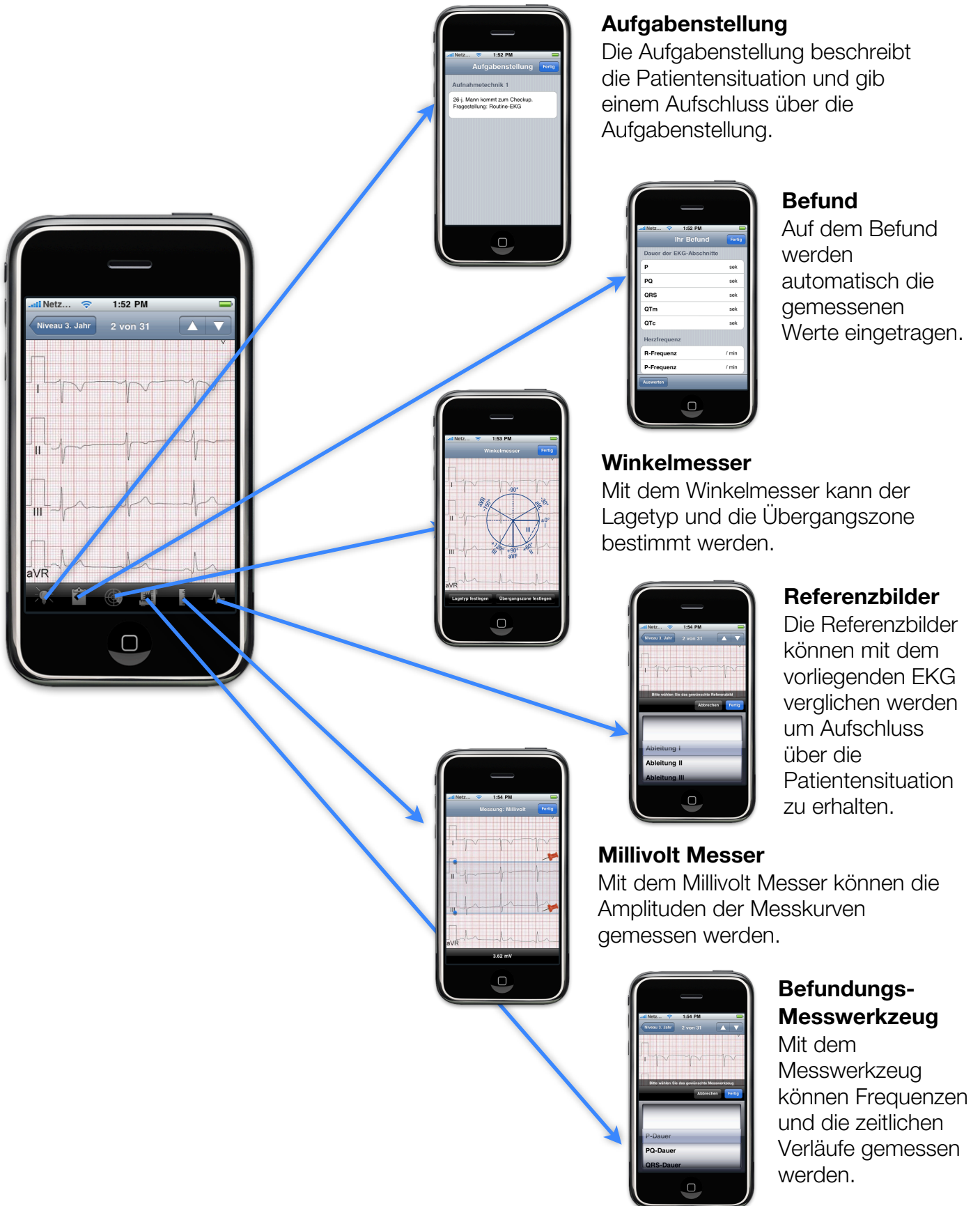
2x tippen das Bild auf volle Grösse zu skalieren
Erneut 2x tippen das Bild auf Minimalgrösse zu verkleinern

Die Image Slides dienen dazu das theoretische Wissen über EKG's und deren Befundung zu vermitteln. Zuerst auf dem Image Slide hat es immer ein Bild. Dieses kann entweder direkt in gewohnter iPhone Manier (mit zwei Fingern) vergrössert werden, oder auf einer separaten Ansicht in voller Grösse begutachtet werden. Um in den Vollbild Modus zu wechseln muss das Bild zweimal angetippt werden.

Im Vollbild Modus kann durch zweimaliges Antippen des Bildes, dieses auf volle Grösse skaliert werden und durch erneutes zweimaliges Antippen wieder verkleinert werden.

10. EKG SLIDE

Hier sehen Sie eine kurze Übersicht der Bereiche des EKG Slides.



11. BEFUND



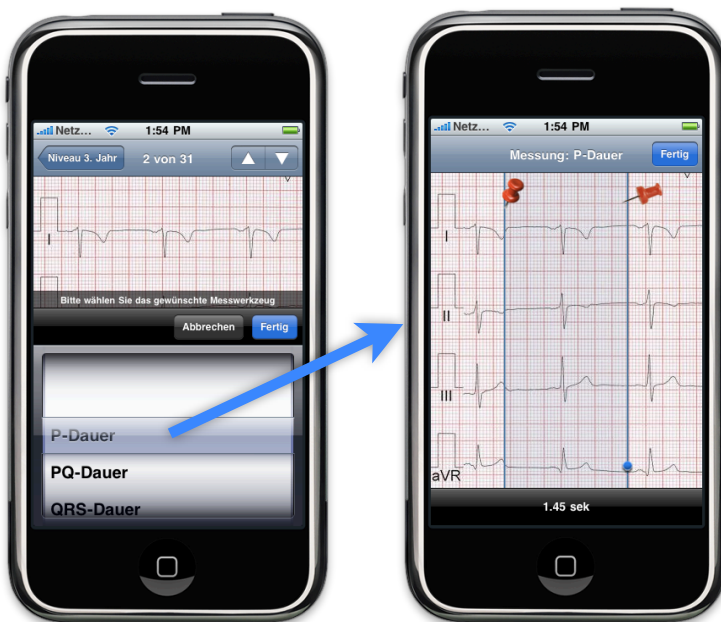
Auf dem Befund sind die bereits vermessenen Messwerte ersichtlich. Diese können entweder manuell mit der virtuellen Tastatur eingetragen werden oder mit den Werkzeugen gemessen werden, worauf die Werte automatisch in den Befund übertragen werden. Beim Antippen des «Auswerten» Buttons werden die eingetragenen Messwerte überprüft und das Ergebnis angezeigt.

12. WINKELMESSER



Der Winkelmesser kann verwendet werden, um den Lagetypp und die Übergangszone zu bestimmen. Im Winkelmesser Modus kann der korrekte Lagetypp und die richtige Übergangszone direkt mittels einer Auswahlkomponente festgelegt werden.

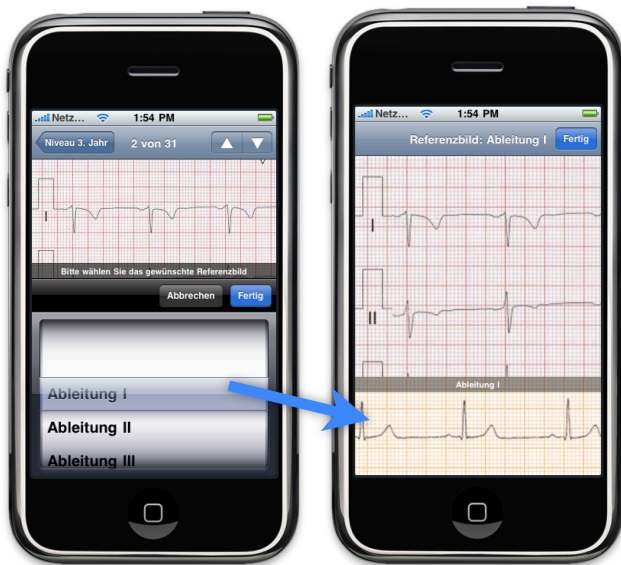
13. MESSWERKZEUG



Mit Hilfe der verschiedenen Messwerkzeuge wie z.B. P-Dauer, PQ-Dauer QRS-Dauer etc. können die entsprechenden Felder im Befund ausgefüllt werden. Mit Hilfe der Auswahlkomponente kann das gewünschte Werkzeug angewählt werden und man gelangt in dessen Messmodus. Beim Antippen von «Fertig» wird der Wert automatisch in den Befund übernommen.

Die Linien kann man entweder mit dem blauen Knopf, oder mittels Scrolling des Hintergrundes verschieben, sofern diese nicht mittels Reissnagel am Hintergrund befestigt wurden. Die Reissnägeln sind unter anderem nützlich um das EKG vergrößern zu können, ohne dass sich die Linien verschieben. Bei vergrößertem Zoom Faktor können die Linien dann noch genauer positioniert werden.

14. REFERENZBILDER



Die Referenzbilder können verwendet werden, um die Patientensituation besser einschätzen zu können. Man kann dazu das Referenzbild für die gewünschte Ableitung mittels Auswahlkomponente auswählen, worauf dieses angezeigt wird. Nun kann man entweder das Referenzbild oder das EKG vergrößern worauf der Zoomfaktor des EKGs / des Referenzbildes angeglichen wird, sodass die Größenverhältnisse stimmen.

Glossar

iPhone Applikation iMedLearn

December 21, 2009

Version: 1.0

Betreuer HSR:

Prof. Hans Rudin, IFS

Michael Graf, Assistent am IFS

Ansprechpartner Auftraggeber:

Dr. med. Ulrich Woermann, Institut für Medizinische Lehre, Uni Bern

Technischer Berater: Markus Fretz, CREALOGIX AG

Autoren: Sebastian Hunkeler, Christian Ramseier

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
16.12.2009	1.0	Erstellung	SH,CR

Inhaltsverzeichnis

1 Änderungsgeschichte	2
2 Begriffe	2

2 Begriffe

SH Sebastian Hunkeler (Projektmitarbeiter)

CR Christian Ramseier (Projektmitarbeiter)

HR Hans Rudin (Betreuer)

MF Markus Fretz (Experte, Crealogix)

UW Ulrich Woermann (Auftraggeber, IML)

MG Michael Graf (Assistent IML)

IML Institut für Medizinische Lehre an der Universität Bern

E-Learning Platform Die am IML bestehende Lernplattform für Medizinstudenten

Backend Umfasst den Webserver und die Datenbank mit den Lerninhalten

Packetierungskomponente Automatisierter Prozess zum Exportieren und Aufbereiten von Lerninhalten aus dem Backend

EKG Abkürzung für Elektrokardiogramm (Herzrhythmusaufzeichnung)

iPhone Mobiltelefon von Apple

Referenzbild Wird bei der Befundung zum Vergleich mit dem einem gesunden Menschen herangezogen

PHP PHP Hypertext Preprocessor - Eine Skriptsprache zur Erstellung von Webapplikation

Objective C Apple's Sprache für die Programmierung von iPhone Applikation

Instruments Tool zur Untersuchung von Programmen (z.B. auf Memory Leaks)

XCode Entwicklungsumgebung zum Erstellen von iPhone und anderen Programmen für Apple Computer

Deployer Komponente zum Einfügen von mit der Paketierungskomponente erstellten Inhalte in die Entwicklungsumgebung XCode