

SOFTWARE ENGINEERING PRAKTIKUM

CAB-TRAIN

Software-Entwicklungspraktikum (SEP)

Sommersemester 2012

P f l i c h t e n h e f t



Auftraggeber

Technische Universität Braunschweig

Peter L. Reichertz Institut für Medizinische Informatik

Prof. Dr. Reinhold Haux

Mühlenpfordstraße 23

38106 Braunschweig

Betreuer: Markus Wagner

Auftragnehmer:

Name	E-Mail-Adresse
Arnaud Chevrollier	arnaud.chevrollier@tu-bs.de
Christoph Harburg	c.harburg@tu-bs.de
Svenja Molitor	svenja_moli@hotmail.de
Bianca Oppermann	BiancaOppermann@gmx.de
Aisha Shnati	aishash777@googlemail.com
Virnadita Sjahan	virnadita@yahoo.de

Braunschweig, 27.04.2012

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
v0.1	22.04.2012	s. Auftragnehmer		Erste Version, Texte in vorgegebenes Layout eingefügt.
v0.2	25.04.2012	s. Auftragnehmer		In erste Version wurden Tabellen eingebunden.
v0.3	27.04.2012	s. Auftragnehmer		In zweite Version wurden Graphiken hinzugefügt.



Inhaltsverzeichnis

1	Einleitung	5
2	Zielbestimmung	6
2.1	Musskriterien	6
2.2	Wunschkriterien	7
2.3	Abgrenzungskriterien	7
3	Produkteinsatz	8
3.1	Anwendungsbereich	8
3.2	Zielgruppen	8
3.3	Betriebsbedingungen	8
4	Produktübersicht	9
5	Produktfunktionen	11
6	Produktdaten	20
6.1	Benutzerdaten	20
6.2	Trainingsdaten	20
6.3	Bewertungsdaten	20
6.4	Sensordaten	21
7	Nichtfunktionale Anforderungen	22
8	Benutzeroberfläche	25
9	Technische Produktumgebung	26
9.1	Software	26
9.2	Hardware	26
9.3	Orgware	26
9.4	Produktschnittstellen	26
10	Glossar	27

Abbildungsverzeichnis

4.1	Use-Case für Software	9
4.2	Use-Case für Framework	10

1 Einleitung

Nach Absprache mit unserem Betreuer des SEPs, Markus Wagner, wurde die Aufgabenstellung vom PLRI, CAB-Train: Framework für computerassistiertes Bewegungstraining, leicht modifiziert.

Der Auftrag wurde dahingehend abgewandelt, dass nun nicht mehr die Entwicklung eines Frameworks im Vordergrund steht, wie zunächst angenommen, sondern die Implementierung eines Beispielprogramms. Das Framework soll jedoch weiterhin als Teil der Musskriterien bestehen. Infolgedessen hat sich der Auftragsnehmer dafür entschieden, das Pflichtenheft sehr allgemein zu halten und nicht auf eine spezifische Übung zu beschränken.

2 Zielbestimmung

Das Ziel dieses Projektes soll es sein, im Rahmen des SEP 2012 eine Testimplementierung für das Ausführen und Auswerten einer medizinischen Übung am Heimcomputer zu entwerfen. Der Patient soll mit Hilfe dieses Produktes nach der Entlassung aus der ambulanten Rehavversorgung oder einer Rehaklinik in der Lage sein, zur weiteren Genesung, ohne die Kontrolle einer ausgebildeten Fachkraft, für ihn zugeschnittene Bewegungsübung auszuführen.

Außerdem soll ein Framework implementiert werden, welches die Möglichkeit bietet, beliebige Geräte zur Erfassung von Bewegungsdaten zu verwenden. Dieses soll möglichst leicht erweiterbar sein.

Die weitere Entwicklung soll, nach Absprache mit dem Kunden, mit Hilfe des Wasserfallmodells erfolgen.

2.1 Musskriterien

Die folgenden Kriterien müssen innerhalb der vorgegebenen Zeit erfüllt sein:

- Erkennen eines Sensors (z.B. konkret das Wii-Balance Board)
- Herstellen einer Verbindung zu einem erkannten Sensor
- Auswählen der Übung
- Erklären der Übung
- Starten einer Übung
- Anzeigen des Übungsablaufs
- Speichern von Daten
- Beenden einer Übung
- Eingabe der Benutzerdaten
- Auswerten der Übung
- Beenden des Programmes

- Framework welches folgende Anforderungen erfüllt
 - Möglichkeit zur Erkennung von verschiedenen Sensoren
 - Möglichkeit der Speicherung von Daten
 - Möglichkeit der generischen Bewertung
 - Möglichkeit verschiedene Übungseinheiten durchzuführen

2.2 Wunschkriterien

Die folgenden Kriterien können die Software erweitern und verbessern:

- Auswählen des zu verwendenden Eingabegerätes
- Anzeigen von erfolgreichem/fehlgeschlagenem Verbindungsaufbau
- Erklären einer Übung als Bild oder Video
- Darstellen der Daten grafisch als Ausdruck oder Versand per E-Mail

2.3 Abgrenzungskriterien

Die folgenden Kriterien sollen in dem Projekt nicht umgesetzt werden.

- Ausführen auf mobilen Endgeräten
- Verwendung für nicht medizinische Bewegungsübungen

3 Produkteinsatz

Das Produkt dient dem Patienten zur kontrollierten Ausführung der Bewegungsübungen ohne ausgebildete Fachkraft im häuslichen Umfeld und damit der Unterstützung des Genesungsvorganges im Rehabereich. Deshalb beläuft sich die Zielgruppe hauptsächlich auf ältere Menschen. Bei Bedarf kann die Software auch für weitere Anwender, wie zum Beispiel Informatiker oder den behandelnden Arzt, zugänglich gemacht werden.

In den nachfolgend aufgeführten Unterpunkten werden der voraussichtliche Anwendungsbereich, die Zielgruppen und die Betriebsbedingungen des Projektes nochmals explizit aufgeführt.

3.1 Anwendungsbereich

Assistierende Gesundheitstechnologien im Bereich der Rehabilitation

3.2 Zielgruppen

Vorrangig ältere Menschen, Patienten des Rehabereichs
Durch das Framework andere Interessengruppen denkbar

3.3 Betriebsbedingungen

Für den Betrieb soll ein Heimcomputer oder ein Notebook mit installierter Java-Software und Bluetooth ausreichen. Dabei ist eine ständige Bluetoothverbindung immer notwendig. Des Weiteren soll das System je nach Bedarf nutzbar sein und benötigt keine ständige Beobachtung durch den Bediener.

4 Produktübersicht

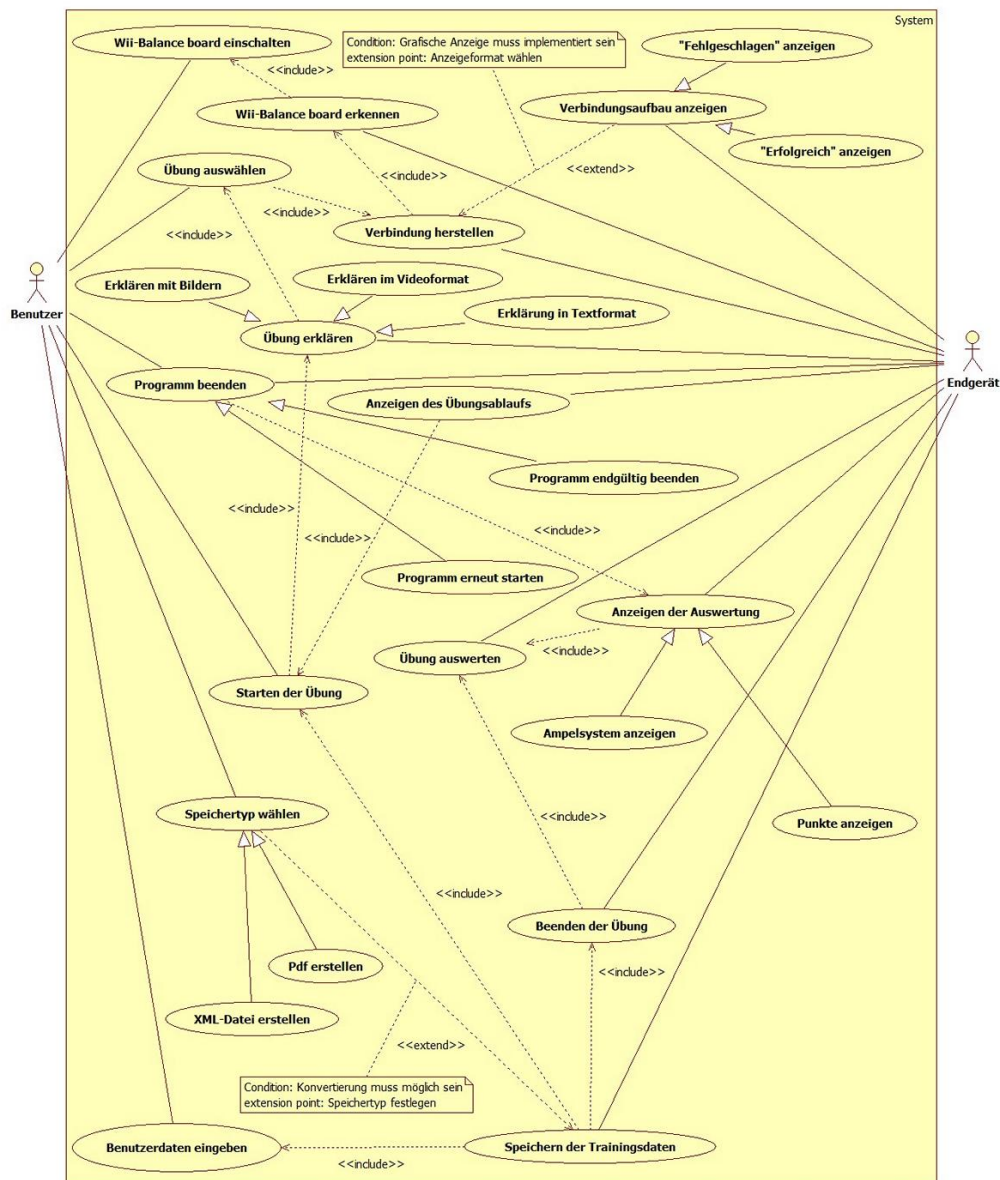


Abbildung 4.1: Use-Case für Software

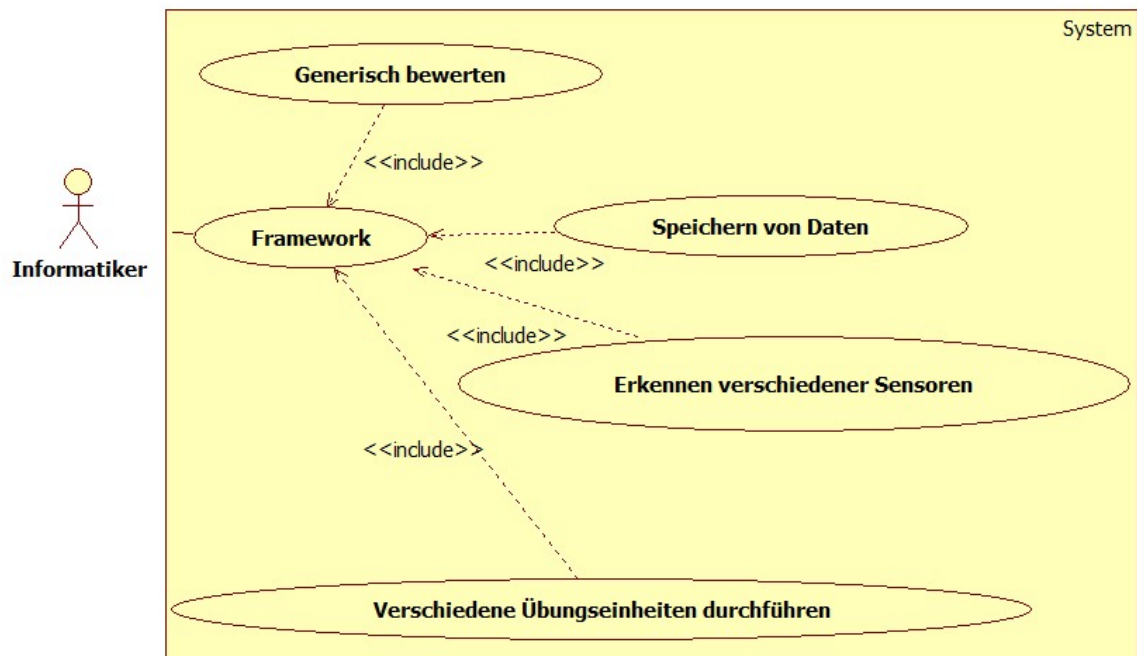


Abbildung 4.2: Use-Case für Framework



5 Produktfunktionen

- /F100/ Erkennen eines Sensors (z.B. konkret das Wii-Balance Board)
 - Geschäftsprozess: Erkennen des Wii-Balance Boards als Sensor
 - Kategorie: Primär
 - Ziel der Funktion: Das Wii-Balance Board wird als Eingabegerät zur Durchführung der Übung genutzt.
 - Vorbedingung: Das Wii-Balance Board muss eingeschaltet und eine Bluetoothverbindung hergestellt sein.
 - Nachbedingung Erfolg: Das Wii-Balance Board wird vom Endgerät erkannt.
 - Nachbedingung Fehlschlag: Das Wii-Balance Board wird vom Endgerät nicht erkannt, wenn keine Bluetoothverbindung besteht. Dies ist zum Beispiel der Fall, wenn das Wii-Balance Board nicht eingeschaltet ist.
 - Akteure: Endgerät und Wii-Balance Board
 - Auslösendes Ereignis: Das Wii-Balance Board wird vom Nutzer eingeschaltet und befindet sich in der richtigen Position für einen Verbindungsaufbau per Bluetooth.
 - Beschreibung: Das Endgerät empfängt ein Bluetoothsignal vom Wii-Balance Board, wenn sich dieses in der Bluetoothumgebung befindet und ein solches Signal versendet.
 - Erweiterung: Weitere Sensoren können erkannt werden, wenn diese eine Verbindung mit dem Endgerät aufbauen können.

- /F200/ Herstellen einer Verbindung
 - Geschäftsprozess: Herstellen einer Verbindung zwischen Endgerät und Wii-Balance Board
 - Kategorie: Primär
 - Ziel der Funktion: Durch das Herstellen einer Verbindung wird die Nutzung des Wii-Balance Boards für das Ausführen der Übung ermöglicht.

- Vorbedingung: Das Wii-Balance Board muss vom Endgerät erkannt worden sein und sich mit einer gültigen MAC-Adresse identifiziert haben.
 - Nachbedingung Erfolg: Es wird das Auswählen einer Übung mit anschließender Durchführung per Wii-Balance Board ermöglicht.
 - Nachbedingung Fehlschlag: Das Auswählen einer Übung ist nicht möglich und somit auch nicht die Ausführung des Trainings, wenn die BluetoothEinstellungen des Endgerätes nicht sachgemäß durchgeführt wurden.
 - Akteur: Endgerät und Sensor
 - Auslösendes Ereignis: Das Wii-Balance Board wurde vom Endgerät erkannt.
 - Beschreibung: Das Endgerät stellt eine Bluetoothschnittstelle für das Wii-Balance Board bereit. Hierfür ist ein Bluetoothgerät am Endgerät notwendig.
-
- /F300/ Auswählen der Übung
 - Geschäftsprozess: Benutzer wählt Übung aus
 - Kategorie: Primär
 - Ziel der Funktion: Dem Benutzer wird es ermöglicht eine geeignete Übung auszuwählen und zu starten.
 - Vorbedingung: Damit eine Übung ausgewählt werden kann, muss zuvor eine Verbindung zwischen dem Endgerät und dem Wii-Balance Board hergestellt worden sein.
 - Nachbedingung Erfolg: Die Erklärung der ausgewählten Übung wird angezeigt.
 - Nachbedingung Fehlschlag: Die Übung kann zwar angeklickt werden, daraufhin folgt aber keine Erklärung. Dies ist auf eine fehlerhafte Implementierung zurückzuführen.
 - Akteur: Benutzer und Endgerät
 - Auslösendes Ereignis: Das Auswahlmenü wurde nach erfolgreichem Verbindungsaufbau angezeigt.
 - Beschreibung: Die vorhandenen Übungen sind in einem extra dafür angelegten Auswahlmenü aufgelistet. Hieraus können die einzelnen Übungen über das Anklicken eines Buttons ausgewählt werden.

- /F400/ Erklären der Übung

- Geschäftsprozess: Vom Benutzer ausgewählte Übung wird erklärt
- Kategorie: Primär
- Ziel der Funktion: Der Benutzer wird in die auszuführende Übung eingewiesen.
- Vorbedingung: Die Übungen werden im Auswahlmenü angezeigt und ein Anklicken der einzelnen Trainingseinheiten wird per Button ermöglicht.
- Nachbedingung Erfolg: Die Erklärung der ausgewählten Übung wird in textueller Form angezeigt.
- Nachbedingung Fehlschlag: Die Erklärung wird nicht angezeigt, aufgrund der Implementierung der Exception bleibt das Auswahlmenü weiterhin bestehen. Außerdem ist es möglich, dass beim Auswählen einer Übung darauffolgend eine falsche Erklärung angezeigt wird. Beides ist auf einen Implementierungsfehler zurückzuführen.
- Akteure: Benutzer
- Auslösendes Ereignis: Der Benutzer hat erfolgreich eine Übung durch Anklicken ausgewählt.
- Beschreibung: Nach dem Anklicken einer Übung wird dem Benutzer in einem sich im Auswahlmenü öffnenden neuen Fenster die Erklärung der Übung angezeigt. Dieses enthält zusätzlich einen Button zum Starten der Trainingseinheit. Beim Anklicken einer anderen Übung wird die vorher angezeigte Erklärung durch die nun passende ersetzt.
- Alternative: Das Auswahlmenü wird geschlossen und dafür ein neues Fenster mit der Erklärung und einem Startbutton geöffnet. Es könnte auch eine in papierbasierter Form vorliegende Erläuterung der Übung vorhanden sein.
- Erweiterung: Die Erklärung wird zusätzlich durch Bilder oder Videos unterstützt.

- /F500/ Starten einer Übung

- Geschäftsprozess: Benutzer startet Übung
- Kategorie: Primär
- Ziel der Funktion: Die ausgewählte Übung wird gestartet und kann somit durchgeführt werden.

- Vorbedingung: Der Startbutton innerhalb des Erklärungsfensters muss vorhanden sein und die Möglichkeit bestehen, diesen anzuklicken.
 - Nachbedingung Erfolg: Die ausgewählte Übung kann gestartet und daraufhin durchgeführt werden.
 - Nachbedingung Fehlschlag: Die ausgewählte Übung kann nicht durchgeführt werden, da beim Anklicken des Buttons das Auswahlmenü bestehen bleibt und das Training folglich nicht gestartet wird. Dies ist auf eine fehlerhafte Implementierung zurückzuführen.
 - Akteure: Benutzer
 - Auslösendes Ereignis: Der Benutzer hat den Startbutton betätigt.
 - Beschreibung: Der Benutzer betätigt den Startbutton. Daraufhin wird das Auswahlmenü geschlossen und der Übungsbildschirm angezeigt.
 - Alternative: Nach dem Auswählen einer Übung wird der Erklärungstext eine gewisse Zeit angezeigt und das Training daraufhin automatisch gestartet.
-
- /F600/ Anzeigen des Übungsablaufs
 - Geschäftsprozess: Der aktuelle Trainingsstand und die auszuführende Teilaufgabe werden angezeigt
 - Kategorie: Primär
 - Ziel der Funktion: Der Benutzer kann sein aktuelles Training verfolgen und seine durchzuführende Aufgabe visuell erfassen.
 - Vorbedingung: Die Datenübertragung zwischen Wii-Balance Board und Endgerät funktioniert einwandfrei.
 - Nachbedingung Erfolg: Visuelle Anzeige des Übungsverlaufs durch Wahrnehmung der Bewegungsänderung.
 - Nachbedingung Fehlschlag: Der Übungsablauf wird nicht angezeigt, obwohl der Startbutton betätigt wurde und sich das Auswahlmenü geschlossen hat. Dies ist auf eine fehlerhafte Implementierung zurückzuführen.
 - Akteure: Benutzer
 - Auslösendes Ereignis: Die Wahl einer Übung und das Betätigen des Startbuttons durch den Benutzer.

- Beschreibung: Nach Betätigen des Startbuttons wird das Auswahlmenü geschlossen und neue Fenster geöffnet, welche den Übungsverlauf und Fortschritt anzeigen. Nach Beendigung der Trainingszeit wird das Fenster geschlossen und das Training beendet.
 - Alternative: Das Auswahlmenü bleibt geöffnet, aber ist nicht mehr anwählbar, bis die Übung beendet ist.
-
- /F700/ Speichern von Daten
 - Geschäftsprozess: Trainingsdaten werden gespeichert
 - Kategorie: Primär
 - Ziel der Funktion: Die Trainingsdaten werden lokal auf dem Endgerät gespeichert und für spätere Aufrufe bereitgestellt.
 - Vorbedingung: Das lokale Speichern der Daten wird durch eine geeignete Implementierung ermöglicht.
 - Nachbedingung Erfolg: Die Daten sind auf dem Endgerät vorhanden und abrufbar.
 - Nachbedingung Fehlschlag: Die Daten konnten nicht gespeichert werden und stehen nicht zur Verfügung. Wenn der Benutzer das Ausführen der Übung für länger als 2 Minuten unterbricht, wird das Training durch vollständiges Schließen des Programmes beendet und es findet keine Datenspeicherung statt.
 - Akteure: Benutzer, Sensor
 - Auslösendes Ereignis: Die Übung wurde gestartet und der Sensor sendet auswertbare Daten an das Endgerät. Die endgültige Speicherung erfolgt nur bei korrekter Ausführung des Programmes.
 - Beschreibung: Das Endgerät empfängt nach gestartetem Training uncodierte 24 Byte Daten vom Wii-Balance Board. Diese werden lokal auf dem Endgerät zur weiteren Benutzung in einem vom Benutzer festgelegten Verzeichnis gesichert. Dies geschieht lediglich, wenn die Übung vollständig durchgeführt, nicht vorzeitig abgebrochen und die Benutzerdaten eingegeben wurden.
 - Alternative: Die Daten werden in einer dafür angelegten Datenbank gespeichert und sind online für in diesem Punkt autorisierte Personen abrufbar.
 - Erweiterung: Die gespeicherten Daten werden per E-Mail an den zuständigen Arzt beziehungsweise Therapeuten geschickt. Weiterhin können sie in verschiedenen Datentypen, beispielsweise als XML-Dateien, gespeichert und zum Ausdrucken bereitgestellt werden.

- /F800/ Beenden einer Übung
 - Geschäftsprozess: Die ausgeführte Übung wird regulär beendet
 - Kategorie: Primär
 - Ziel der Funktion: Eine sachgemäße Speicherung der Daten nach Abschluss des Trainings.
 - Vorbedingung: Der dazugehörige Übungsablauf wurde auf dem Bildschirm ausgegeben und in einer vorgeschriebenen Zeit ausgeführt.
 - Nachbedingung Erfolg: Die temporären Trainingsdaten werden auf der Festplatte des Endgerätes gespeichert.
 - Nachbedingung Fehlschlag: Die temporären Trainingsdaten werden gelöscht, weil das Programm nicht ordnungsgemäß beendet wurde.
 - Akteure: Benutzer
 - Auslösendes Ereignis: Starten und vollständiges Durchführen einer Übung.
 - Beschreibung: Die ausgewählte Übung wurde in der vorgegebenen Zeit ordnungsgemäß durchgeführt und damit die Trainingseinheit beendet. Daraufhin öffnet sich ein neues Fenster zur Eingabe der Benutzerdaten.
 - Alternative: Die Trainingseinheit wird nach einer festgelegten Anzahl an Wiederholungen beendet.
 - Erweiterung: Durch einen zusätzlichen Button kann das Training auf Wunsch vorzeitig beendet werden.

- /F900/ Eingabe der Benutzerdaten
 - Geschäftsprozess: Der Benutzer gibt seine Personendaten ein
 - Kategorie: Primär
 - Ziel der Funktion: Die Benutzerdaten werden gespeichert, damit die Trainingsdaten einer Person zugeordnet werden können.
 - Vorbedingung: Nach dem Ablauf des Trainings muss ein neues Fenster geöffnet werden, in dem der Benutzer seine personenbezogenen Daten eintragen kann.
 - Nachbedingung Erfolg: Die Personendaten können eingetragen und zusammen mit den Trainingsdaten gespeichert werden.

- Nachbedingung Fehlschlag: Die Personendaten können nicht eingetragen werden, da kein entsprechendes Eingabefenster auf der Benutzeroberfläche erscheint. Aufgrund dessen können weder die Benutzerdaten noch die Trainingsdaten gespeichert werden. Dies ist auf einen Implementierungsfehler zurückzuführen.
 - Akteure: Benutzer
 - Auslösendes Ereignis: Die auszuführende Übung wurde nach Ablauf der vorgegebenen Zeit beendet.
 - Beschreibung: für den Benutzer öffnet sich nach dem Beenden der Übung ein weiteres Fenster in dem er aufgefordert wird, seine personenbezogenen Daten einzugeben. Die Datenspeicherung und die weitere Ausführung des Programms erfolgt erst, nachdem der Benutzer seine Eingabe mit einem Button bestätigt hat.
 - Erweiterung: Wenn der Benutzer keine Daten eingibt, können trotzdem die Trainingsdaten mit dem Zusatz „Anonym“ gespeichert werden.
-
- /F1000/ Auswerten der Übung
 - Geschäftsprozess: Werte zur Kontrolle des Trainingsverlaufs werden ausgegeben
 - Kategorie: Primär
 - Ziel der Funktion: für den Benutzer soll eine visualisierte Bewertung, zur eigenständigen Überprüfung des Trainingserfolgs, auf der Benutzeroberfläche angezeigt werden.
 - Vorbedingung: Mit Hilfe der gespeicherten Trainingsdaten, die abrufbar sind, kann eine geeignete Auswertung berechnet werden.
 - Nachbedingung Erfolg: Die Auswertung der Trainingseinheit wird durch ein Punktesystem, und zum leichteren Verständnis, zusätzlich als Ampelsystem dargestellt.
 - Nachbedingung Fehlschlag: Die Auswertung der Trainingseinheit wird nicht dargestellt, da ein Abruf der notwendigen Daten zur Berechnung der Bewertungssysteme nicht möglich ist. Dies könnte ein Implementierungsfehler sein, da der Pfad in dem sich die gespeicherten Daten befinden nicht gefunden werden kann und die Daten somit nicht verwendbar sind.
 - Akteure: Endgerät
 - Auslösendes Ereignis: Der Benutzer hat seine Personendaten eingetragen und seine Eingabe mit einem Button bestätigt.

- Beschreibung: Nachdem die Benutzer- und Trainingsdaten erfolgreich gespeichert und verwendbar sind, können diese in ein Punktesystem eingetragen und anschließend ein Mittelwert berechnet werden. Dieser wird verwendet, um eine geeignete Beurteilung für das Ampelsystem, welches dem Benutzer angezeigt wird, zu erhalten.
 - Alternative: Die Auswertung wird nicht durch ein Ampelsystem, sondern andere Graphiken, dargestellt.
 - Erweiterung: Eine weitere Ausgabe auf der Benutzeroberfläche, die den Wochenverlauf des Trainings in einem Graphen anzeigt.
-
- /F1100/ Beenden des Programmes
 - Geschäftsprozess: Endgültiges Beenden oder erneutes Starten des Programmes
 - Kategorie: Primär
 - Ziel der Funktion: Dem Benutzer soll es ermöglicht werden, nach der Auswertung seines zuvor ausgeführten Trainings, das Programm entweder vollständig zu beenden oder es von neuem zu starten und damit weitere Übungen ausführen zu können.
 - Vorbedingung: Das Bewertungsfenster wurde angezeigt und innerhalb des Fensters werden Buttons bereitgestellt.
 - Nachbedingung Erfolg: Nach dem Betätigen eines Buttons wird das Programm entweder endgültig beendet oder abermals gestartet.
 - Nachbedingung Fehlschlag: Nach dem Anwenden eines Buttons wird das Programm weder beendet, noch wird es erneut gestartet. Dies könnte einem Implementierungsfehler zu Grunde liegen.
 - Akteure: Benutzer, Endgerät
 - Auslösendes Ereignis: Der Benutzer hat einen der vorhandenen Buttons betätigt.
 - Beschreibung: Nachdem der Benutzer die Auswertung seiner Übung eine für sich angemessene Zeit lang betrachtet hat, kann er nun selbstständig wählen, ob er das Programm gänzlich schließen oder es aber neu starten möchte, um weitere Übungen ausführen zu können.
 - Alternative: Das Programm wird, nachdem das Bewertungsfenster eine vorgegebene Zeit sichtbar war, komplett beendet und kann nun nur manuell neu gestartet werden.

- /F1200/ Framework

- Geschäftsprozess: Bereitstellung eines Frameworks
- Kategorie: Sekundär
- Ziel der Funktion: Die Software soll leicht erweiterbar und flexibel sein, damit die Möglichkeit besteht weitere Sensoren und Übungen hinzuzufügen.
- Vorbedingung: Hierarchien innerhalb der Software müssen logisch aufgebaut und Schnittstellen für die Sensoren vorhanden sein.
- Nachbedingung Erfolg: Es besteht die Möglichkeit verschiedene Sensoren, ohne größeren Implementierungsaufwand vornehmen zu müssen, anzuschließen, verschiedene Übungen mit den unterschiedlichen Sensoren durchzuführen und eine generische Bewertung zu gewährleisten.
- Nachbedingung Fehlschlag: Wenn die Hierarchien des Frameworks nicht richtig definiert sind ist es nicht möglich weitere Sensoren anzusprechen, Übungen hinzuzufügen oder eine generische Bewertung zu erhalten.
- Akteure: Informatiker
- Beschreibung: Es soll ein Framework implementiert werden, welches auf einfache Art und Weise die Möglichkeit bietet, Erweiterungen vorzunehmen, sodass neue Sensoren erkannt, die von ihnen gesendeten Daten übertragen und gespeichert werden können. Desweiteren soll die Option vorhanden sein, zusätzliche Übungen einzufügen und deren Bewertungen generisch vorzunehmen.

6 Produktdaten

6.1 Benutzerdaten

- /D10/ Benutzername
 - Vorname, Nachname
- /D20/ Geburtsdatum
 - Alter
- /D30/ Gewicht (in kg)

6.2 Trainingsdaten

- /D40/ Datum
 - Tag, Monat, Jahr
 - Uhrzeit
- /D50/ Trainingsdauer
 - Minuten, Sekunden

6.3 Bewertungsdaten

- /D60/ Rohdaten
 - Als Diagramm
- /D70/ Mittelwert

- /D80/ Bewertung
 - Ampelsystem
 - * Gut, Mittel, Schlecht
 - Punktesystem

6.4 Sensordaten

- /D90/ G-Sensordaten
 - Uncodierte 24 Byte Daten, aufgeteilt auf:
 - * TOP RIGHT
 - * TOP LEFT
 - * BOTTOM RIGHT
 - * BOTTOM LEFT

7 Nichtfunktionale Anforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		x		
Richtigkeit			x	
Interoperabilität		x		
Ordnungsmäßigkeit		x		

Dabei beschränkt sich die Sicherheit hauptsächlich auf die Benutzerdaten.

Produktqualität	sehr gut	gut	normal	nicht relevant
Sicherheit				
Zuverlässigkeit		x		
Reife			x	
Fehlertoleranz	x			
Wiederherstellbarkeit		x		

Produktqualität	sehr gut	gut	normal	nicht relevant
Benutzbarkeit				
Verständlichkeit	x			
Erlernbarkeit	x			
Bedienbarkeit	x			
Effizienz			x	
Zeitverhalten			x	
Verbrauchsverhalten			x	

Die Software sollte intuitiv zu bedienen sein, um die Einarbeitungszeit so gering wie möglich zu halten. Die Software sollte ressourcenschonend arbeiten und keine unnötige Rechnerkapazität verschwenden.

Produktqualität	sehr gut	gut	normal	nicht relevant
Änderbarkeit				
Analysierbarkeit	x			
Modifizierbarkeit	x			
Stabilität		x		
Prüfbarkeit	x			

Das zusätzlich implementierte Framework sollte sehr gut erweiterbar sein, damit zu einem späteren Zeitpunkt gegebenenfalls weitere Sensoren angesprochen werden können und die Software mit weiteren Trainingseinheiten erweitert werden kann.

Produktqualität	sehr gut	gut	normal	nicht relevant
Übertragbarkeit				
Anpassbarkeit	x			
Installierbarkeit	x			
Konformität		x		
Austauschbarkeit	x			

Dabei sollte darauf geachtet werden, dass die Software die Sicherheit und Stabilität des Systems nicht beeinflusst.

Die folgenden Punkte beinhalten sowohl die Pflichtenforderungen, welche essentiell für das Projekt sind, als auch Wunschanforderungen.

- /Q10/ Die Funktion /F100/ darf nicht länger als 30 Sekunden benötigen.
- /Q20/ Die Funktion /F200/ darf nicht länger als 5 Sekunden benötigen.
- /Q30/ Die Funktion /F1000/ darf nicht länger als 2 Minuten benötigen.
- /Q40/ Alle Reaktionszeiten auf Benutzeraktionen müssen unter 2 Sekunden liegen (außer Funktion /F1000/).
- /Q50/ Das Produkt soll Plattformunabhängig sein.
- /Q60/ Das Produkt soll Benutzerfreundlich sein. Es soll auch für Benutzer mit geringen EDV-Kenntnissen intuitiv nutzbar sein.
- /Q70/ Die Programmstruktur soll mit wenig Zeitaufwand für Außenstehende verständlich sein. Es soll ein gut strukturierter und kommentierter Quellcode vorhanden sein.
- /Q80/ Kommentare und Quellcode sollen in Englisch verfasst werden.

- /Q90/ Notwendige Kommentare sollen nach Javadoc-Standards erfolgen.
- /Q100/ Die Software soll leicht erweiterbar und flexibel sein (Neue Übung, neuer Sensor, etc.).
- /Q110/ Das Produkt muss ausreichend stabil laufen.
- /Q120/ Das Produkt soll fehlertolerant bezüglich Bedien- und Eingabefehler sein.

8 Benutzeroberfläche

- /B10/ Javagenerierte Graphical User Interface (GUI)
- /B20/ Fensterlayout
 - Übungswahlfenster
 - * Anzeige der wählbaren Übungen
 - Anklickbare Buttons
 - * Beenden-Button
 - Erklärungsfenster
 - * Textuelle Beschreibung der Übung
 - * Start-Button
 - Fenster des Übungsablaufs
 - * Zeitanzeige
 - * Graphische Anzeige der Bewegungsimpulse
 - Eingabefenster für Benutzerdaten
 - * Textfelder für die Aufnahme verschiedener Daten
 - * Button zum Fortsetzen
 - Bewertungsfenster
 - * Ampel
 - * Punktesystem
 - * Beenden-Button

9 Technische Produktumgebung

9.1 Software

Betriebssystem: Windows/Unix

Bibliothek: Wii-Bibliothek (zum Beispiel: Wii-RemoteJ)

9.2 Hardware

Hardware-Plattform:

- Handelsüblicher PC/Notebook
- Wii-Balance Board

Wireless-Hardware: Bluetooth

9.3 Orgware

Nicht relevant

9.4 Produktschnittstellen

Das Abspeichern der Programmdateien erfolgt in Text-Format.

10 Glossar

1. **Ambulante Rehaversorgung:** Bei einer ambulanten Rehaversorgung wird es dem Patienten ermöglicht, weiterhin zu Hause zu wohnen und trotzdem eine hochwertige und intensive Rehabilitation in einer zugelassenen Rehaeinrichtung in der Nähe des Wohnortes durchzuführen.
2. **Assistierende Gesundheitstechnologien:** Assistierende Gesundheitstechnologien weisen ein großes Potential auf, die Selbstständigkeit älterer Personen in ihrer gewohnten häuslichen Umgebung so lange wie möglich zu erhalten bzw. wiederherzustellen. Dies ist nur möglich, wenn begleitend innovative Formen von Versorgungsstrukturen und medizinischen Dienstleistungen entwickelt und etabliert werden.
3. **Button(JAVA):** Eine Schaltfläche, die für einen Anwender verfügbar ist, wenn dieser eine Aktion auslösen möchte. Schaltflächen sind meist beschriftet und stellen eine Zeichenkette dar.
4. **EDV:** Die Abkürzung steht für “Elektronische Datenverarbeitung” und ist ein Sammelbegriff für die Erfassung und Bearbeitung von Daten durch elektronische Maschinen oder Computer.
5. **Exception (englisch für “Ausnahme“):** Eine Ausnahme bezeichnet in der Computertechnik ein Verfahren, Informationen über bestimmte Programzustände (meistens Fehlerzustände) an andere Programmebenen zur Weiterbehandlung weiterzureichen.
6. **Framework (englisch für “Rahmenstruktur“ oder “Fachwerk“):** Ein Framework wird speziell in der Softwaretechnik verwendet. Dort dient dieses als ein Programmiergerüst, vor allem im Rahmen der objektorientierten Softwareentwicklung, sowie bei komponentenbasierten Entwicklungsansätzen.
7. **Genesungsvorgang:** Unter Genesungsvorgang versteht man allgemein den Prozess zur vollständigen Genesung über einen bestimmten Zeitraum.
8. **GUI:** GUI ist die Abkürzung für “Graphical User Interface” und heißt übersetzt “graphische Benutzerschnittstelle“. Diese ist eine Software-Komponente, die dem Benutzer eines Computers die Interaktion mit der Maschine über graphische Symbole erlaubt.

9. **Hierarchien(Framework):** Hierarchien sind Ordnungssysteme innerhalb eines Systems, bei denen es ein ausgezeichnetes oberstes Element gibt und jedes andere Element genau ein übergeordnetes Element, aber beliebig viele untergeordnete Elemente hat.
10. **Implementierung:** In der Softwareentwicklung ist die Implementierung das Umsetzen eines Algorithmus oder Softwareentwurfs in ein Computerprogramm.
11. **Javadoc:** Javadoc ist ein Dokumentationswerkzeug für Java-Quelltexte, die daraus automatisch eine HTML-Dokumentationsdatei erstellt.
12. **MAC-Adresse:** Die MAC-Adresse (Media-Access-Control-Adresse) ist die Hardware-Adresse jedes einzelnen Netzwerkadapters, die zur eindeutigen Identifizierung des Geräts in einem Rechnernetz dient.
13. **PLRI:** PLRI ist eine Abkürzung für "Peter L. Reichertz Institut", welches sich mit der Medizinischen Informatik beschäftigt.
14. **Quellcode:** Ein Quellcode ist ein aus Befehlen, Anfragen, Anweisungen oder Ähnlichem aufgebauter Programmcode einer beliebigen Software. Der Code kann in verschiedenen Programmiersprachen geschrieben werden (Java, C++).
15. **Schnittstelle:** Eine Schnittstelle dient in der objektorientierten Programmierung der Vereinbarung gemeinsamer Strukturen von Methoden, welche in unterschiedlichen Klassen implementiert werden.
16. **Temporäre Dateien:** Temporäre Dateien sind Dateien, die vom Betriebssystem oder von anderen Programmen verwendet werden und zur zeitlich begrenzten Speicherung von Daten dienen. Diese sollen vom selben oder einem anderen Programm weiterverarbeitet werden können.
17. **Textfeld(JAVA):** Ein Textfeld ist ein Steuerelement einer graphischen Benutzeroberfläche, das Benutzereingaben oder Programmausgaben in Form einer Zeichenkette aufnehmen kann.
18. **Use-Case-Diagramm:** Ein Use-Case-Diagramm ist eines von diversen Diagrammarten, welches dazu dient Strukturen und das Verhalten von Software- und anderen Systemen zu modellieren. Dabei stellt es Anwendungsfälle und Akteure mit ihren jeweiligen Abhängigkeiten und Beziehungen dar.
19. **Unix:** Unix ist ein Mehrbenutzer-Betriebssystem, d.h. es hat die Fähigkeit, Arbeitsumgebungen für verschiedene Benutzer bereitzustellen und voneinander abgrenzen zu können.
20. **Wasserfallmodell:** Das Wasserfallmodell ist ein, besonders für die Softwareentwicklung, konstruiertes Vorgehensmodell, welches sich wiederum in bestimmte Phasen unterteilen

lässt. Dabei arbeitet man die Phasen nach der gegebenen Reihenfolge ab. (1. Analyse 2. Entwurf 3. Implementierung 4. Test, Integration 5. Wartung)

21. **Wii- Balance Board:** Die Wii ist im Allgemeinen eine fernsehgebundene (stationäre) Videospiel-Konsole. Beim Wii-Balance Board handelt es sich um eine Balance-Körper-Waage, die aus vier betretbaren Sensorflächen besteht. Position und Verlagerungen des Körpergewichts werden per Bluetooth an die Konsole gesendet, um das Spiel zu beeinflussen.
22. **Wii-RemoteJ:** Wii-RemoteJ ist eine freie Java API und Bibliothek für Bluetooth-Kommunikation mit Wiimote. Wii-RemoteJ bietet eine Schnittstelle für Java-Entwickler.
23. **Windows:** Windows ist ein Betriebssystem des Unternehmens Microsoft.