



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG



RUPRECHT KARLS UNIVERSITÄT HEIDELBERG
HOCHSCHULE HEILBRONN

Bachelorarbeit im Studiengang Medizinische Informatik

Zur Erlangung des akademischen Grades Bachelor of Science

Re-Engineering einer Trainingssoftware für hirnbeschädigte Patienten

Betreuung:

Referent: Prof. Dr. Martin Haag, Hochschule Heilbronn

Korreferent: Prof. Dr. Einrich Kasten, Universität Hamburg

Eingereicht von:

Stephanie Gnipieven

Matrikelnummer: 175883

Studiengang: Medizinische Informatik Bachelor

Hiermit erkläre ich, die vorliegende Bachelorarbeit selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Außer den angegebenen Quellen habe ich keine weiteren Hilfsmittel verwendet.

Heilbronn, den 13.03.2014

Stephanie Gnipieven

Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Danksagung.....	iii
Abbildungsverzeichnis.....	iv
Einleitung.....	1
1.1 Motivation.....	1
1.2 Problematik und Zielsetzung.....	2
1.3 Aufbau der Arbeit.....	3
2 Grundlagen.....	4
2.1 Das menschliche Gehirn.....	4
2.1.1 Gedächtnis und Lernen.....	7
2.1.1.1 Das Kurzzeitgedächtnis.....	9
2.1.1.2 Das Langzeitgedächtnis.....	9
2.2 Gedächtnisstörung nach Hirnschädigung.....	10
2.2.1 Neuropsychologische Therapie: Computergestütztes Training kognitiver Fähigkeiten.....	11
2.3 JQuery.....	13
3 Analyse und Konzeption.....	16
3.1 Anforderungsanalyse.....	16
3.1.1 Analyse des jetzigen Programms.....	16
3.1.2 Anforderung an die neue Applikation.....	22
3.1.2.1 Qualitätszielbestimmungen.....	25
3.2 Entscheidung für das neue System.....	26
3.2.1 Programmiersprache.....	27
3.2.2 Konzeption: Architektur und Entwurf der grafischen Benutzeroberfläche.....	28
4 Implementierung.....	31
4.1 Shortcuts.....	34
4.2 Sounds.....	36
4.3 Speichern von Statistiken.....	37
4.4 Implementierung der Spiellogik.....	41
5 Test.....	44

5.1	Testfall: Beginnen einer Spielrunde	44
5.2	Testfall: Anzahl angezeigter Bilder passend zu der eingegebenen Nummer	44
5.3	Testfall: Switchen zwischen den Portraits	44
5.4	Testfall: Korrekte Statistiken, Funktionieren des Tons und der Tastenkürzel	44
5.5	Testfall: Ende der Spielrunde: Anzeigen und Speicherung der Ergebnisse	45
6	Zusammenfassung und Ausblick	46
Anhang A: Literatur		48
A.1	Bücher	48
A.2	InternetSeiten	48

Danksagung

Hiermit möchte ich mich ganz herzlich bei meinem Betreuer Prof. Dr. Haag für seine zahlreichen Tipps in allen Phasen dieser Arbeit bedanken. Ich möchte mich weiter bei Prof. Dr. Kasten bedanken für die Informationen, die er mir während dieser Arbeit, trotz der Entfernung zwischen uns, gegeben hat.

Ein besonderer Dank geht an meine Familie, insbesondere meine liebevollen Mutter Martine, für alles was sie für mich mein lebenslang gemacht hat. Weiterhin möchte ich mich auch bei meinen besten Mitstreiterinnen, Freundinnen, Schwestern, Sandra und Daniele für Ihren Beistand, Ihre Ratschläge jeden Tag in meinem Leben und auch während meiner Thesis bedanken.

An meinen Freunden insbesondere Marcel und Sonia für eure liebevolle Aufmunterungen, Tipps will ich einen herzlichen Dank heißen, ohne euch hätte ich es nicht geschafft!

Ein letzter Dank geht auch an meinen sehr guten Freund Marcus für die sprachliche Korrektur dieser Arbeit.

Abbildungsverzeichnis

Abbildung 1- Gehirn Lappen, [BrainLa].....	5
Abbildung 2- Transversaler Schnitt des menschlichen Gehirns [BrainHu]	6
Abbildung 3- Gedächtnismodell: Zeitliche Dimension, vgl. [GedMo]	10
Abbildung 4- Einstieg in die Module	17
Abbildung 5- Modul "Gesichter merken" Anzahl Personen auswählen	17
Abbildung 6- Modul "Gesichter merken", Spielrunde	18
Abbildung 7- Modul "Wie merkt man sich Namen und Gesichter?"	18
Abbildung 8- Aktivitätsdiagramm: Darstellung des Kernablaufs der Module "Gesichter- merken" und "Wie merkt man sich Namen und Gesichter?"	20
Abbildung 9- Systemarchitektur des alten Programs	21
Abbildung 10- Auszug aus dem Modul "Gesichter merken"	22
Abbildung 11- Use-Case-Diagramm: Funktionalitäten der neuen Software	23
Abbildung 12- Systemarchitektur der neuen Anwendung	29
Abbildung 13- Startseite der neuen Applikation	32
Abbildung 14- Benutzeroberfläche des Moduls „Wie merkt man sich Namen und Gesichter?“	33
Abbildung 15- Benutzeroberfläche des Moduls „Gesichter merken“	33
Abbildung 16- Deutsche Tastatur	34
Abbildung 17- Shortcuts Implementierung	35
Abbildung 18- Sound Implementierung 1	36
Abbildung 19- Sound Implementierung 2	37
Abbildung 20- Statistiken Implementierung 1	37
Abbildung 21 - Statistiken Implementierung 2	38
Abbildung 22- Statistiken speichern: Php-Script in HTML Formular	39
Abbildung 23- Statistiken speichern: Php-Script	40
Abbildung 24- Gespeicherte Statistiken geöffnet mit IE	40
Abbildung 25- Aufbau der JavaScript-Datei Gesichter.js: Eigenschaft- und Funktionsübersicht	41
Abbildung 26- Dynamische Generierung von Antwortmöglichkeiten in Gesichter.js	42
Abbildung 27- Prototype-Eigenschaft von JavaScript : Codeauszug aus Gesichter.js	43

Tabelle 1- Qualitätszielbestimmungen..... 26

Liste der verwendeten Abkürzungen

NPT-PC	Neuropsychologische Therapie am PC
ZNS	Zentralnervensystems
HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
Ajax	Asynchronous JavaScript and XML
GWT	Google Web Kit
YUI	Yahoo User Interface Library
DOM	Document Object Model
CSS	Cascading Style Sheets
UML	Unified Modeling Language
IE	Internet Explorer
PHP	Hypertext Preprocessor, ursprünglich Personal Home Page
GUI	Graphical User Interface

Einleitung

1.1 Motivation

Ein Schlaganfall, ein Tumor, ein schwerer Verkehrs- oder Sportunfall, ein unglücklicher Sturz auf den Kopf und das Leben der Betroffenen ändert sich von heute auf morgen. Ehemals aktive und selbständige Menschen sind aufgrund einer erworbenen Hirnschädigung rund um die Uhr auf Hilfe angewiesen.

Menschen mit erworbener Hirnschädigung leiden an leichten bis schweren überwiegend kognitiven Schädigungen, Kurzzeitgedächtnis-, Langzeitgedächtnis- und Wahrnehmungsstörungen. Nach einem Krankenhausaufenthalt sind sie weiterhin auf neuropsychologische Therapien angewiesen.

„Unter neuropsychologischer Therapie wird das Training kognitiver Funktionen verstanden, welche durch erworbene Hirnschädigungen beeinträchtigt wurden.“(Grimm 1983 zitiert nach [Kast01], S. 13).

Eine unter zahlreichen neuropsychologischen Behandlungsmethoden ist die NPT-PC (Neuropsychologische Therapie am PC). Sie beruht auf einem Auswahlprozess von Trainingssoftware unter Berücksichtigung von individuellen Patientenparametern und dem therapeutischen Ziel. Sie erfolgt in den Bereichen ambulant, stationär sowie in unterschiedlichen Rehabilitationsphasen mit unterschiedlichen Zielsetzungen, vom Training von Basisleistungen bis hin zu komplexen, berufsrelevanten Integrationsleistungen. Allerdings konnte für die NPT-PC bisher schon bei beeinträchtigten Hirnfunktionen in den Funktionsbereichen Aufmerksamkeit und visuelle Wahrnehmung eine effiziente Wirksamkeit belegt werden. Außerdem kann die NPT-PC in den frühesten Phasen der Rehabilitation selbst bei Schwerbehinderten einen wichtigen Beitrag zur Stabilisierung basaler Aufmerksamkeits- und Wahrnehmungsfähigkeiten leisten und auch erste Selbstwirksamkeitserfahrungen und Erfolgserlebnisse ermöglichen (vgl. [Kast01] S.372-373).

Des Weiteren eröffnet die besondere Form der Interaktion (Patient-PC-Therapeut) und die automatische Registrierung des Leistungsverhaltens dem Therapeuten die Möglichkeit zur ausgedehnten Verhaltensbeobachtung und schafft Freiräume für direkte adaptive Interventionen. Infolgedessen kann die Therapie flexibel auf die Bedürfnisse und Erfordernisse des einzelnen Patienten ausgerichtet werden und die Gesamtheit der Rehabilitationsproblematik eines Patienten auch in ihren psychosozialen Dimensionen berücksichtigt werden (vgl. [Kast01] S. 372, S. 374).

Die Pflege der angewendeten Software ist ein sehr wichtiger Aspekt beim Computertraining und von daher nicht zu unterschätzen. Aufgrund der neuen medizinischen bzw. neuropsychologischen Erkenntnisse veralten diese Software und die dazu gehörige Hardware sehr schnell. Sie müssen also zur optimalen Behandlung der Patienten ständig auf den neuesten Stand gebracht werden.

1.2 Problematik und Zielsetzung

Zur Unterstützung von Patienten mit erworbener Hirnschädigung und daraus resultierender Konzentration- und Gedächtnisstörung wurde die unter Windows lauffähige Version (2.0) der Software "Gesichter" von Prof. Dr. Einrich Kasten, Psychologe und Sozialtherapeut, im Jahr 2006 implementiert. Das Programm unterteilt sich in drei Module und zwar in "Gesichter merken", "who-is-who?" und "wie merkt man sich Namen und Gesichter?".

Das Programm wurde für ein neues Release immer wieder gepatched, damit es auf neuen Systemen lauffähig bleibt. Außerdem enthält die Software noch viele Bugs. Des Weiteren lässt sich der zum Programm gehörende Ton seit Windows 7 nicht mehr abspielen.

Im Augenblick wird das Programm noch vom Hogrefe-Verlag verkauft, aber inzwischen mit lächerlich geringen Zahlen, da die Versionen auf modernen PCs nicht mehr fehlerfrei laufen.

Im Rahmen dieser Arbeit wird versucht, ein paar Module der Software "Gesichter" neu zu entwickeln. In erster Linie wird das Programm auf Benutzerfreundlichkeit inspiziert, anschließend wird die Programmlogik angesichts neuer medizinischer Erkenntnisse zur Therapie erworbener Hirnschädigungen mit dem PC überprüft. Abschließend wird entschieden, ob eine Neuimplementierung erforderlich ist oder ob das Programm lediglich verbessert werden muss.

Das neue Programm wird mindestens folgende Funktionalitäten enthalten: Die Möglichkeit, die Anwendung mit der Tastatur zu steuern und die Möglichkeit, eine akustische Wiedergabe bei Antworten zu zulassen oder zu unterdrücken. Außerdem wird dem Patienten eine kurze Einleitung zum Modul zur Verfügung gestellt. Ein weiterer wichtiger Punkt ist die Möglichkeit, die Ergebnisse der unterschiedlichen Aufgaben zu speichern und auch ausdrucken zu können, damit man die Fortschritte des Patienten beobachten kann.

All dies wird mit einer hohen Benutzerfreundlichkeit entwickelt. Das Programm muss möglichst autonom sein, damit der Benutzer möglichst wenig konfigurieren muss.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in sechs Kapitel. Kapitel 1 stellt die Problematik der Arbeit dar. Nach einer Darlegung der Gefälligkeit des Computer-Trainings werden die vorliegende Software “Gesichter“ und ihre Schwächen sowie mögliche Verbesserungen angesichts des aktuellen Stands der Technik vorgestellt. Die Vorgehensweise wird in der Zielsetzung beschrieben.

Im Kapitel 2 werden wichtige medizinische und technische Konzepte vorgestellt, die für die Realisierung der neuen Applikation erforderlich sind. Außerdem wird das menschliche Gehirn vorgestellt. Es wird anschließend gezeigt, wie eine erworbene Hirnschädigung zu Stande kommt und welche Strukturen im Gehirn dabei betroffen sind.

Kapitel 3 beschäftigt sich hauptsächlich mit der Untersuchung der vorliegenden Software sowie mit der konkreten Evaluierung der Vorgehensweise und der Auswahl der geeigneten Technologie zur Realisierung der neuen Anwendung.

Kapitel 4 beschreibt, wie die Implementierung realisiert wurde. Dabei wird auf die Implementierung einzelner definierten Anforderungen eingegangen.

In Kapitel 5 werden Testfälle, die mit dem neuen Programm durchgeführt werden können, vorgestellt.

Das letzte Kapitel 6 fasst die Arbeit zusammen. Hierbei werden alle erreichten Ergebnisse entsprechend der Zielsetzung vorgestellt. Abschließend werden im Ausblick Erweiterungsmöglichkeiten für das neue Programm vorgeschlagen.

2 Grundlagen

2.1 Das menschliche Gehirn

Als Gehirn bezeichnet man den im Kopf gelegenen Teil des zentralen Nervensystems ZNS (Gehirn und Rückenmark) der Wirbeltiere. Mit nur 2% unseres Körpergewichts ist das menschliche Gehirn die komplexeste Ansammlung von Materie auf unserem Planeten (vgl. [Huch07] S.172).

Es liegt geschützt in der Schädelhöhle, wird umhüllt von Hirnhaut und besteht hauptsächlich aus Nervengewebe. „Die Gesamtheit des Nervengewebes des Menschen wird als Nervensystem bezeichnet. Das Nervensystem dient zur Erfassung, Verarbeitung, Speicherung und Aussendung von Informationen“ ([Huch07] S.156).

Das menschliche Gehirn unterscheidet sich von dem anderer Lebewesen durch die Fähigkeit logisch zu denken, mit den Mitmenschen zu kommunizieren sowie der Größe des Erinnerungsvermögens.

Das Gehirn besteht hauptsächlich aus vier Teilen: Das Großhirn, das Zwischenhirn, das Kleinhirn und der Hirnstamm.

Das Großhirn bildet den größten Teil des menschlichen Gehirns und den differenziertesten des ZNS. Es ist verantwortlich für viele Denk- und Handlungsprozesse. Durch den Interhemisphärenspalt ist es in zwei Halbkugeln (Hemisphären) getrennt. Die Hemisphären können unter Bezugnahme auf die sie umhüllenden Schädelknochen nochmals in je vier Lappen gegliedert werden: der Frontallappen, der Parietallappen, der Temporallappen, der Okzipitallappen (Siehe Abbildung 1)(vgl. [GroHirn]).

Die Abbildung 1 zeigt einen transversalen Schnitt des Gehirns mit seinen vier Lappen.

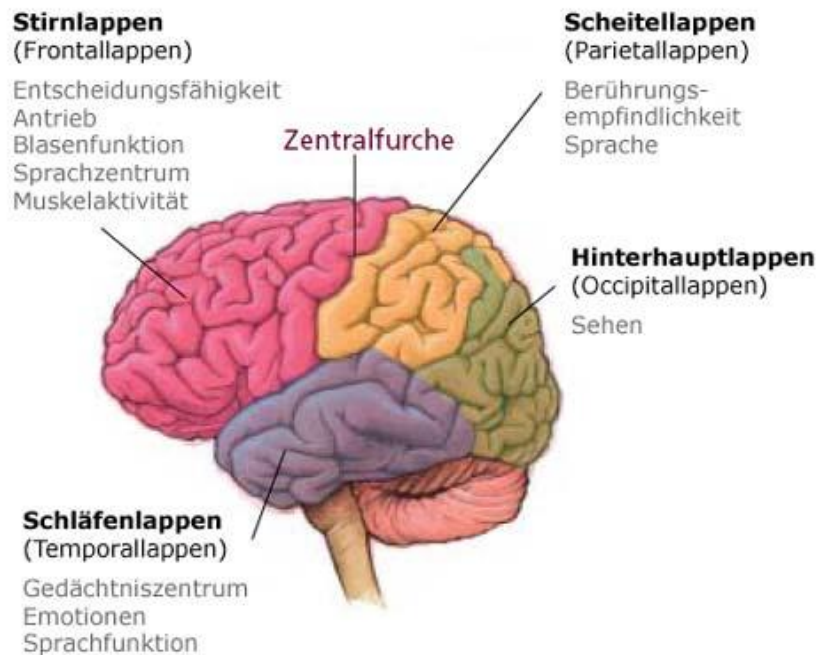


Abbildung 1- Gehirn Lappen, [BrainLa]

Der Hirnstamm ist der unterste und älteste Gehirnabschnitt. Er ist in drei Teile gegliedert: Mittelhirn, Brücke, verlängertes Mark.

Das Zwischenhirn (Diencephalon) ist Schaltstelle zwischen Großhirn und Hirnstamm. Es besteht hauptsächlich aus dem Thalamus und dem Hypothalamus, an dem die Hypophyse hängt. Es erfüllt wichtige vegetative Aufgaben – dazu zählt unter anderem die Haltung der Balance zwischen Sympathikus und Parasympathikus sowie die Steuerung des Biorhythmus. Dabei arbeitet das Diencephalon eng mit dem Großhirn zusammen. (vgl. [Huch07] S.175, [ZwiHirn]).

Das Kleinhirn liegt unterhalb des Hinterhauptlappens des Großhirns und hinter der Brücke. Es ist für die Planung, Koordination und Feinabstimmung von Bewegungen verantwortlich.

Die funktionelle Einheit, die aus Strukturen des Großhirn, des Zwischenhirns und des Mittelhirns gebildet wird, heißt limbisches System. Dieses entwicklungsgeschichtlich sehr alte System spielt eine führende Rolle bei der Entstehung von Gefühlen (Angst, Wut, sexuellen Wünschen), dem Lernen und dem Gedächtnis. Zum limbischen System gehören unter anderem: der Mandelkern, der Hippocampus und Teile des Hypothalamus (vgl. [Huch07] S.177).

Der Hippocampus liegt im Temporallappen und ist vor allem an der Gedächtnisbildung beteiligt. Durch beidseitige Läsion (Schädigung) des Hippocampus kann es zu Ausfällen im Temporallappen kommen, welche eine Gedächtnisstörung verursachen, so dass neue Informationen kaum behalten werden können; altes Wissen bleibt dabei aber unverändert (vgl. [Schu12], S.17, [Kast07], S.42).

Die Abbildung 2 zeigt einen transversalen Schnitt des menschlichen Gehirns mit seinen wichtigsten Teilen.

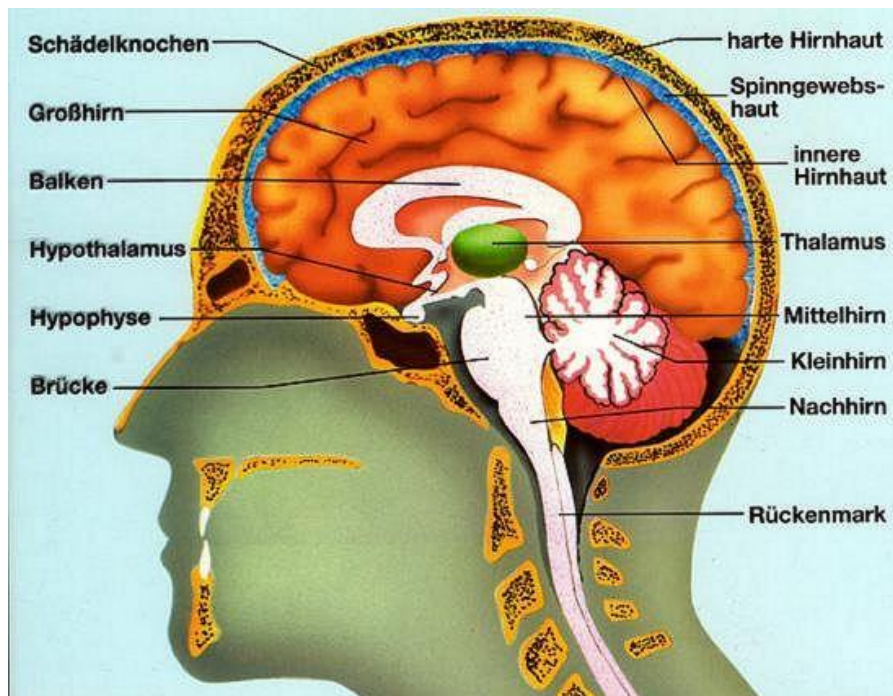


Abbildung 2- Transversaler Schnitt des menschlichen Gehirns [BrainHu]

2.1.1 Gedächtnis und Lernen

Unter Gedächtnis wird die lernabhängige Speicherung ontogenetisch¹ erworbener Informationen verstanden, die sich in phylogenetischen² neuronalen Strukturen selektiv artgemäß einfügt und für ein situationsangepasstes Verhalten verfügbar gemacht werden kann. (Sinz(1979, S.19) zitiert nach [Otto05] S.3). In anderen Wörter ist das Gedächtnis die Fähigkeit des Gehirns, neue Informationen zu speichern, diese zu einem späteren Zeitpunkt wieder zu aktivieren und unwichtige, nicht gebrauchte Informationen zurückzudrängen, um Speicherplatz für neue, wichtigere Informationen zu schaffen.

Nach dem Modell der *parallel-distributiven Verarbeitung* von Rumelhardt und McClelland werden Informationen im Gehirn lediglich in Form von Aktionsmustern repräsentiert, die kleine Neuronenverbände (Zell-Assemblies) umfassen, welche über erregende und hemmende Verschachtelung miteinander verknüpft werden.

Zur Wiedererkennung von Gesichtern wird also eine umrissene Gruppe von Neuronen benötigt. Abhängig von den Neuronen, die diese Zell-Assemblies durch einlaufende Informationen erregen und welche gehemmt werden, ist das Gehirn in der Lage, ein Gesicht wiederzuerkennen. Es sind also die neuronalen Untereinheiten für Gesichtsform, Haarpracht, Stimme sowie Körperkontur (vgl. [Kast07], S.44-45).

Außerdem unterscheidet man zwei qualitativ verschiedene Arten von Gedächtnis. Das deklarative Gedächtnis, was das Faktenwissen beinhaltet, wie z.B. Namen, Zahlen und deren Bedeutung. Diese Inhalte können in Sprachform (deklarativ) wiedergegeben werden. Beteiligte Hirnstrukturen sind der Hippocampus, Teile des Thalamus und der Schläfenlappen des Großhirns. Das nicht deklarative Gedächtnis speichert unter anderem Informationen über die Fähigkeit, bestimmte Dinge auszuführen, wie z.B. das Schreiben oder ein Instrument zu spielen. Abhängig von der Lernaufgabe sind die Großhirnrinde, der Streifenkörper, der Mandelkern und das Kleinhirn beteiligt (vgl. [Huch07], S182).

Um einen Denkprozess durchzuführen, arbeitet das bewusste Denken sequentiell, Schritt für Schritt ab. Dabei muss sich das Gehirn die Aufgabe, Zwischenschritte und den eigentlichen Arbeitsschritt merken (vgl. [Kast07], S.47). Somit kann das Gehirn wie ein Arbeitsspeicher angesehen werden, der eine begrenzte Kapazität hat.

¹ *Entwicklungsbiologie*: die Entwicklung des Individuums betreffend

² *Biologie, Evolution*: die Stammesentwicklung der Lebewesen, die biologische Entwicklung der Menschheit betreffend

Nach dem Gedächtnismodell von Atkinson und Shiffrin von 1968 wird das Gedächtnis grundsätzlich in drei zeitliche Stadien unterteilt und zwar in den sensorischen Informationsspeicher (Ultrakurzzeitgedächtnis), das Kurzzeitgedächtnis und das Langzeitgedächtnis. Die einzelnen Stadien können wie einzelne Arbeitsspeicher mit unterschiedlichen Kapazitäten und Speicherzeit dargestellt werden (vgl. [Huch07], S.182) (Siehe Abbildung 3).

Damit eine Information Eingang ins Gedächtnis finden kann, muss zunächst einmal ein Reiz auf eine Sinneszelle treffen. In Form eines elektrischen Erregungsimpulses wird der Reiz an ein Neuron weitergeleitet. Die einzelnen Neuronen sind über Synapsen miteinander verbunden. Bei jeder neuen Information werden Synapsen neu aktiviert. Wenn die Information im Gehirn verankert und damit nutzbar bleiben soll, muss sie in einer spezifischen Synapsenkombination kodiert werden.

Über diese Synapsen kann nun die Information von der ersten Nervenzelle in andere Nervenzellen weitergeleitet werden. Je mehr Synapsen – und damit Nervenzellen - beteiligt sind, umso stärker wird die Information im Gehirn verankert und desto besser kann man sich an sie erinnern. Wenn der Reiz wiederholt wird – zum Beispiel beim Lernen – werden auch die bei der Informationsverarbeitung beteiligten Synapsen erneut aktiviert. Die Kontakte zwischen den einzelnen Neuronen werden somit verstärkt. Die häufige oder regelmäßige Verwendung macht die mit dieser Information befassten Nervenkontakte schneller und bedeutender im Konzert der vielen Milliarden Gehirnzellen.

Daher werden ständig in Gebrauch befindliche Nervenkontakte gegenüber anderen bevorzugt. Auch Wiederholungen und die regelmäßige Nutzung von Informationen tragen dazu bei, dass man sich besser an die Information erinnern kann.

Aus der Tatsache, dass sich die Synapsen bei jeder neuen Information neu kombinieren, ergibt sich theoretisch ein Problem. Die in den weniger in Gebrauch befindlichen Strukturen gespeicherten Informationen werden zurückgedrängt, werden langsamer und für die schnellen Denkprozesse nicht mehr nutzbar. Dennoch gehen diese Informationen nicht vollständig verloren, jedoch wird der Zugang zu ihnen für das Gehirn schlechter auffindbar. Das nennen wir vergessen: Nicht in Gebrauch Befindliches ist für das Gehirn nicht mehr nutzbar. In der Tat geschieht dies ständig. Um jedoch zu verhindern, dass wichtige Informationen gelöscht werden, stabilisiert ein spezielles Protein jene Strukturen, die Träger dieser wichtigen Informationen sind. Auf diese Weise können wir Unwichtiges „vergessen“, um Neues zu lernen, ohne dass wir dafür auch wichtige Informationen aufgeben müssten.

2.1.1.1 Das Kurzzeitgedächtnis

Das Kurzzeitgedächtnis kann mit der Frage „Was gab es gestern Mittag zu essen?“ zusammengefasst werden.

Es hat nur eine begrenzte Kapazität von nur 7 ± 2 *Chunks*³ (vgl. [Kast07], S.48).

Es stellt die Verbindung zwischen Gegenwart und unmittelbarer Vergangenheit her. Seine Speicherinhalte überdauern Minuten bis wenige Tage und ändern sich ständig. Neu aufgenommene Informationen verdrängen die schon vorhandenen (vgl. [Huch07], S183).

Damit eine Information im Kurzzeitgedächtnis behalten wird, muss sie also stetig wiederholt werden. Wenn die Information dann für längere Zeit im Kurzzeitspeicher festgehalten wird, besteht die Chance, dass sie ins Langzeitgedächtnis überführt wird(vgl. [Kast07], S.48).

2.1.1.2 Das Langzeitgedächtnis

Das Langzeitgedächtnis nimmt im Vergleich zu dem Kurzzeitgedächtnis neue Informationen sehr langsam auf.

Was im Langzeitgedächtnis angeht ist, bleibt lebenslang fixiert, nur der Zugriff auf die Information kann vergessen werden. Die Langzeitspeicherung wird durch Wiederholung einer Information gefördert, besonders wenn sie mit bereits Gespeichertem verknüpft werden kann. Durch den Aufbau chemischer Moleküle, die für die Bildung und Modifikation von Synapsen bzw. Rezeptoren und Transmittern erforderlich sind, werden langfristige Informationen wie z.B. ein Geburtsdatum im Gehirn abgespeichert. Auch das „Vergessen“ wird auf molekularer Ebene mitgesteuert(vgl. [Huch07], S183).

Das Langzeitgedächtnis kann schließlich, solange unser Gehirn gesund bleibt, bestimmte Informationen lebenslang speichern.

³ 7 ± 2 *Chunks* sieben bis neun Worte

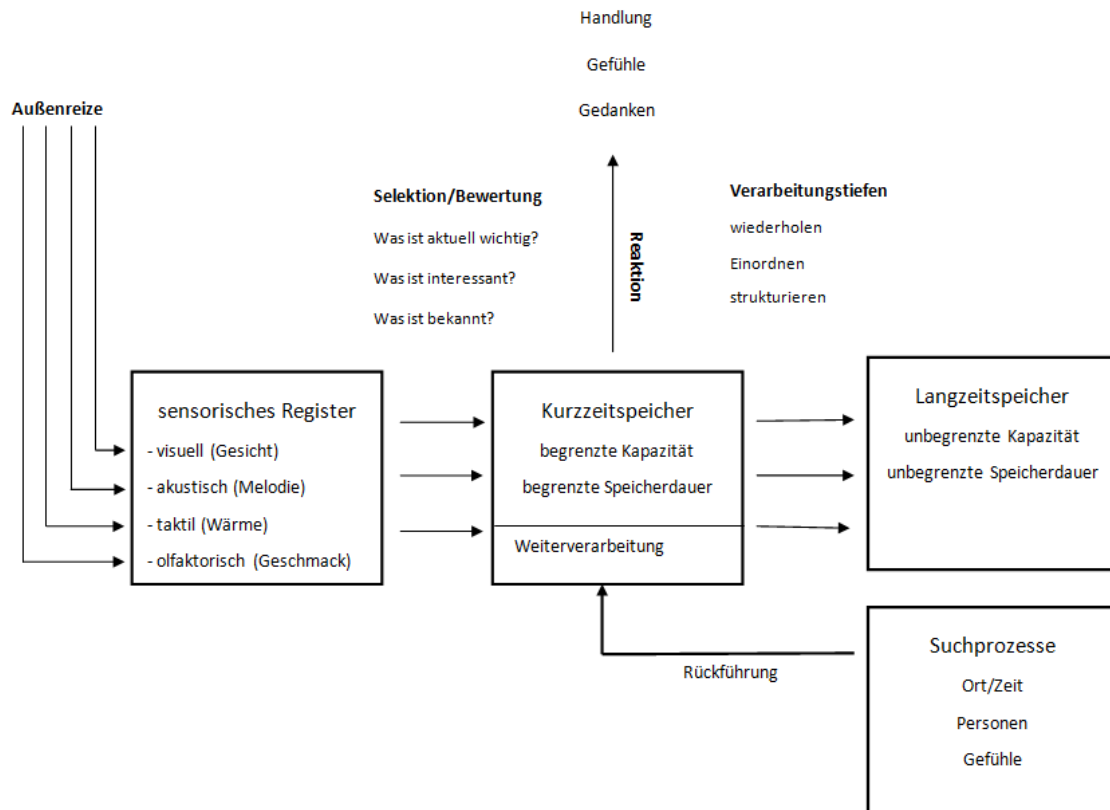


Abbildung 3- Gedächtnismodell: Zeitliche Dimension, vgl. [GedMo]

2.2 Gedächtnisstörung nach Hirnschädigung

„Erworbene Hirnschädigungen sind facettenreich, vielschichtig, häufig sehr komplex und führen oft zu bleibenden Beeinträchtigungen“ ([Schu12], S.7).

Sie können durch eine Entzündung der Hirnhäute (Dura mater, Arachnoidea, Pia Mater (Meningitis)), Entzündung des gesamten Gehirns (Enzephalitis), traumatische Hirnschäden durch Gewalteinwirkung (Schlaganfall, Hirnblutung, Schädel-Hirn-Trauma usw.), Hirnkrankheiten (Tumore, Multiple Sklerose, Demenz) und endokrine Funktionsstörungen (Hormonstörungen, Bewusstseinsstörung) entstehen(vgl. [Kast07], S.63-69).

Gedächtnisstörungen gehören zu den häufigsten kognitiven Störungen nach einer Hirnschädigung und können für die Frage der Wiedererlangung der Selbständigkeit der Patienten von großer Bedeutung sein.

„Über das Nervensystem tauschen unsere Körperteile Informationen aus, um sinnvolle Abläufe zu gewährleisten. Die Verschachtelungen sind so komplex, dass sich ein solches Nervensystem in jedem Individuum nur einmal und endgültig entwickelt. Während andere Zellen die Fähigkeit zur Teilung haben, um damit den Verlust von abgestorbenen Zellen

auszugleichen, können Neuronen dies nach der Geburt nicht mehr. Nach einer Hirnschädigung müssen die übrig gebliebenen Teile mit zusätzlichen Aufgaben belastet werden, um die verloren gegangenen Funktionen zu kompensieren.“([Kast07], S. 22).

Zur Wiedererlangung von Gedächtnisfähigkeiten sind Therapiemethoden abhängig vom Schweregrad der Gedächtnisstörung auszuwählen (vgl. [Otto05], S.51).

2.2.1 Neuropsychologische Therapie: Computergestütztes Training kognitiver Fähigkeiten

In der neuropsychologischen Therapie unterscheidet man drei grundlegende Behandlungsformen: die Funktionstherapie, die Kompensationstherapie und die Integrative Behandlungstherapie.

In dieser Arbeit wird nur auf die Funktionstherapie und zwar auf die computergestützte Therapieform eingegangen.

Bei Funktionstherapien geht es grundsätzlich um das Üben beeinträchtigter kognitiver Fähigkeiten.

Kognition kann grundsätzlich als geistige Wahrnehmung definiert werden. Die kognitiven Fähigkeiten des Menschen beschreiben seine Fähigkeit, Signale der Umwelt wahrzunehmen und weiterzuverarbeiten. Die Kognition findet immer und überall statt, sie macht die Umwelt erst zu einem Teil des menschlichen Lebens, ist sozusagen die Schnittstelle zwischen Umwelt und Gehirn. Die Kognition lässt sich in verschiedene Bereiche untergliedern:

- Wahrnehmung der Umwelt: durch die Sinne
- Aufmerksamkeit auf spezielle Geschehen: Objekte des Interesses
- Nachdenken: Verarbeitung der Information im Gehirn
- Speicherung der Information: Gedächtnisspeicherung für spätere Erinnerung
- Zuweisung von Bedeutungen: meist über die Sprache.

(vgl. Hayes 1995 zitiert nach [KogniF])

Allerdings kann die Gedächtnisleistung besonders in der Frühphase durch andere kognitive Defizite, insbesondere durch die Aufmerksamkeitsstörung, überlagert und negativ beeinflusst werden. Zur Auswahl der besten Therapieform bleiben an dieser Stelle die Meinungen zerstritten. Es muss von einer unterschiedlichen Wirksamkeit einer Therapie, abhängig von

der jeweils vorliegenden Schädigung, ausgegangen werden. Allerdings sind abhängig von der Ätiologie unterschiedliche Trainingsmethoden erfolgreich.

Jedoch haben Primär- und Meta-Analysenstudien den Beweis über die Wirksamkeit von computergestützten Therapien im Bereich Aufmerksamkeit erbracht. Die Wirksamkeit dieser Therapien kann also als gesichert angesehen werden. Auch für andere Funktionsbereiche, wie Sprachstörung und Gedächtnisstörung, sind Studien zu finden, die die Effizienz von NPT-PC belegen.

Tatsächlich kann durch Computertrainingsprogramme die Gedächtnisleistung durch die Verbesserung der Aufmerksamkeitsfunktionen gestärkt werden. Zudem haben die Patienten durch das direkte Feedback via das Trainingsprogramms die Möglichkeit, eigenständig zu lernen und sind durch das gemütliche „von zuhause aus Trainieren“ motivierter. Hinzukommend kann dank der Patient-Computer-Pfleger Interaktion während eines Trainings am Computer ein objektives oder subjektives erlebtes Leistungsversagen zum Ausgangspunkt für Gespräche zur Krankheitsverarbeitung werden.

Für den Nutzen der Therapieform am Computer sind psycho-emotionale Beeinträchtigungen, wie Depressionsphasen, Antriebschwäche, Aufmerksamkeits- Schwankungen, welche häufig bei neuropsychologischen Patienten auftreten, beim Training zu berücksichtigen.

Bei irreversiblen Beeinträchtigungen soll auf die Therapieform am PC verzichtet werden, da dadurch die Wahrnehmung und die Akzeptanz der Ereignisse verhindert wird. In diesem Fall sollte eine passende Behandlungsmethode ausgesucht werden. (vgl. [Otto05] S.51-54, [Born09] S. 281, [Kast01] S.372 - 375).

Im Bereich Therapiesoftware sind für das Training kognitiver Fähigkeiten mehrere Programme zu finden. Einige davon sind LernReha-Programme entwickelt von Prof. Dr. Erich Kasten. Sie sind eine Sammlung von Programmen, die hauptsächlich den Abbau von Defiziten im Bereich Wahrnehmung, Gedächtnisleistung, Konzentration unterstützt. Sie unterstützen das Training des Gedächtnisses (Mittelzeit-, Kurzzeit-, Langzeitgedächtnis) und der Wahrnehmung durch aktives Abfragen. Auch das Lesen, das Schreiben und das logische Denken werden gefördert.

Ein weiteres Beispiel ist sicherlich auch die Software LifeTool EF-1 von der LifeTool Solutions GmbH. Sie ist für Menschen mit erworbenen sowie angeborenen Hirnschädigungen, wie zum Beispiel Kinder mit AS(H)S oder für Personen mit Störung der Handlungsabfolgen, geeignet. Kognitive Funktionen, wie das Arbeitsgedächtnis, die

Daueraufmerksamkeit und das Verstehen von Anweisungen, werden durch Übungen mit Handlungsabfolgen und Kontrollfunktionen gefördert. Übungen werden in Form von Aufträgen in einem virtuellen Fotolabor formuliert. Dabei ist der Patient angehalten, Einstellungen, wie z.B. die Bildschärfe, die Farbe und den Bildausschnitt, gemäß schriftlich gegebener Anweisungen vorzunehmen. Somit kann ein Ziel über mehrere Schritte hinweg erreicht werden. Außerdem wird währenddessen die Eingabe kontrolliert, beabsichtigte Handlungen werden ausgeführt und ungewollte unterdrückt. (vgl. [Born09] S.281-S.289).

Zusammenfassend ist klar, dass es bei der Wahl der besten Therapieform unterschiedliche Ansichten gibt, nichtsdestotrotz wurde die Wirkung von computergestützten Therapieformen bewiesen, jedoch hat der behandelnde Arzt das letzte Wort.

2.3 JQuery

Ein Leben ohne Internet ist für viele Menschen heute geradezu unvorstellbar geworden. Es hat heutzutage fast niemand mehr keinen Internetzugang zu Hause oder unterwegs.

Der Erfolg hat häufig viele Väter, auch der des Internets. Als einer der wichtigsten Gründer des heute weltweiten Kommunikationsnetzes gilt der US-Psychologe J.C.R. Licklider, der sich Ende der 50er Jahre mit technischen Netzwerken beschäftigte. Licklider war Direktor eines Forschungsarms des Pentagons und wollte den Austausch mit Wissenschaftlern von US-Universitäten verbessern.

Die Weiterentwicklung zum World Wide Web gelang schließlich 1989 dem Briten Tim Berners-Lee. Er entwickelte am europäischen Kernforschungszentrum Cern die Seitenbeschreibungssprache HTML⁴ auf der noch heute die Strukturierung von Internetinhalten basiert. Im September 1995 wurde JavaScript von Brendon Eich erfunden.

JavaScript stellt bekanntermaßen neben (X)HTML und CSS⁵ die dritte Säule für die Erstellung von Webseiten dar und sorgt für die Dynamik und Interaktivität eines Webdokuments.

Diese Dynamik hat in den vergangenen Jahren eine immer größere Bedeutung gewonnen. Sie ist ein Teil der Erwartung des Nutzers geworden. Es gibt kaum noch Webseiten ohne

⁴ HypertextMarkup Language

⁵ Cascading Style Sheet

dynamisch gesteuerte Menüs, interaktive Oberflächen oder das Nachladen von Inhalten via Ajax.

Zur Realisierung dieser Dynamik sind viele Frameworks für JavaScripts zu Stande gekommen. Hierzu zählen das von Google zur Verfügung gestellte GWT, dessen Konkurrenz YUI, Dojo und jQuery sind.

Frameworks haben unter anderem die Aufgabe, Unterschiede zwischen Browsern zu überbrücken. Der damalige umständliche Umgang mit dem DOM wird durch Frameworks vereinheitlicht und erleichtert. Die DOM-orientierte Programmierung wird stark vereinfacht, bleibt aber sehr umständlich. Lange Funktionsnamen führen zu unleserlichen Codes.

Außerdem bestehen noch Inkompatibilitäten im Bereich Ajax. Infolgedessen wird für eine konsistente Entwicklung eine Abstraktionsschicht über dem DOM benötigt, die eine wirklich verwendbare und einheitliche API zur Verfügung stellt. Zudem sollte die Abstraktionsschicht funktionierende Iteratoren und eine nützliche Ajax-API bieten. Die erste Bibliothek, die diese Features bietet, ist jQuery. (vgl. [Matt10] S.19-21, [Ochs12] S.177 - 179).

jQuery wurde von John Resig implementiert. Die Knappheit des Codes, der nach Einbinden der Bibliothek für die Programmierer zu schreiben bleibt, macht jQuery bei Programmierern sehr beliebt und macht die Besonderheit von jQuery aus. Diese Knappheit wird von [Dem10] als „*Schreibe weniger, schaffe mehr*“ bezeichnet. Die Philosophie hinter dieser Bezeichnung beruht auf drei Konzepten:

- Elemente über CSS-Selektoren finden, mit denen dann etwas mit Hilfe von jQuery-Methoden getan wird
- mehrere jQuery-Methoden werden für ein Set aus verschiedenen Elementen verkettet
- den jQuery-Wrapper und die implizite Iteration nutzen.

Das fundamentale Konstrukt in jQuery ist die Funktion `$()`. Die Aufgabe dieser Funktion ist das Finden von DOM-Elementen, die durch `$` gezeichnet sind. Mit diesem Aufruf wird ein neues jQuery-Objekt erzeugt und zurückgegeben. Das Ergebnis dieses Aufrufs wird als Arbeitsgrundlage benutzt. Auf das selektierte Element können viele jQuery-Funktionen aufgerufen werden. Des Weiteren verfügt jQuery über ein eigenes Event Handling.

Ein sehr großer Vorteil von jQuery ist sicherlich auch noch seine Erweiterbarkeit. Dafür bietet jQuery eine Plug-in-Architektur, mithilfe derer nicht vorhandene Funktionalitäten mittels

Plug-ins ergänzt werden können. Ein solches Plug-in ist jQuery-UI. (vgl. [Matt10] S.19-42, [Ochs12] S.177 -181).

Zu den Aufgaben von Frameworks bzw. Bibliotheken zählt auch die Bereitstellung von Mitteln für die Webprogrammierer zur Erstellung dynamischer Webseiten, die modernen Benutzererwartungen entsprechen. Dieser Teil wird bei jQuery durch Bibliotheken aus jQuery-UI abgedeckt. Zu den modernen Usererwartungen gehören unter anderem Tabs, Datepicker, Progressbar und Slider. Außerdem bietet jQuery-UI noch subtilere Funktionalitäten zur reichhaltigen Gestaltung von Benutzeroberflächen. Allerdings können einzelne Elemente und Strukturen der Webseite per Drag & Drop verschoben oder auch deren Größe verändert werden. Solche Funktionalitäten werden Iterationen genannt. Sie sind in fünf Gruppen unterteilt: Draggable (Drag), Droppable (Drop), Resizable (Anpassen der Größe), Selectable (Auswählbar) und Sortable (Sortierbar).

Weiterhin gibt es auch Effekte zum Aus- und Einblenden von Elementen. (vgl. [Matt10] S.21, [Demm10] S.325, [Ochs12] S.183 -187).

Bibliotheken von JQuery- UI können von einer Seite heruntergeladen werden. Dadurch bekommt man UI-Elemente mit Standardeinstellungen, die sich je nach Bedarf individuell anpassen lassen. Dafür ist der sogenannte Themeroller da. Alle jQuery-UI-Widgets sind auf einer Seite zu finden. Auf der Seite kann man den Look eines Themes manipulieren und das Ergebnis kann als Stylesheet gespeichert werden. Es wird empfohlen, die Möglichkeiten des Themerollers auszunutzen und nur in dem Fall, dass sie unzureichend sind, sich eigene Stylesheets zu schreiben. (vgl. [Ochs12] S. 187).

3 Analyse und Konzeption

Im Rahmen dieser Arbeit wird eine Applikation entwickelt, die den Patienten bei der Wiedererlangung des Langzeitgedächtnisses unterstützt und das Ergebnis jeder Sitzung zur Kontrolle des Vorschlritts speichert. Im Folgenden wird zuerst das vorhandene System analysiert; darauf aufbauend werden die Anforderungen an die Applikation definiert. Zunächst wird die optimale Darstellungsform für die Applikation ausgewählt. Anschließend wird der Entwurf der grafischen Oberfläche erläutert.

3.1 Anforderungsanalyse

3.1.1 Analyse des jetzigen Programms

Als Einstieg in die Anforderungsanalyse wurde die alte Software „Gesichter“, und zwar die Module „Gesichter merken“ und „Wie merkt man sich Namen und Gesichter?“, analysiert.

Wie bereits im Kapitel 1.2 erwähnt, wurde die in dieser Arbeit betrachtete Version der Software „Gesichter“ (Version 2.0) von Prof. Dr. Erich Kasten im Jahre 2006 mit der Programmiersprache VB5 (Visual Basic Version 5) entwickelt. VB ist eine

Programmiersprache von Microsoft, die nur zur Entwicklung von Windows Anwendungen geeignet ist. VB5 wurde im Jahre 1998 herausgegeben, der Sprache wurden objektorientierte Elemente hinzugefügt, jedoch ohne alle Kriterien der objektorientierten Programmierung.

Als erstes wurde die vorhandene Programmbeschreibung betrachtet. Sie besteht hauptsächlich aus Systemvoraussetzungen, einer Installationseinleitung und einer detaillierten Beschreibung der einzelnen Module. Unter den Systemvoraussetzungen wird zu Beginn verdeutlicht, dass das Programm nur unter Windows Betriebssystemen lauffähig ist, was man auch an der benutzten Programmiersprache sofort erkennen könnte.

Zur weiteren Programmuntersuchung wurde im nächsten Schritt das eigentliche Programm analysiert. Allerdings wurde an erster Stelle die graphische Benutzeroberfläche betrachtet, welche in den folgenden Abbildungen dargestellt wird.

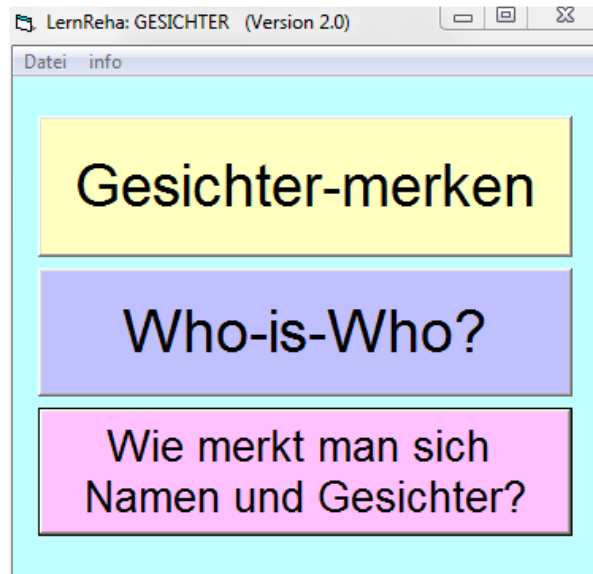


Abbildung 4- Einstieg in die Module

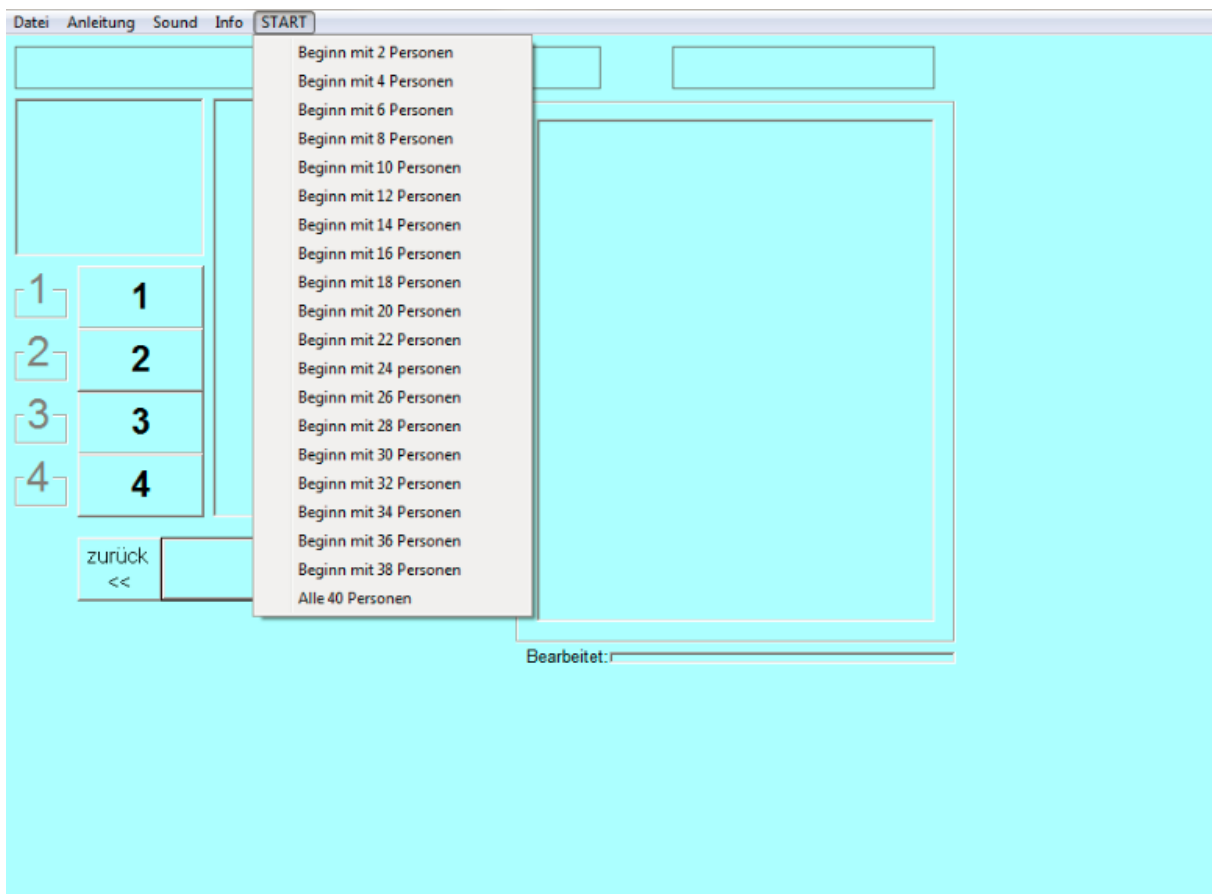


Abbildung 5- Modul "Gesichter merken" Anzahl Personen auswählen

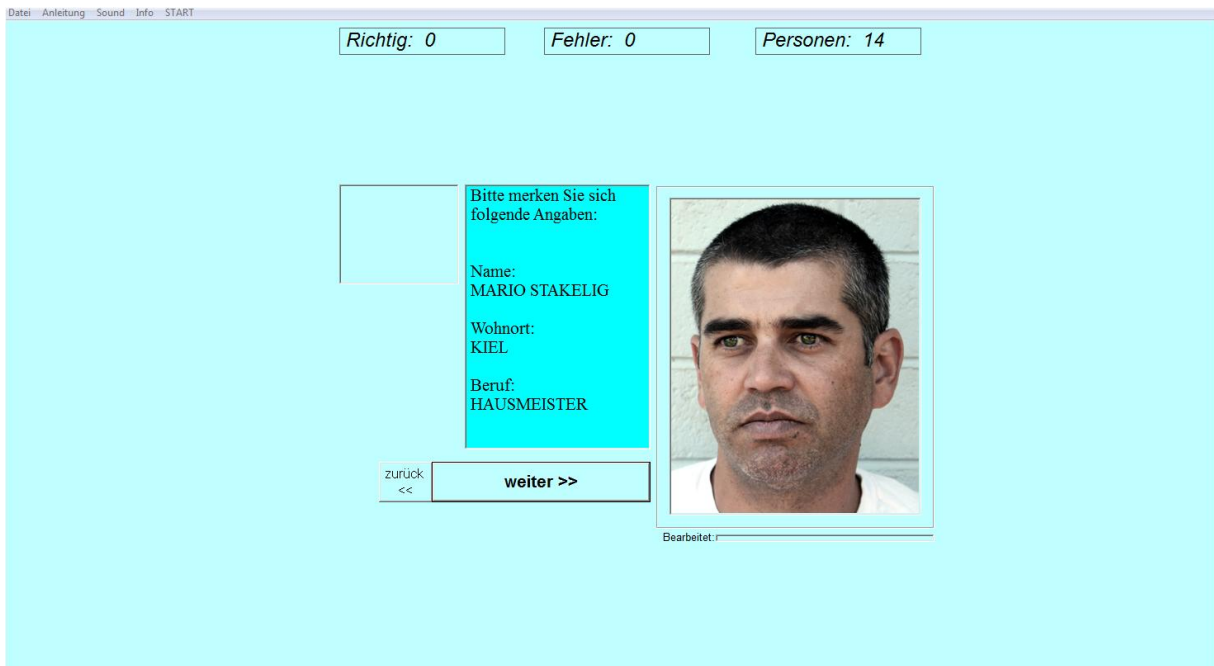


Abbildung 6- Modul "Gesichter merken", Spielrunde

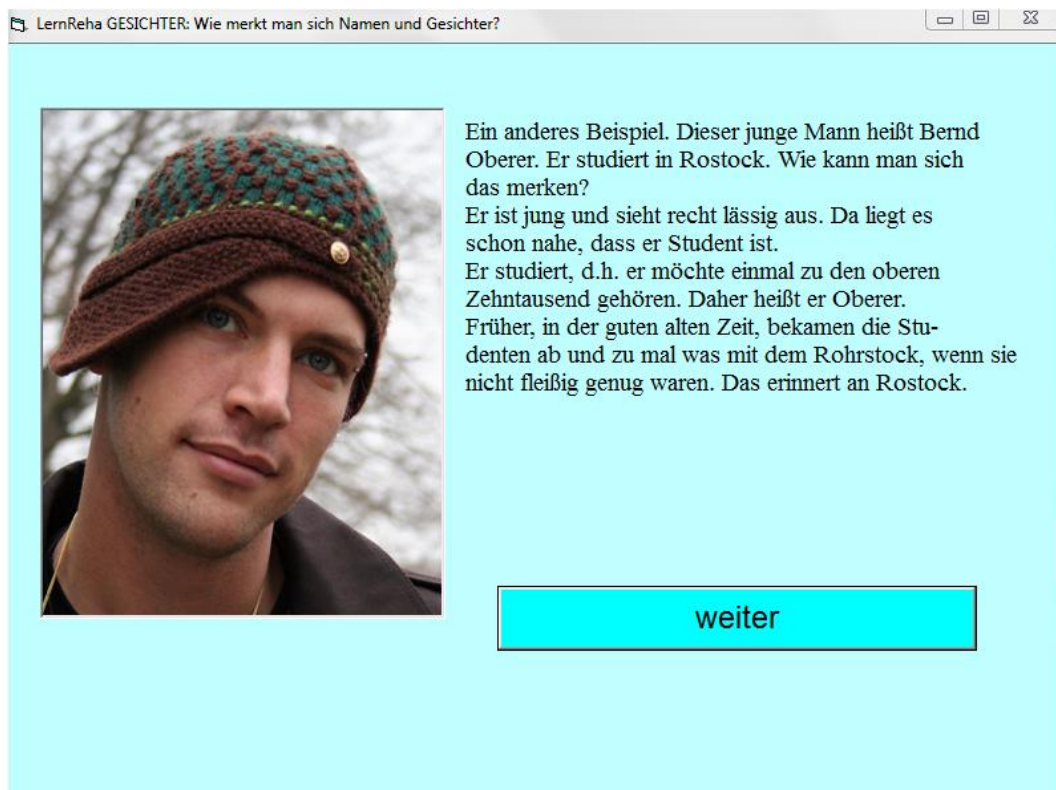


Abbildung 7- Modul "Wie merkt man sich Namen und Gesichter?"

Die Abbildung 4 ist der Einstiegspunkt in die jeweiligen Module. Durch das Klicken, zum Beispiel auf die Schaltfläche(engl. Button) "Gesichter-merken", gelangt man auf das Modul "Gesichter merken". Die folgende Abbildung (Abbildung 5) stellt das Modul „Gesichter merken“ dar, welche eine Oberfläche mit Buttons zeigt, auf denen Nummern und Überschriften stehen, womit man erst einmal nichts anfangen kann, da man an diesem Punkt noch nicht mal die Anzahl der Personen ausgewählt hat. Außerdem bekommt man beim Anklicken auf den Menü-Punkt „Start“ eine lange Liste mit der Anzahl von Personen, welche fast die ganze Oberfläche bedeckt - dies weist auf einen Mangel von Design und Softwareergonomie hin. Nach der Auswahl der Personenbilder beginnt das eigentliche Spiel. Man bekommt eine Oberfläche zu sehen, an der es nicht viel zu beanstanden gibt.

Als nächstes wurde die Oberfläche des Moduls „Wie merkt man sich Namen und Gesichter?“ untersucht. Die Oberfläche (Siehe Abbildung 7) ist einfach und verständlich, aber wenn man sich durch sie bewegt, merkt man sofort, dass es keine Möglichkeit gibt, auf die vorherigen Tipps zurückzukommen, was eine sehr große Einschränkung für den Benutzer darstellt und auch die unzulängliche Softwareergonomie bezeugt.

Aus dem Durchspielen der Software und dem Mangel der objektorientierten Programmiersprache wurde zur Veranschaulichung des Ablaufs der beiden Module, die in dieser Arbeit in Betracht gezogen werden, das folgende UML Aktivitätsdiagramm erstellt. Das Diagramm zeigt die Einzelschritte zum Auswahl eines Moduls bis zur Absolvierung des Trainings bzw. bis zum Durchlesen aller Ratschläge zum Gesichter Merken.

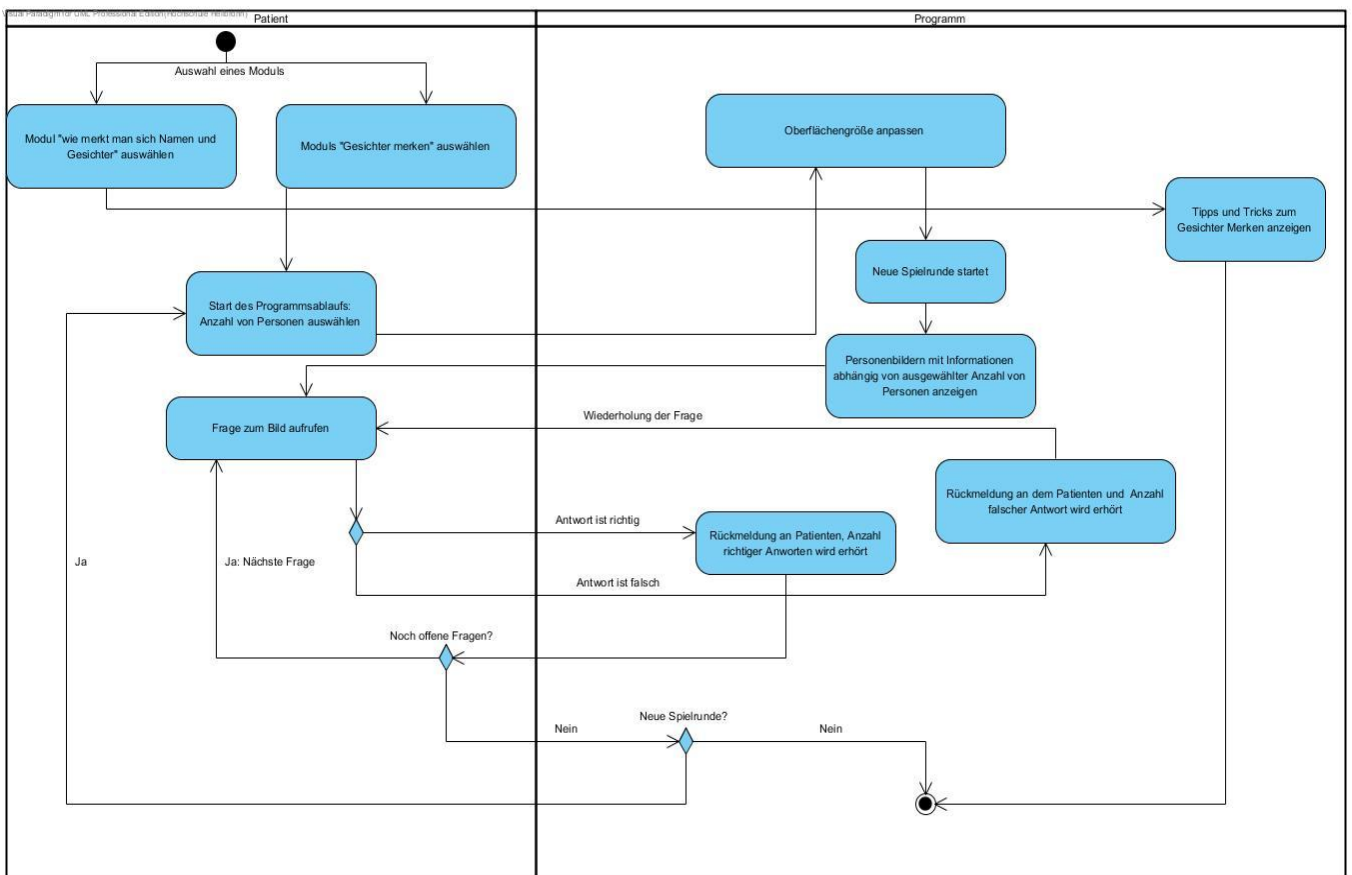


Abbildung 8- Aktivitätsdiagramm: Darstellung des Kernablaufs der Module "Gesichter-merken" und "Wie merkt man sich Namen und Gesichter?"

Nach dem Erproben der Benutzeroberfläche wurde die Programmlogik, und zwar der Quellcode, unter die Lupe genommen. Aus dieser Untersuchung hat sich eine grobe Systemarchitektur ergeben, welche mit der folgenden Skizze veranschaulicht wird(Siehe Abbildung 9).

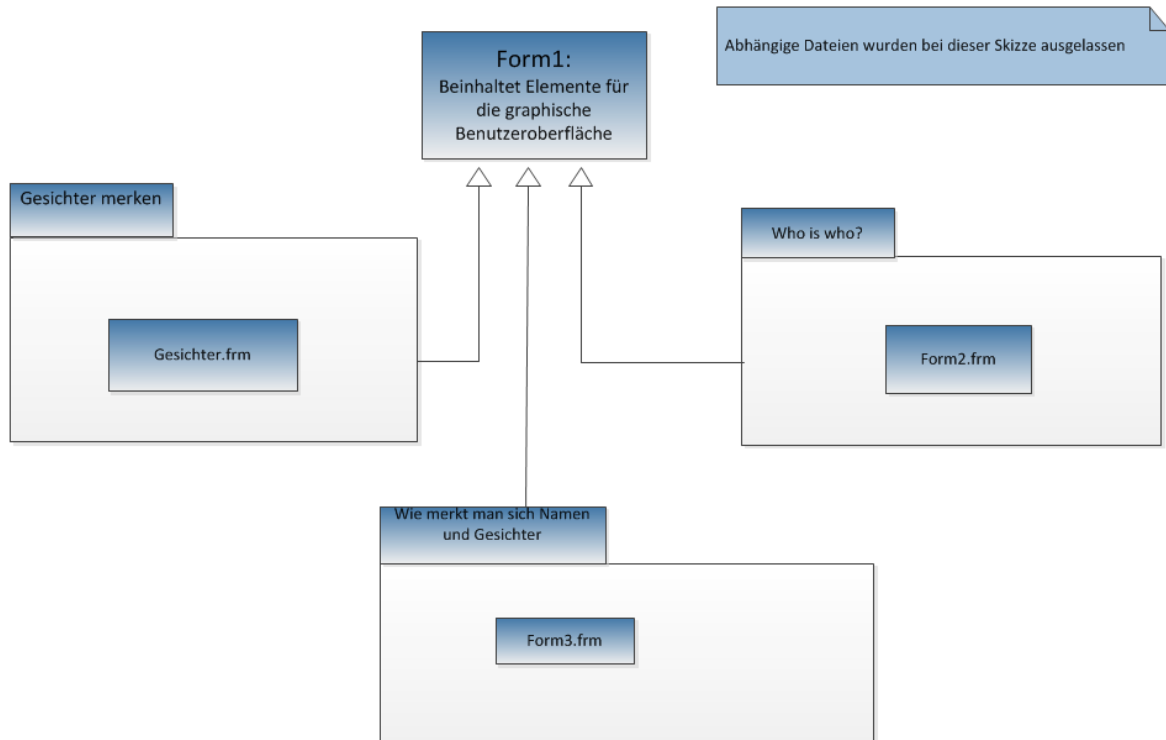


Abbildung 9- Systemarchitektur des alten Programs

Bemerkenswert ist, dass jedes Modul in einer einzigen „Klasse“ bzw. Form realisiert wurde, welche sowohl die graphische Benutzeroberfläche (in Form1) als auch die Logik des Programms beinhaltet, was schon einen großen Mangel an Objektorientierung zeigt bzw. die Wiederverwendung stark einschränkt.

Des Weiteren weist die Namensgebung der einzelnen „Klassen“ (Form2, Form3) eine stark fehlende Aussagekraft auf. Auch der Name des Formulars (Form1), welches die Elemente der graphischen Oberfläche beinhaltet, ist nicht viel sagend und deutet auf ein unzureichendes Design hin, wie schon vorher festgestellt wurde.

Zum Abschluss der Analyse des alten Programms wurde der Code tiefer untersucht und unter anderem eine sehr hohe Statik festgestellt. Allerdings werden alle Elemente der graphischen Ebene am Anfang des Programms statisch erstellt (in Form1) und nur bei Bedarf ein- oder ausgeblendet. Das gleiche wird in jeder Funktion gemacht - dies ist kein gutes Vorbild für das Ressourcen-Management in der Softwareentwicklung (nimmt viele Ressourcen in Anspruch). Der folgende Auszug aus dem Modul „Gesichter merken“ soll die verdeutlichen (siehe Abbildung 10).

```

Sub abfragauswahl()
Command5.Caption = "": Command5.Enabled = False: Command5.Visible = False
Command6.Caption = "": Command6.Enabled = False: Command6.Visible = False
Command1.Visible = True
Command2.Visible = True
Command3.Visible = True
Command4.Visible = True

Frame1.Visible = True
Frame2.Visible = True
Frame3.Visible = True
Frame4.Visible = True

```

Abbildung 10- Auszug aus dem Modul "Gesichter merken"

Zusammenfassend wurde das Programm bzw. die Module „Gesichter merken“ und „Wie merkt man sich Namen und Gesichter?“ von der graphischen Oberfläche bis zum Code untersucht und daraus hat sich ergeben, dass der Programmcode einen großen Mangel an Wiederverwendung zeigt und gar keine Objektorientierung beinhaltet. Zudem weist das Programm Designschwächen und massive Codeduplizierung auf. Die Verbesserung des Codes würde noch in Frage kommen, wenn die Software nicht nur auf Windows Betriebssystemen lauffähig wäre. Diese Systemeinschränkung stellt aber einen sehr großen Nachteil für die Software dar. Außerdem ist das Programm eine Desktop-Anwendung, d.h. sie kann nur von dem Computer, auf dem sie installiert ist, benutzt werden, was die Mobilität sehr einschränkt.

Daraus resultiert die Unumgänglichkeit einer Neuimplementierung.

Die Anforderungen an die neue Software werden im nächsten Kapitel verdeutlicht.

3.1.2 Anforderung an die neue Applikation

Als erstes wurde die Benutzergruppe betrachtet. Hierbei geht es grundsätzlich um das ärztliche Personal, das für die Betreuung der Patienten zuständig ist. Die Hauptbenutzer sind aber die hirngeschädigten Patienten, die die Applikation auch von zu Hause aus benutzen sollen. Dementsprechend soll die Applikation so einfach und intuitiv wie möglich gehalten werden. Zudem sollte die Installation so einfach wie möglich sein.

Ein weiteres Bestreben ist es, die Applikation mobil zu machen. Allerdings sollte man auf die neue Software von überall aus zugreifen können. Hierfür stehen viele Technologien zur Wahl, welche im nächsten Kapitel vorgestellt werden.

Zu einem gelungenen Training gehört sicherlich auch eine Fortschrittsanzeige. Um dies zu ermöglichen, soll das Programm so interaktiv wie nur möglich sein. In der Tat soll die neue Anwendung über die Möglichkeit verfügen, dem Anwender eine Rückmeldung bei einer Antwort zu geben. Diese soll sowohl akustisch als auch visuell erfolgen. Hierbei soll die akustische Wiedergabe natürlich auch unterdrückt werden können. Ein weiterer wichtiger Punkt zur Verfolgung des Fortschritts und guten Betreuung der Patienten ist die Möglichkeit, Ergebnisse der einzelnen Spielrunden speichern und ausdrucken zu können.

Schließlich beruhen die Funktionalitäten der neuen Applikation auf denen der alten. Die Funktionalitäten der beiden Module werden in dem folgenden Use-Case-Diagramm zusammengefasst.

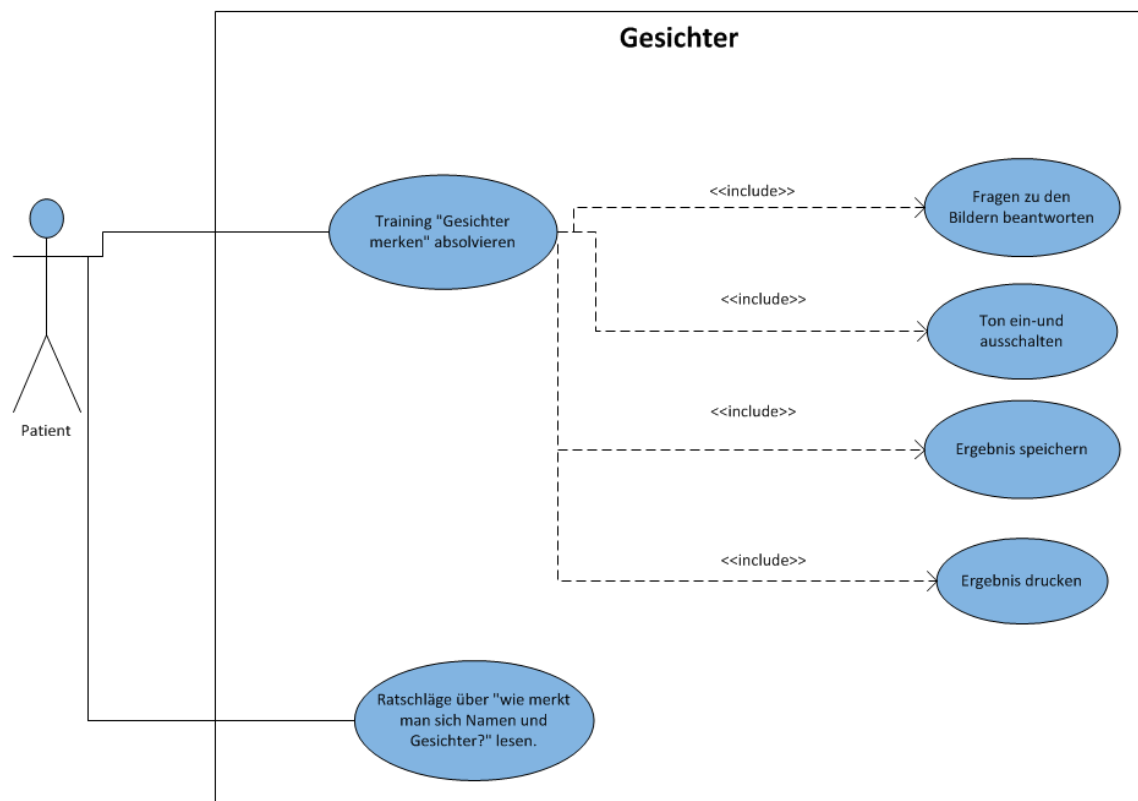


Abbildung 11- Use-Case-Diagramm: Funktionalitäten der neuen Software

Im Folgenden werden die Abläufe der Use-Cases im Detail näher erläutert.

Use Case Name: Training "Gesichter merken" absolvieren

Beschreibung: „Gesichter merken“ ist ein rundenbasiertes Training. Nach dem Anzeigen von Personenbildern mit den dazugehörigen Informationen ist der Patient angehalten, Fragen zu den Bildern zu beantworten.

Subsystem: Gesichter

Akteur: Patient

Normal-Ablauf:

1. Der Patient wählt die Anzahl der unterschiedlichen Personenbilder pro Spielrunde aus.
2. Das System zeigt dem Patienten die Personenbilder mit den jeweiligen Informationen an und der Patient soll sich diese merken.
3. Dem Patienten werden Fragen zu den gezeigten Bildern gestellt, die er zu beantworten hat.
4. Wiederholung der Schritte 1 und 2 in beliebiger Häufigkeit.

Normalergebnis:

Der Patient hat eine Spielrunde abgeschlossen und die Statistiken liegen vor.

Notizen:

1. Der Patient hat die Möglichkeit, sich die Anleitung durchzulesen.
2. Der Patient hat die Möglichkeit, die akustische Wiedergabe ein- und auszuschalten.
3. Der Patient hat die Möglichkeit, die Ergebnisse einer Spielrunde zu speichern und auszudrucken.

Use Case Name: Ratschläge über „Wie merkt man sich Namen und Gesichter?“ lesen.

Beschreibung: Anzeigen von Tipps, mithilfe derer man sich Namen und Gesichter am besten merken kann.

Subsystem: Gesichter

Akteuren: Patient

Normal-Ablauf:

1. Der Patient liest sich die Ratschläge durch.

Normalergebnis:

Der Benutzer hat alle Ratschläge gelesen.

3.1.2.1 Qualitätszielbestimmungen

Großer Wert wird bei dem zu entwickelnden System auf die Benutzerfreundlichkeit und die Korrektheit gelegt. Die vom Patienten gemachten Eingaben sollen auf Korrektheit geprüft werden. Von daher ist es wichtig, dass die Software stabil ist, damit die Ergebnisse mit hoher Genauigkeit bzw. Korrektheit berechnet werden. Außerdem ist die Verfügbarkeit von großer Wichtigkeit. Es ist auch wichtig, dass ein Benutzer jederzeit Zugriff auf das System haben kann, damit eine Trainingsstunde rechtzeitig durchgeführt wird.

Ein Mangel, der zur Verzögerung eines Trainingsfortschritts führt, kann nicht toleriert werden.

Die Benutzerfreundlichkeit sowie die Einfachheit der Benutzeroberfläche sind Anforderungen mit hoher Priorität. Es ist vorauszusetzen, dass jede Trainingssitzung reibungslos abläuft. Aus diesem Grund muss das System hochgradig auf die Benutzerfreundlichkeit optimiert sein.

Die zu erwartenden Zugriffszahlen von Nutzern und Verwaltern auf das System sind hier nicht zu berücksichtigen, da die entwickelte Applikation größtenteils Client-seitig laufen sollte.

Das System muss auch nicht besonders robust gegenüber Angriffen von außen und innen sein. Manipulationen am System, die z.B. die Korrektheit der Eingaben gefährden könnten, werden wegen der geringen Größe der Applikation nicht beachtet.

Die folgende Tabelle zeigt die oben aufgezählten sowie andere qualitative Vereinbarungen bezüglich der Anforderungen an das Softwareprodukt.

	Sehr wichtig	wichtig	Weniger wichtig	unwichtig
Verfügbarkeit	x			
Benutzerfreundlichkeit	x			
Wartbarkeit			x	
Stabilität		x		
Robustheit				x
Korrektheit	x			

Tabelle 1- Qualitätszielbestimmungen

3.2 Entscheidung für das neue System

Im Abschnitt 3.1.2 wurden die Anforderungen an die neue Software vorgestellt. Darauf aufbauend werden in diesem Abschnitt Systeme analysiert und ein passendes ausgewählt.

Das entscheidende Kriterium für die Neuentwicklung war die Plattformunabhängigkeit. Die neue Applikation sollte auf so vielen Betriebssystemen wie möglich laufen können. Darausfolgend werden .net-Anwendungen (u.a. C-Sharp, VB) ausgeschlossen, da sie Windows spezifisch sind.

Als erste Überlegung wurde der Gebrauch eines mobilen Geräts in Betracht gezogen, um die Mobilität der neuen Applikation zu ermöglichen. Unter diese Kategorie fallen zahlreiche Gerätetypen wie zum Beispiel das Smartphone, das Netbook und der Tablet-PC.

Aufgrund der Größe des Displays und der Benutzergruppe (Hirngeschädigte) wurden zwei Gerätetypen berücksichtigt, und zwar das Netbook und der Tablet-PC. Beide sind gut für den mobilen Gebrauch geeignet, was eine sehr wichtige Rolle für das hier entwickelte System spielt. Netbooks sind „Mini-Notebooks“, die quasi alles bieten, was ein Laptop kann. Tablet-PCs sind ebenso Mini-Computer, die sich allerdings durch ihr Touchscreen-Display auszeichnen. Es bleibt aber zu berücksichtigen, dass diese mobilen Geräte unter unterschiedlichen Betriebssystemen laufen können, u.a. Windows, iOS von Apple und Android von Google. Eine Applikation zu entwickeln, die auf allen dieser Betriebssysteme laufen würde, wäre nicht bzw. nur sehr schwer realisierbar. Diese Option hätte zwar das

Problem der Mobilität gelöst, aber nicht des Installationsaufwands und der Plattformunabhängigkeit. Sie wird daher ausgeschlossen.

Die zweite Überlegung, das Ziel der Mobilität und die Plattformunabhängigkeit weiter verfolgend, ist der Einsatz von Webanwendungen. Eine Webanwendung, oder kurz Web-App, ist ein Anwendungsprogramm, das beim Benutzer in einem Webbrowser dargestellt wird. Der Zugriff auf eine Web-App erfolgt über das Web, nicht zwangsläufig über das Internet oder das Intranet, sondern über ein Übergangsprotokoll wie http. Eine Web-App erfordert im Gegensatz zu Desktopanwendungen kein spezielles Betriebssystem. Die Benutzer hätten die Möglichkeit, überall auf das Programm Zugriff zu haben, völlig unabhängig von dem Betriebssystem.

Im nächsten Abschnitt wird die passende Programmiersprache ausgewählt.

3.2.1 Programmiersprache

Zur Erstellung von Webseiten mit dynamischem Inhalt eignen sich verschiedene Techniken. Im Folgenden wird die einfachste Programmiersprache, gemessen an den formulierten Anforderungen, ausgewählt.

In Anbetracht der kleinen Größe der Applikation und derer geringeren Komplexität, wird auf die Einbindung zu einem Datenbanksystem verzichtet. Außerdem sollte keine Installation von zusätzlichen Laufzeitumgebungen nötig sein, außer den, die schon auf dem Zielsystem (u.a. Browser) vorhanden sind. Daraus ergibt sich das Ausschließen von Java (u.a. JSP, JSF, GWT) als Programmiersprache, da die Java Runtime Environment(JRE) auf jeden Fall herunterzuladen und zu installieren wäre, um die Applikation zum Laufen zu bringen.

Die gewählte Programmiersprache sollte so anpassungsfähig wie möglich sein, um Releases zu ermöglichen. Außerdem sollte sie schnell sein.

Die Programmiersprache, die diese Ansprüche größtenteils erfüllt, ist JavaScript.

JavaScript ist eine der einfachsten, anpassungsfähigsten Scripting-Sprachen, die verwendet werden kann, um die Funktionalität von Webseiten zu gestalten. Sie ist für die Erstellung visueller Effekte auf dem Bildschirm, Berechnen von Daten und Durchführen von Prozessen auf Webseiten sehr geeignet. Weiterhin kann JavaScript auf der Client-Seite oder vom Prozessor des Benutzers verwendet werden, dadurch wird die Verarbeitung von Prozessen sofort erledigt und ist von daher im Vergleich zu anderen Programmiersprachen schneller.

Dennoch muss das System auf einen Server hinterlegt werden, um einige Dinge, wie z.B. das Speichern, ausführen zu können, da dies JavaScript aus Sicherheitsgründen selber nicht unterstützt.

3.2.2 Konzeption: Architektur und Entwurf der grafischen Benutzeroberfläche

Die Anwendung „Gesichter“ muss immer mit dem Benutzer interagieren. Darunter wird die Tatsache verstanden, dass er immer weiß, was das System tut. Die erste Ansicht, die der Anwender sehen wird, ist das Hauptmenü. Hier werden alle Funktionalitäten der Applikation zusammengefasst. Dieses Menü wird möglichst selbsterklärend dargestellt. Es werden drei *Usability*-Kriterien in den Vordergrund gestellt.

Die ersten zwei Kriterien sind das *Ease of learning* und das *Ease of remembering*. Dies bedeutet, dass der Benutzer sich bei der ersten Benutzung ohne besondere Einführung zurechtfinden können soll. Bei den späteren Anwendungen sollte er sich möglichst gut an die Benutzung erinnern können. Das dritte Kriterium ist die Antwort des Systems auf die Benutzeranfrage. Sie soll schnell und fehlerfrei erfolgen. Hinter jedem Befehlssatz sollte genau die Vorstellung des Benutzers verborgen sein. In der Tat, eine gute *Usability* erhöht die Akzeptanz der Anwendung.

Eines der Hauptziele von „Gesichter“ ist das Feedback auf die Antwort des Patienten. Es soll so wiedergegeben werden, dass der Benutzer es sofort versteht. Dieses Feedback soll „live“ stattfinden. Eine akustische und visuelle Rückmeldung stellen dafür eine gute Möglichkeit dar.

Zur Erstellung einer JavaScript basierten Web-Anwendung sind unterschiedliche Dateitypen nötig, nämlich html-, .js⁶- und .css- Dateien. In .html-Dateien werden die Webseite selbst sowie andere Informationen, die in einem Web-Browser angezeigt werden können, definiert. .css- Dateien definieren, wie html-Elemente anzuzeigen sind (u.a. Größe, Breite und Lage eines Elements). Zur dynamischen Darstellung von Inhalten sind .js-Dateien wichtig. Sie verfügen über Funktionen zum Ändern, Löschen und Einfügen von Elementen und ermöglichen die Interaktion mit dem Benutzer der Anwendung.

Zum Entwurf einer graphischen Oberfläche, die die oben genannten Anforderungen erfüllt, wurde die folgende Systemarchitektur erstellt.

⁶ *js: JavaScript*

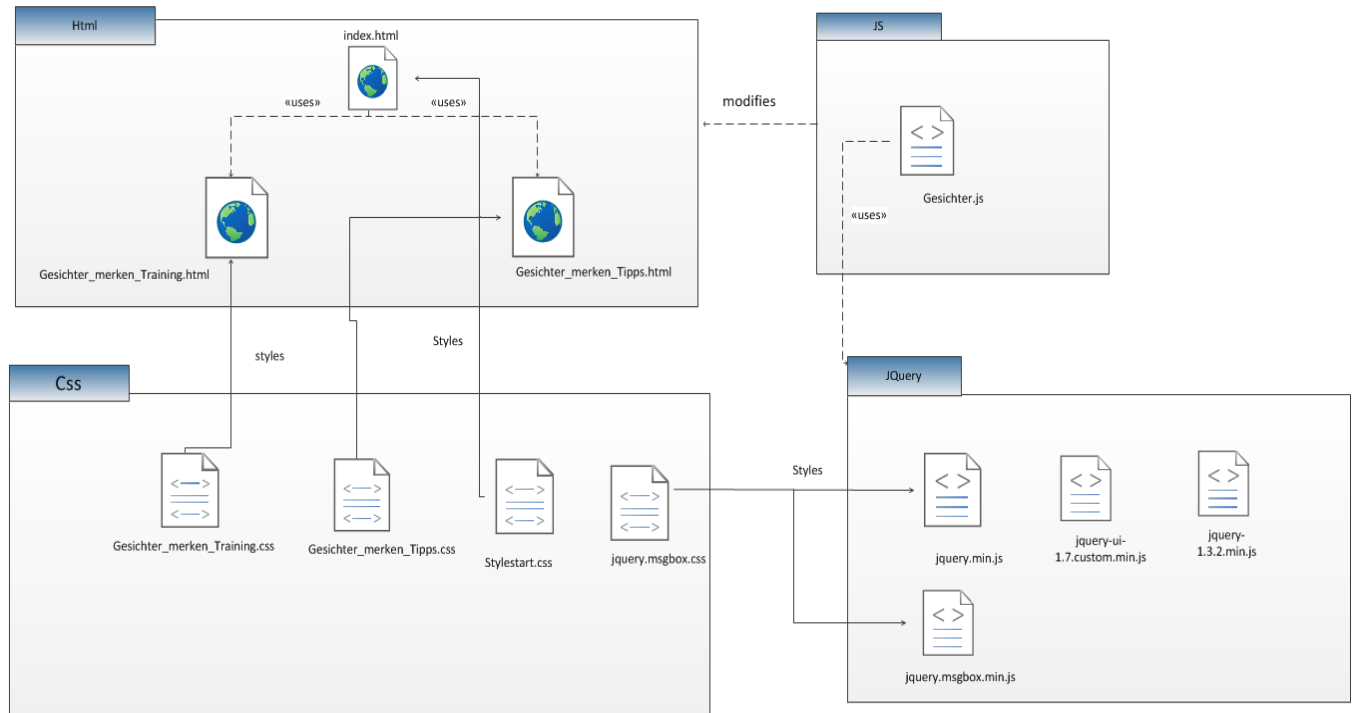


Abbildung 12- Systemarchitektur der neuen Anwendung

Neben der eigentlichen Programmkomponente sind auf dem Schaubild die verwendeten Bibliotheken JQuery und JQuery UI sowie das zugehörige Stylesheet zu sehen.

In allen `.html`-Dateien werden statische Elemente (u.a. Menüs, Textfields, Tabellen) erzeugt und diese werden in `.js`-Dateien abgeändert.

`index.html` stellt die Startseite der Applikation dar. Auf der Seite werden die zwei Module „Gesichter merken“ und „Wie merkt man sich Namen und Gesichter?“ vorgestellt.

Die statische Oberfläche des Moduls „Wie merkt man sich Namen und Gesichter?“ wird in `Gesichter_merken_Tipps.html` implementiert. Die Gestaltung geschieht aber in dem Stylesheet `Gesichter_merken_Tipps.css`. Dort werden alle graphischen Elemente positioniert und ihnen wird eine Größe und Farbe gegeben.

Die eigentliche Logik der Anwendung liegt in der Datei `Gesichter.js`. Sie ist für die Steuerung der Programmabläufe zuständig. Allerdings wird dort auch die Prüfung der Eingaben des Anwenders auf die Richtigkeit sowie die passende Rückmeldung implementiert. Die Rückmeldung erfolgt sowohl akustisch als auch visuell (Anzeigen der Anzahl von richtigen

und falschen Eingaben sowie die dazu gehörigen Smileys). Sie ist auch für das Speichern von Daten verantwortlich. Zudem ist die Datei *Gesichter.js* auch für die Steuerung der Elemente des Moduls „Wie merkt man sich Namen und Gesichter?“ zuständig.

Im nächsten Kapitel wird gezeigt, wie diese Konzeption ins Leben gerufen wurde und welche Probleme dabei aufgetreten sind.

4 Implementierung

Dieses Kapitel beschreibt, wie die Anforderungen an die Applikation realisiert wurden. Dabei wird viel Wert auf die Schwierigkeiten gelegt, die während der Implementierung beobachtet wurden. Es wird gezeigt, wie mit ihnen umgegangen wurde bzw. wie sie gelöst wurden.

Als Einstieg in die Implementierung müssen ein paar Konzepte erklärt werden und ein paar Systemanforderungen festgelegt werden.

Zum Ausführen von JavaScript-Anwendungen ist meistens ein JavaScript-Interpreter (u.a. Chrome benutzt V8, IE benutzt JScript, Firefox benutzt SpiderMonkey) in dem Browser des Nutzers integriert. Ein JavaScript-Interpreter hat die Aufgabe, in der Laufzeit den Programmcode einzulesen, zu analysieren und auszuführen. Da jeder Browser einen unterschiedlichen Interpreter hat, kann es zu Inkonsistenzen bei der Interpretierung des Codes kommen, daher müssen die Browser-Versionen für die Anwendung „Gesichter“ festgehalten werden. Allerdings läuft die Applikation unter Firefox 26, Google Chrome 35 und IE⁷ 11.

Im Folgenden wird ein Blick auf die Implementierung der im Abschnitt 3.1.2 formulierten Anforderungen geworfen.

Wie im Abschnitt 3.2.2 beschrieben, stellt index.html die Homepage der Applikation dar. Die folgende Abbildung zeigt die Realisierung.

⁷ Internet Explorer

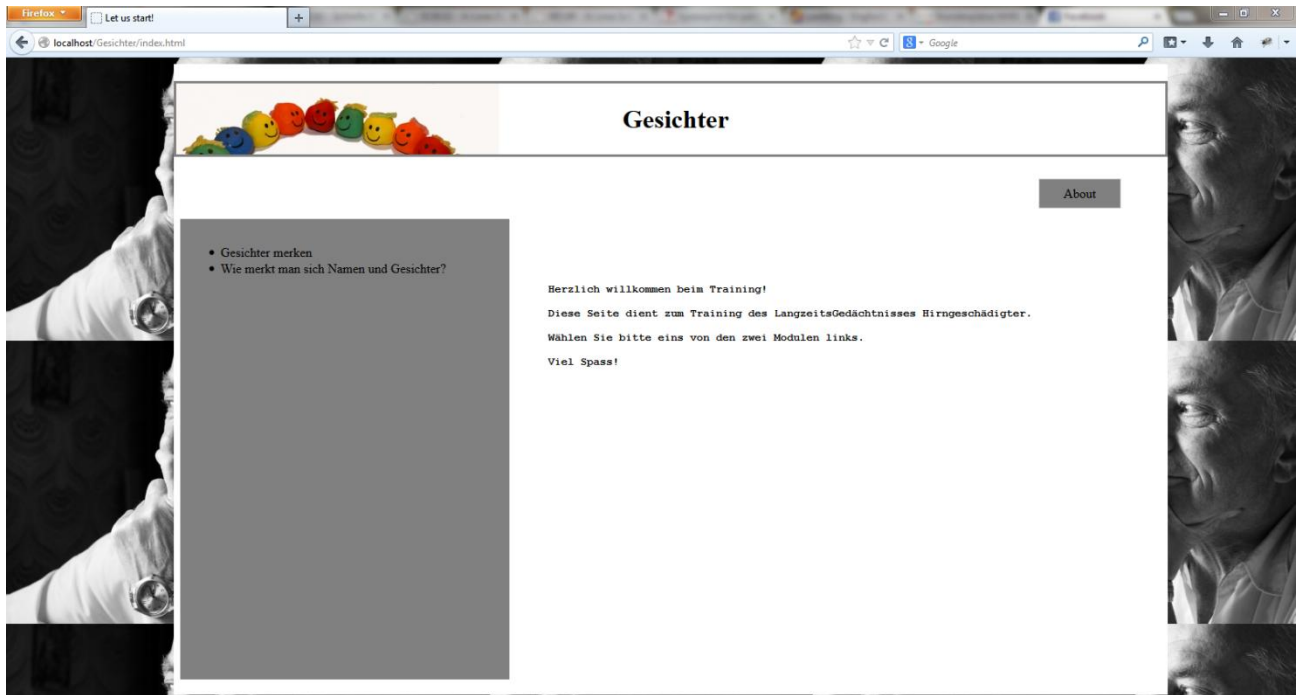


Abbildung 13- Startseite der neuen Applikation

Auf der Seite rechts ist der Button „About“ zu sehen. Er gibt dem Benutzer Auskunft über die Applikation und beschreibt diese. Links sind die zwei Module aufgelistet und mit einem Klick gelangt man zu dem jeweiligen Modul .

Die folgende Abbildung(Abbildung 14) zeigt die Oberfläche des Moduls „Wie merkt man sich Namen und Gesichter?“

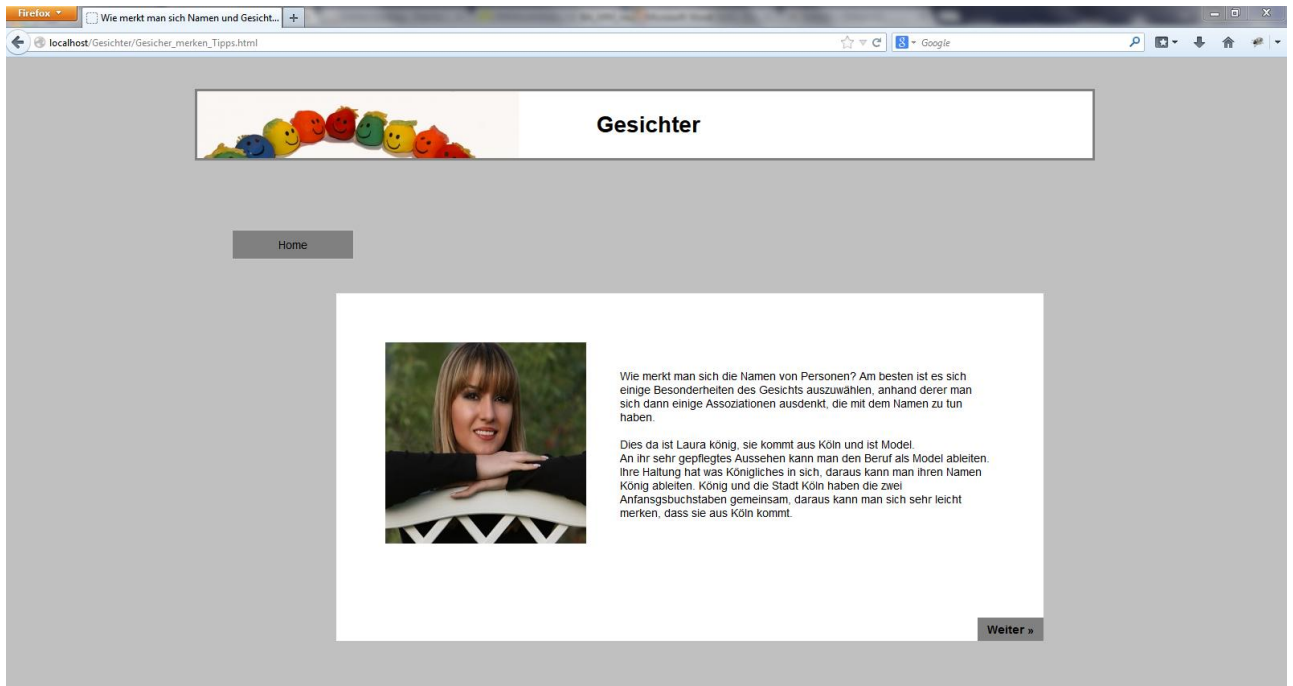


Abbildung 14- Benutzeroberfläche des Moduls „Wie merkt man sich Namen und Gesichter?“

Als nächstes wird ein Blick auf das Modul „Gesichter merken“ geworfen. Die Abbildung Abbildung 15 zeigt die Oberfläche des Moduls an.

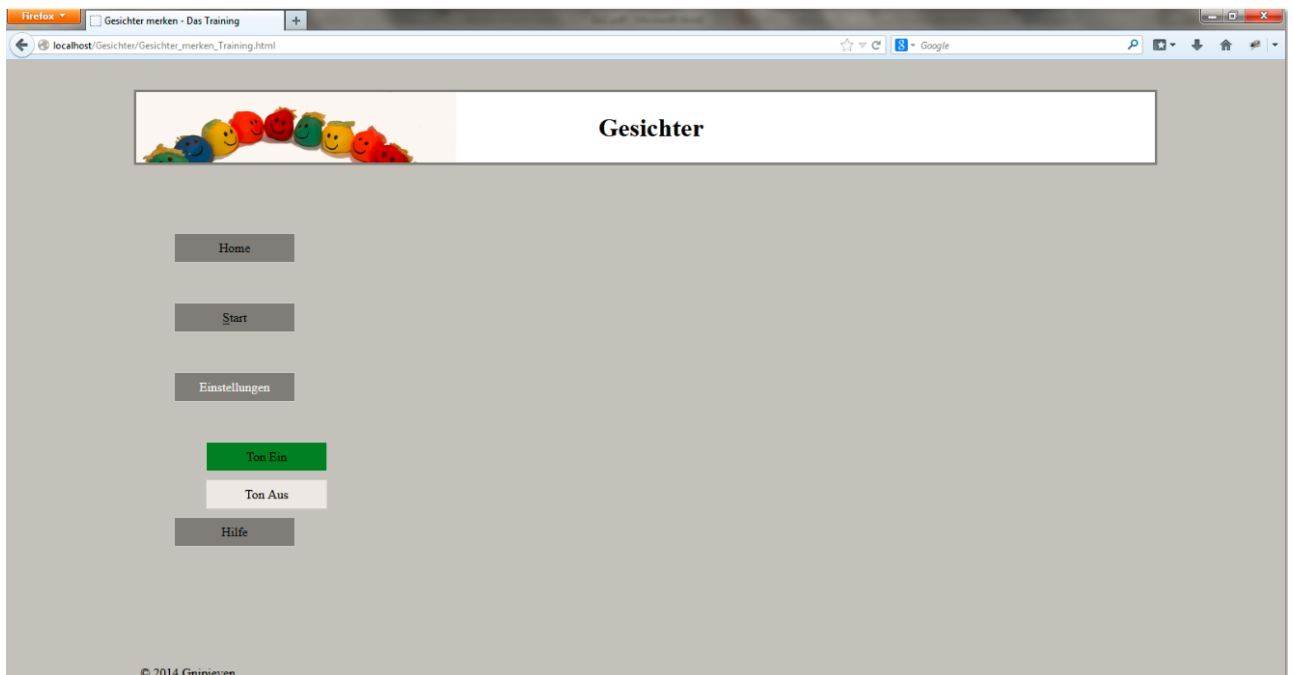


Abbildung 15- Benutzeroberfläche des Moduls „Gesichter merken“.

Es fällt sofort auf, dass der Menü-Punkt „Start“ einen Unterstrich unter dem ersten Buchstabe hat. Dieser dient zum Benutzen der Tastatur zur Steuerung des Spiels. Im nächsten Abschnitt wird erklärt, wie es realisiert wurde.

4.1 Shortcuts

Eine der wichtigsten Anforderungen an die neue Applikation war die Möglichkeit, das Spiel mithilfe der Tastatur zu steuern. Bevor wir in den Code hineinschauen, müssen zunächst ein paar Konstrukte erklärt werden. Als Einstieg werfen wir erst einmal einen Blick auf die deutsche Tastatur (QWERTZ), veranschaulicht in Abbildung 16. Es stellt sich die Frage zu wissen, was passiert, wenn man auf eine Tastaturtaste drückt.

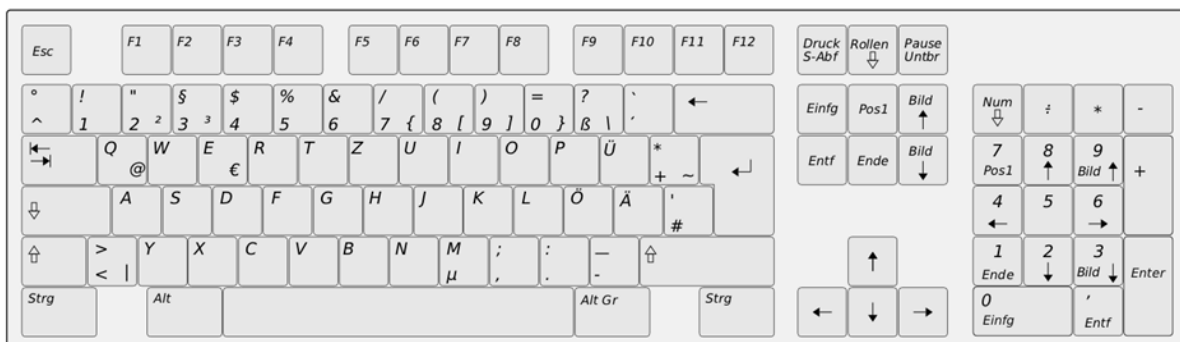


Abbildung 16- Deutsche Tastatur

Das einfache Drücken und Loslassen einer Taste auf der Tastatur erzeugt drei Ereignisse in dieser Reihenfolge: *onkeydown*, *onkeypress* und *onkeyup*.

onkeydown und *onkeypress* feuern kontinuierlich, solange der Benutzer die Taste gedrückt hält. Wenn der Benutzer die Taste loslässt, feuert *onkeyup* ein einziges Mal.

Onkeydown ruft einen Event Handler auf, wenn der Benutzer eine Taste auf der Tastatur herunterdrückt. *Onkeyup* ruft einen Event Handler auf, wenn der Benutzer eine Taste auf der Tastatur losgelassen hat. *Onkeypress* ruft einen Event Handler auf, wenn der Benutzer eine Taste auf der Tastatur herunterdrückt oder festhält.

In der DOM Level 2-Empfehlung sind keine Tastatur-Ereignisse zu finden. In der Unicode-Welt sind Tastatur-Ereignisse noch zu kompliziert, um sie in einen generellen Standard einzubinden. Aber Tastatur-Ereignisse sind fundamental, in allen modernen Browsern

implementiert und der HTML 4-Standard erlaubt *keydown*, *keypress* und *keyup*-Ereignisse für viele Elemente.

Bei einem Tastatur-Ereignis wird ein numerischer Code für die jeweilige Taste erzeugt. Für die Taste „A“ ist es z.B. die Zahl 65. Dieselbe Taste erzeugt mindestens zwei Zeichen, nämlich „A“ und „a“, je nachdem, ob die Shift-Taste gedrückt wurde oder nicht. Auf einem Mac erzeugt die A-Taste noch weitere Zeichen, wenn die Alt-Taste zusätzlich gedrückt wird. Für die Implementierung von „Gesichter“ wurde *keypress* benutzt, wie auf dem folgenden Codeauszug zu sehen ist.

```
/*Shortcuts ON KEYPRESS*/
window.addEventListener('keypress', function (e) {

    if(e.charCode == 102 || e.charCode == 70 ) //f or F, for "Frage stellen"
        $("#AskQuestion").trigger('click');
    else if(e.charCode == 103 || e.charCode == 71 ) // g or G, for "zu den Fragen"
        $("#GoToQuestion").trigger('click');
    else if(e.charCode == 115 || e.charCode == 83 ) // s or S, for "start"
        $("#start").trigger('click');
    else if(e.charCode == 119 || e.charCode == 87 ) // w or W, for "weiter"
        $("#next").trigger('click');
    else if(e.charCode == 122 || e.charCode == 90 ) // z or Z , for "zurück"
        $("#prev").trigger('click');

    /**trigger on buttonanswers*/
    else if(e.charCode == 97 || e.charCode == 65 ) // a or A
        $("#Q0").trigger('click');
    else if(e.charCode == 98 || e.charCode == 66 ) // b or B
        $("#Q1").trigger('click');
    else if(e.charCode == 99 || e.charCode == 67 ) // c or C
        $("#Q2").trigger('click');
    else if(e.charCode == 100 || e.charCode == 68 ) // d pr D
        $("#Q3").trigger('click');

    /*trigger for save*/
    else if(e.charCode == 112 || e.charCode == 80 ) // p or P , for saving the results
        $("#saveButton").trigger('click');

}, false);
```

Abbildung 17- Shortcuts Implementierung

Der Aufruf *charCode* gibt den erzeugten numerischen Code der gedrückten Taste zurück und so kann das DOM Element dementsprechend angesteuert werden (*trigger*).

Es ist dennoch anzumerken, dass das Abfragen der Keyboard-Events nicht immer zuverlässig ist. Auf Tablets oder Smartphones wird das Drücken der Shift-Taste nicht unbedingt abgefangen und eine Unterscheidung in Groß- und Kleinschreibung kann schwer fallen.

Ein anderes Feature dieser Applikation war das Ertönen von Geräuschen bei Ereignissen. Dieses wird im nächsten Abschnitt vorgestellt und erklärt.

4.2 Sounds

Zu den Anforderungen an die neue Software gehörte auch an erster Stelle die Implementierung des An- und Ausschaltens des Tons.

Wie auf der Abbildung 15 zu sehen ist, kann mit einem Klick der Ton ein- bzw. ausgeschaltet werden. Die folgende Abbildung zeigt, wie es implementiert wurde.

Zuerst wird den Buttons für den Ton, die in der Datei `Gesichter_merken_Training.html` statisch erzeugt worden sind, eine Hintergrundfarbe gegeben. Die Variable „*Playsound*“ wird benutzt, um festzuhalten, ob den Ton ein- oder ausgeschaltet werden soll.

```
/*Sound on as default*/
$("#soundOn").css('background-color', 'green');

// Turn sound on, "sound on" buttons color green
$("#soundOn").click(function()
{
    playSound = true;
    $("#soundOff").css('background-color', '');
    $(this).css('background-color', 'green');

}
);

// Turn sound off, "sound off" buttons color green
$("#soundOff").click(function()
{
    playSound = false;
    $("#soundOn").css('background-color', '');
    $(this).css('background-color', 'green');

}
);
```

Abbildung 18- Sound Implementierung 1

JavaScript verfügt über eine Funktion, um an DOM-Elementen Sound oder ein Video anzuhängen. Der Parameter „*audio*“ in `document.createElement()` sagt JavaScript, dass es sich um ein Element vom Typ „*audio*“ handelt, das bedeutet, dass dem Element eine Datei vom Typ `.mp3` oder `.wav` angehängt wird. In dem Fall, dass die Variable „*playSound*“ den Wert „*true*“ hat, d.h. ein Ton bei Antworten ausgegeben werden soll, wird anschließend dem Element, entsprechend dem Erfolg auf die jeweilige Frage, der passende Ton angehängt.

Dies wird in der folgenden Abbildung veranschaulicht.

```

/* play sound*/
if(playSound)
{
    var audioElement = document.createElement('audio');
    if(answer)
    audioElement.setAttribute('src', 'sounds/success.wav');
    else
    audioElement.setAttribute('src', 'sounds/fail.mp3');

    audioElement.setAttribute('autoplay', true);
    audioElement.load();
}

```

Abbildung 19- Sound Implementierung 2

4.3 Speichern von Statistiken

Ein weiteres Anliegen an die neue Anwendung war die Möglichkeit, Statistiken speichern und ausdrucken zu können.

Sobald die Spielrunde zu Ende ist, bekommt der Benutzer die Statistiken der Spielrunde in einer Tabelle angezeigt und er hat dann die Möglichkeit, sie auszudrucken. Die Abbildung 20 zeigt, wie die Statistiken auf der graphischen Benutzeroberfläche dargestellt werden.

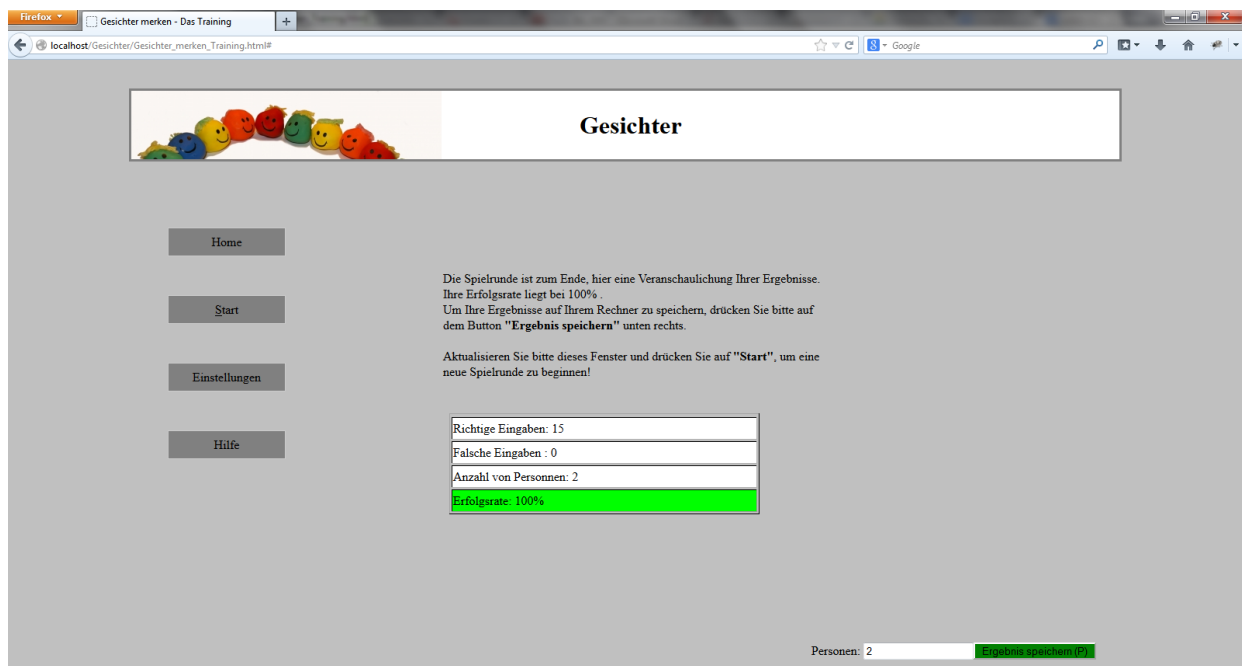


Abbildung 20-Statistiken Implementierung 1

Die Eingaben sind in zwei Gruppen unterteilt und zwar in richtig und falsch. Zusätzlich wird die Erfolgsrate berechnet.

Die Erfolgsrate repräsentiert die Anzahl der falschen Eingaben im Verhältnis zu der Anzahl von Personen multipliziert mit der Anzahl von möglichen Fragen. Außerdem ist der Buchstabe „P“ auf der Schaltfläche zu sehen. Er weist den Anwender darauf hin, dass er das Tastaturkürzel (engl. shortcut) (groß oder klein) benutzen kann, anstatt mit der Maus darauf zu klicken.

Als nächstes wird der Code genau betrachtet.

```

/* feedback to the user about his result, the results are displayed in a table*/
var tableR = '<td>' + 'Richtige Eingaben: ' + prot.getNumberofRights() + '</td>';
var tableF = '<td>' + 'Falsche Eingaben : ' + prot.getNumberofWrongs() + '</td>';
var tableP = '<td>' + 'Anzahl von Personen: ' + prot.getNumberofPersons() + '</td>';

var succesRate = Math.round(100 - (prot.getNumberofWrongs() * 100) / (prot.getNumberofPersons() * 3)) ;

if(prot.getNumberofWrongs() > 0)
{
    var finalText = '<p>' + 'Die Spielrunde ist zum Ende, hier eine Veranschaulichung Ihrer Ergebnisse. Ihre Erfolgsrate liegt bei ' + succesRate + '% .' + '</br></br>' + 'Um Ihre Ergebnisse auf Ihrem Rechner zu speichern, drücken Sie bitte auf dem Button "Ergebnis speichern" unten. ' + '</br>' + 'Aktualisieren Sie bitte dieses Fenster und drücken Sie auf "Start", um eine neue Spielrunde zu beginnen!' + '</p>'
    var tableS = '<td>' + 'Erfolgsrate: ' + succesRate + '% .' + '</td>';
}
else
{
    var finalText = '<p>' + 'Die Spielrunde ist zum Ende, hier eine Veranschaulichung Ihrer Ergebnisse. Ihre Erfolgsrate liegt bei ' + '100% .' + '</br>' + 'Um Ihre Ergebnisse auf Ihrem Rechner zu speichern, drücken Sie bitte auf dem Button <b>Ergebnis speichern</b> unten. ' + '</br></br>' + 'Aktualisieren Sie bitte dieses Fenster und drücken Sie auf "Start", um eine neue Spielrunde zu beginnen!' + '</p>'
    var tableS = '<td>' + 'Erfolgsrate: ' + '100%' + '</td>';
}
$('#r').append(tableR);
$('#f').append(tableF);
$('#P').append(tableP);
$('#score').append(tableS);
$('#finalR').append(finalText);

```

Abbildung 21 - Statistiken Implementierung 2

Wie schon vorher ausgeführt, bietet JavaScript auch die Möglichkeit, Elemente dynamisch zu erzeugen. Der Tag `<td>` wird in HTML benutzt, um Tabellen zu zeichnen. Tatsächlich wird dieser für die Anwendung „Gesichter“ benutzt, um Statistiken tabellarisch darzustellen. Die Funktionsaufrufe (u.a. `prot.getNumberofRights()`) werden benutzt, um die Anzahl von falschen bzw. richtigen Eingaben zu extrahieren und sie in der Tabelle anzeigen zu lassen. Die genauere Bedeutung eines solchen Aufrufs wird zu einem späteren Zeitpunkt erklärt.

Wie auf der Abbildung 20 zu sehen ist, kann der Benutzer mit einem Klick auf die Schaltfläche die Statistiken auf seinem Rechner speichern.

Aus Sicherheitsgründen ist es sehr schwierig, mit JavaScript in einer Datei zu schreiben und sie dann auf dem Client-Rechner zu speichern. Selbst wenn man es schafft, ist die Validität auf vielen Browsern nicht gewährleistet. Deswegen wurde für die Anwendung „Gesichter“

ein PHP-Script (Endung .php) implementiert, welches das Schreiben und Speichern übernimmt.

Das Script muss einer Form hinzugefügt werden und mithilfe der Eigenschaft „*POST*“ werden Daten vom Browser (Client) an den Webserver gesendet, wo diese dann ausgewertet werden können. Nach der Auswertung sendet der Server dem Client eine Antwort, das heißt die Speicherung erfolgt.

Um eine solche Transaktion allerdings durchführen zu können, braucht man einen Webserver. Dieser kann leicht aus dem Internet heruntergeladen werden und die JavaScript-Applikation auf diesen gelegt. Für die Realisierung von „Gesichter“ wurde der Server „WAMPSEVER“ benutzt. Die folgenden Abbildungen (Abbildung 22 und Abbildung 23) illustrieren den oben erklärten Prozess.

```
<form name="textFieldform" id= "textFieldform" action = "statisticsStoring.php" method="POST">
<div id="RigthLabel">

</div>
<div id="WrongLabel">

</div>
<div id="persLabel">
</div>
```

Abbildung 22- Statistiken speichern: Php-Script in HTML Formular

```

<?php
$File = "Gesichter_Results.txt";
$Handle = fopen($File, 'w');
$Data = "LernReha-Software:\t";
    fwrite($Handle, $Data);
$Data = "GESICHTER\n";
    fwrite($Handle, $Data);
$Data = "Aufgabe: Zu Portraits Namen, Wohnort und Beruf merken.\n\n";
    fwrite($Handle, $Data);
$Data = "Datum & Zeit:\t\t\t";
    fwrite($Handle, $Data);
$Data = date('D, d M Y H:i:s T')." \n";
    fwrite($Handle, $Data);
$Data = "Richtige Eingaben:\t\t";
    fwrite($Handle, $Data);
$Data = $_POST['True']." \n";
    fwrite($Handle, $Data);
$Data = "Falsche Eingaben:\t\t";
    fwrite($Handle, $Data);
$Data = $_POST['False']." \n";
    fwrite($Handle, $Data);
$Data = "Anzahl von Personen: \t\t";
    fwrite($Handle, $Data);
$Data = $_POST['Personen']." \n";
    fwrite($Handle, $Data);
header("Location:Gesichter_merken_Training.html");
fclose($Handle);

?>

```

Abbildung 23- Statistiken speichern: Php-Script

Nach dem Klick auf die Schaltfläche zum Speichern der Ergebnisse wird ein Dokument vom Typ .txt im gleichen Verzeichnis, in dem der Server installiert wurde, angelegt. Dieses Dokument kann mit dem Programm WordPad, welches auf fast allen modernen Rechnern vorinstalliert ist sowie mit Notepad ++ geöffnet werden. Diese Datei kann auch mit einem Webbrowser gelesen werden. Die folgende Abbildung (Abbildung 24) zeigt die erzeugte Datei, welche mithilfe des Webbrowsers Internet Explorer geöffnet wurde.

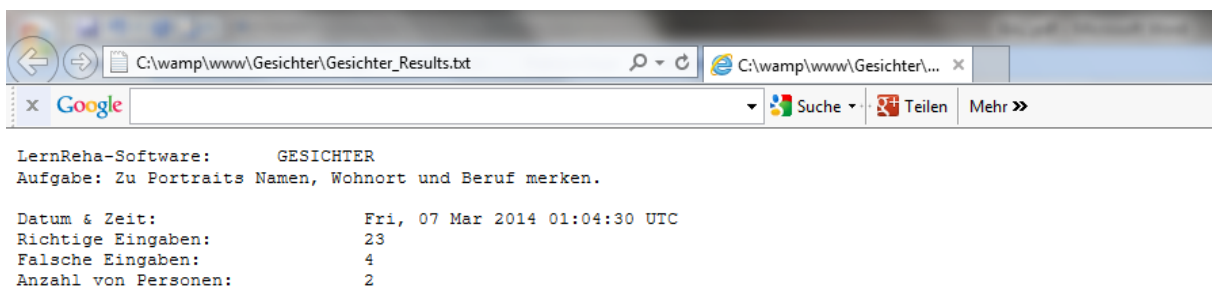


Abbildung 24- Gespeicherte Statistiken geöffnet mit IE

Nach dem Vorstellen der Features der neuen Anwendung wird jetzt auf die Implementierung der Spiellogik eingegangen. Im nächsten Abschnitt wird diese erläutert.

4.4 Implementierung der Spiellogik

Bei der Implementierung dieser Logik sind mehrere Funktionen involviert.

Hier ist eine Übersicht über die Funktionen, die an der Implementierung der Interaktion beteiligt sind. Alle diese Funktionen sind in `Gesichter.js` implementiert.

logic Manager	Statistics
<code>+imagesManager()</code> <code>+questionAsker()</code> <code>+questionsSwitcher()</code> <code>+answerChecker()</code> <code>+shuffleArray()</code> <code>+finishRound()</code> <code>+tabsManager()</code>	<code>-correctAnswers</code> <code>-wrongAnswers</code> <code>-personNumber</code> <code>-nextQuestionCounter</code> <code>-nextImageCounter</code> <code>-indexOfCurrentImage</code> <code>-indexOfQuestion</code> <code>+increaseCorrectAnswer()</code> <code>+increaseWrongAnswer()</code> <code>+getIndexOfCurrentImage()</code> <code>+getIndexOfQuestion()</code> <code>+getNumberOfRightsr()</code> <code>+getNumberOfWrongs()</code> <code>+getNumberOfPersons()</code> <code>+setindexOfCurrentImage()</code>

Abbildung 25- Aufbau der JavaScript-Datei `Gesichter.js`: Eigenschaft- und Funktionsübersicht

Die Funktionen `imagesManager()`, `questionsSwitcher()`, `answerChecker()` und `questionsAsker()` sind für die Steuerung in dem Modul „Gesichter merken“ zuständig.

Die Funktion `imagesManager()` ist für die Darstellung von Portraits verantwortlich. Dort werden Events wie das Klicken auf die Schaltflächen „zurück“ und „weiter“ gehandelt.

Die Funktion `questionsSwitcher()` ist für das Umschalten (engl. switch) zu der nächsten Frage zuständig. In der Funktion `answerChecker()` wird die Eingabe des Anwenders auf ihre Richtigkeit geprüft, dementsprechend wird ein Ton ausgegeben und die Anzahl von korrekten bzw. falschen Antworten erhöht.

Zur dynamischen Generierung von Antwortmöglichkeiten wurde die Methode `questionsAsker()` implementiert. Der folgende Codeauszug (Abbildung 26) soll ein Beispiel für die Generierung sein.

```

// shuffle array to have a different order of Names / works / villages on the buttons
shuffleArray(buttonArray);

for(var indx = buttonArray.length - 1; indx >= 0;
indx -- )
{
    var idB = "#Q" + indx;
    var newButton = $(idB).text(buttonArray[indx]);

    var buttonText = buttonArray[indx];

    if(collectImageQSliced.indexOf(picture) == collectNameQSliced.indexOf(buttonText))
    {
        $(idB).addClass('correct');
    }

    $(newButton).click(function ()
    {
        if($(this).hasClass("correct"))
            answerChecker(true);

        else
            answerChecker(false);
    }
    );
}

```

Abbildung 26- Dynamische Generierung von Antwortmöglichkeiten in Gesichter.js

Die Funktion *tabsManager()* ist für das dynamische Erzeugen und Steuerung von Portraits, Beschreibungen, und Buttons („weiter“ und „zurück“) für das Modul „Wie merkt man sich Namen und Gesichter“ verantwortlich.

Zunächst wird das Array mit Antwortmöglichkeiten mithilfe der Funktion *shuffleArray()* zufällig gemixt, so dass die Antwortoptionen nach jedem Durchlauf anders angezeigt werden können. Anschließend werden in *Gesichter_merken_Training.html* dynamisch hergestellte Schaltflächen in einem Schleifentext (Antwortoptionen) hinzugefügt und die korrekte Option gekennzeichnet. Abschließend werden Schaltflächen an einen Event-Handler angehängt, auf welchen die Benutzereingaben auf ihre Richtigkeit geprüft werden sollen.

Außer den gewöhnlichen Eigenschaften zum Abändern von DOM-Elementen, besitzen die Funktionen in JavaScript zusätzliche prototype-Eigenschaften. Dies ist Dank des JavaScript-Frameworks *Prototyp* der Fall.

Ein Codebeispiel aus der JavaScript - Datei *Gesichter.js*(Abbildung 27)

```

function VariablesForPrototyping()
{
    this.correctAnswers = 0;
    this.wrongAnswers = 0;
    this.personNumber = 0;
    this.nextQuestionCounter = - 1;
    this.nextImageCounter = 0;
    this.indexOfCurrentImage = 0;
    this.indexOfQuestion = 0;

}
var prot = new VariablesForPrototyping();

VariablesForPrototyping.prototype.getIndexOfCurrentImage = function()
{
    return this.indexOfCurrentImage;
}
;

```

Abbildung 27- Prototype-Eigenschaft von JavaScript : Codeauszug aus Gesichter.js

Für mehr Verständnis kann die Funktion *variablesForPrototyping()* als Konstruktor in Java, in welchem die Variablen definiert sind, angesehen werden. Um auf diese Variablen zuzugreifen bzw. sie manipulieren zu können, muss ein solcher Aufruf, wie dies im dritten Abschnitt der Abbildung 27 zu sehen ist, gemacht werden.

5 Test

Die Testphase ist eine sehr wichtige Phase bei dem Softwareentwicklungsprozess. Gefundene Fehler in der Testphase schützen vor sehr hohen Kosten in der Einführungsphase.

Im Software-Test gibt es viele Arten des Testens. Eine davon ist der Akzeptanztest. Bei einem Akzeptanztest oder User Acceptance Test (UAT) wird geprüft, ob das implementierte System die formulierten Anforderungen erfüllt. Dafür werden Testfälle formuliert. Für einen Akzeptanztest wird das Blackbox-Verfahren angewendet, d. h. der Test orientiert sich nicht am Code der Software, sondern nur am Verhalten der Software bei spezifizierten Situationen bzw. Handlungen (Eingaben des Benutzers, Grenzwerte bei der Datenerfassung, usw.).

In diesem Kapitel wird ein Akzeptanztest auf dem Modul „Gesichter merken“ durchgeführt. Wie im Kapitel 4 angekündigt, wird die Lauffähigkeit der Applikation nur auf den Browsern Firefox 26, Google Chrome 35 und IE 11 garantiert.

5.1 Testfall: Beginnen einer Spielrunde

Anweisung: Öffnen Sie die Anwendung und klicken Sie auf die Schaltfläche „Start“. Alternativ drücken Sie auf die Tastaturtaste „S“ (groß oder klein).

Soll-Verhalten: Es öffnet sich ein Fenster, in dem Sie eine Zahl eingeben können.

5.2 Testfall: Anzahl angezeigter Bilder passend zu der eingegebenen Nummer

Anweisung: Das Fenster zum Eingeben der Anzahl von Personenbildern ist jetzt offen. Geben Sie eine Zahl Ihrer Wahl zwischen zwei(2) und zehn(10) ein.

Soll-Verhalten: Die Anzahl gezeigter Personenbilder passt zu der angegebenen Zahl.

5.3 Testfall: Switchen zwischen den Portraits

Anweisung: Personenbilder werden jetzt angezeigt. Switchen Sie zwischen den Bildern mithilfe der Schaltflächen „Weiter“ und „Zurück“. Alternativ benutzen Sie das Tastaturkürzel „W“ (groß oder klein) für „Weiter“ oder „Z“ (groß oder klein) für „Zurück“.

Soll-Verhalten: Das Switchen zwischen den Bildern erfolgt reibungslos.

5.4 Testfall: Korrekte Statistiken, Funktionieren des Tons und der Tastenkürzel

Szenario: Die Vorstellungsrunde ist vorbei, jetzt wird abgefragt. Sie bekommen ein Portrait mit vier (4) Antwortmöglichkeiten zu sehen.

a. Korrekte Antwort

Anweisung: Vergewissern Sie sich, dass der Ton eingeschaltet ist und drücken Sie auf die richtige Antwortmöglichkeit. Alternativ benutzen Sie einen der Buchstaben zur Auswahl der richtigen Antwort.

Soll-Verhalten: Das Portrait, die Antwortmöglichkeiten und die Frage werden für ein paar Sekunden ausgeblendet. Ein glücklicher Smiley wird angezeigt und gleichzeitig wird ein Sound abgespielt. Zudem wird in der Fußzeile die Anzahl der richtig beantworteten Fragen um eins erhöht.

Danach wird das nächste Portrait mit der gleichen Frage oder mit einer anderen Frage erscheinen. In dem Fall, dass eine andere Frage erscheint, wird die neue Frage orange eingefärbt sein und für ein paar Sekunden blinken.

b. Falsche Antwort

Anweisung: Drücken Sie auf eine falsche Antwortmöglichkeit oder benutzen Sie alternativ einen Buchstaben zur Auswahl der Antwort.

Soll-Verhalten: Das Portrait, die Antwortmöglichkeiten und die Fragen werden für ein paar Sekunden ausgeblendet, eine traurige Animation wird angezeigt und gleichzeitig ein Sound abgespielt. Zudem wird in der Fußzeile die Anzahl falsch beantworteter Fragen um eins erhöht.

Danach wird das gleiche Portrait mit der gleichen Frage erscheinen.

5.5 Testfall: Ende der Spielrunde: Anzeigen und Speicherung der Ergebnisse

Szenario: Die Spielrunde ist zu Ende.

Soll-Verhalten: Es erscheinen auf dem Bildschirm Anweisungen und eine Tabelle mit den Statistiken der vergangenen Spielrunde.

Anweisung: Zum Speichern Ihrer Ergebnisse drücken Sie auf der Schaltfläche „Ergebnis speichern“ oder benutzen alternativ das Tastaturkürzel „P“ (Groß oder klein).

Soll-Verhalten: Die Ergebnisse werden in dem Benutzerrechner gespeichert.

6 Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde die Web-Anwendung „Gesichter“ konzipiert und entwickelt. „Gesichter“ unterstützt Patienten bei der Wiedererlangung ihres durch Hirnschädigung verlorenen Langzeitgedächtnisses. In diesem Kapitel wird die Arbeit zusammengefasst. Bei dieser Zusammenfassung werden die Ergebnisse vorgestellt. Anschließend werden Möglichkeiten zur Erweiterung der Anwendung dargestellt und gezeigt, wie diese umgesetzt werden könnten.

Am Anfang stand die Frage, ob die vorliegende Software „Gesichter“ lediglich korrigiert werden sollte oder ob eine Neu-Implementierung erforderlich ist. Um dies zu klären wurde die Software vom Groben ins Feine examiniert. Dabei wurde unter anderem der Mangel an Ergonomie der Benutzeroberfläche sowie eine massive Codeduplizierung und statisches Erzeugen der Elemente festgestellt. Außerdem war es wünschenswert, dass die neue Anwendung auf mehreren Betriebssysteme lauffähig ist und nicht nur auf Windows Systemen. Daraus folgend fiel die Entscheidung für eine Neu-Implementierung der Applikation.

Anhand der im Abschnitt 1.2 formulierten Ziele wurden zuerst nach einem selektiven Verfahren die Technologien und derer Techniken ausgewählt. Aus einer in VB5 implementierten Windows Desktop-Applikation ist eine JavaScript-Anwendung geworden. Die Anwendung lässt sich dank der genutzten Bibliothek leicht erweitern.

Zusammenfassend kann man sagen, dass die Realisierung der formulierten Anforderungen an die neue Applikation erfolgreich verlaufen ist. Die statische Generierung von Elementen wurde größtenteils durch eine dynamische ersetzt.

Die Benutzerfreundlichkeit beachtend wurde eine einfache, selbstsprechende und dynamische Oberfläche konzipiert und entwickelt. Außerdem ist die Interaktion zwischen dem Programm und dem Anwender mithilfe von Tonausgaben und dem Anzeigen von Smileys erreicht worden. Ferner wurde der Wunsch, Benutzereingaben festhalten zu können, formuliert. Auch dieses Ziel wurde erreicht. Nun können Statistiken (Eingaben, Anzahl von Personen) mit der Uhrzeit im Benutzerrechner gespeichert werden. Da von jedem Rechner aus eine Datei ausgedruckt werden kann, wurde die Möglichkeit des Druckens der Statistiken aus der Anwendung außer Acht gelassen.

Ein weiteres Bestreben war, die neue Applikation an aktuelle medizinische Kenntnisse anzupassen. Davon wurde aber im Laufe der Arbeit abgesehen. Tatsächlich wurde in

Abschnitt 2.2.1 erläutert, dass es keine allgemein gültige Ansicht über die Effektivität von Computer-Training formuliert werden kann.

Die in dieser Arbeit hergestellte Anwendung ist noch nicht vollständig, da es sich bei dieser nur um einen Prototyp handelt. Der erste Stein wurde gelegt, der Rest muss noch in einem neuen Release hinzugefügt bzw. korrigiert werden. Allerdings läuft die aktuelle Programmversion noch nicht fehlerfrei. Das Inkrementieren der richtig- und falsch- Eingaben erfolgt noch nicht fehlerfrei. Außerdem die Schaltfläche zum Speichern der Ergebnisse funktioniert nicht bei IE, jedoch mithilfe DesTastaturkürzels „P“ funktioniert es einwandfrei. Wünschenswert sind die Erweiterung der Applikation mit mehr Personenbildern sowie das Einfügen von verschiedenen Schwierigkeitsleveln.

Anhang A: Literatur

A.1 Bücher

- [Kast01]: Erich Kasten, Gabriele Schmid, Reinhard Eder:
Effektive neuropsychologische Behandlungsmethoden
Deutscher Psychologen Verlag, Überarbeitete Neuauflage, Bonn 2001
- [Huch07]: Renate Huch, Klaus D. Jürgens :
Mensch Körper Krankheit
Urban & Fischer Verlag, 5. Auflage, München 2007
- [Kast07]: Erich Kasten:
Einführung Neuropsychologie
Reinhardt Verlag, München 2007
- [Schu12]: Gisela C. Schulze, Andreas Zieger (Hrsg.):
Erworbene Hirnschädigungen.
Klinkhardt Verlag, Kempten 2012
- [Born09]: Günter Born:
Computer trotz Handicap
Markt + Technik Verlag, München 2009
- [Otto05]: Angelika Thöne-Otto, Hans J. Markowitsch:
Gedächtnisstörungen nach Hirnschäden
Hogrefe Verlag, Göttingen 2005
- [Demm10]: Thomas Demming
jQuery Kochbuch
O'Reilly Verlag, Köln 2010
- [Matt10]: Stephan Mattescheck
jQuery Das Praxisbuch
Galileo Computing Verlag, 1.Auflage, Bonn 2010
- [Ochs12]: Oliver Ochs
JavaScript für Enterprise-Entwickler
dpunkt Verlag, 1.Auflage, Heidelberg 2012

A.2 Internetseiten

- [BrainHu2]: <http://www.airflag.com/Hirn/w3/schnitt.jpg>
Stand März 2013
- [BrainHu]: <http://www.borderlinezone.org/gehirn/gehirn1.jpg>
Stand März 2014

- [BrainLa]: [http://www.pfizer.de/fileadmin/pfizer.de/images/gesundheit/Humanmedizin/Gehirn und Nervensystem/Epilepsie/31 04 6 gehirn 100514_hq.jpg](http://www.pfizer.de/fileadmin/pfizer.de/images/gesundheit/Humanmedizin/Gehirn%20und%20Nervensystem/Epilepsie/31_04_6_gehirn_100514_hq.jpg)
Stand Januar 2014
- [GedMo]: http://www.meb.uni-bonn.de/psychiatrie/medpsy/manuskripte/sitzung_1_gedaechtnis_ss2005_kurz_3b_4b_5a.pdf
Stand Dezember 2013
- [ZwiHirn]: [http://www.onmeda.de/anatomie/gehirn_anatomie-zwischenhirn- \(diencephalon\)-1478-4.html](http://www.onmeda.de/anatomie/gehirn_anatomie-zwischenhirn-_(diencephalon)-1478-4.html)
Stand Januar 2014
- [GroHirn]: <http://flexikon.doccheck.com/de/Gro%C3%9Fhirn>
Stand Januar 2014
- [KogniF]: <http://www.medien.ifi.lmu.de/lehre/ws0506/mmi1/kognitive-faehigkeiten.xhtml>