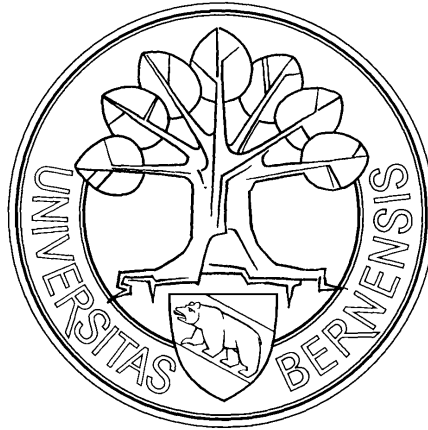


# Informatikprojekt

März – August 2003



## Stroke Datenbank

Reto Zurbuchen



Betreuer:

Prof. O. Nierstrasz, PD Dr. Maja Steinlin, Dr. Julia Pavlovic

# Informatikprojekt

## Stroke Datenbank

### Dokumentation

Inhaltsverzeichnis:

- Projektübersicht
- Arbeitsauftrag / Requirements Overview
- Architektur und Designoverview
- Entwicklerdokumentation
- Benutzerdokumentation
- Fazit des Projektes
- Appendix

## **Akuter cerebraler Infarkt im Kindesalter: Langzeitverlauf und Wiederholungsrisiko** (Kurzbeschreibung)

Akuter cerebraler Insult ist eine seltene, aber schwerwiegende Erkrankung im Kindesalter. In den wenigen Studien betreffend akutem cerebralen Insult im Kindesalter wird eine unterschiedliche Häufigkeit von 1-10 / 100'000 Kindern / Jahr festgestellt. Abklärungen und Therapien unterscheiden sich von denjenigen im Erwachsenenalter, bedingt durch unterschiedliche Ätiologien (immunologische Probleme, Stoffwechselstörungen, angeborene Vasculopathie, unreifes Koagulationssystem etc.) sowie unterschiedlicher Prognose betreffend Erholung. Nur wenige Studien haben sich mit diesem Problem im Kindesalter beschäftigt (Schoenberg, Kirkham, Eeg-Olafson). Indikationen für Behandlungen in der akuten Episode sowie spätere prophylaktische Therapien betreffend Wiederholungsrisiko sollten auf dem Wissen um Ursache, Prognose und Wiederholungsrisiko des akuten Insultes im Kindesalter basieren.

Seit dem 1.1.2000 führen wir in der Schweiz eine epidemiologische Studie betreffend Inzidenz, Ursache und kurzfristigen Verlauf der in der Schweiz lebenden Kinder mit akutem Insult durch (**Swiss Neuropaediatric Stroke Registry**). In einer multizentrischen Studie sollten nun Erholung und Wiederholungsrisiko einiger dieser Kinder genauer und über längere Zeit untersucht werden.

### **Patienten und Methoden:**

Folgende Untergruppen der durch die SNPSR anonym registrierten Kinder werden in einer multizentrischen Studie betreffend Erholung und Wiederholungsrisiko genauer verfolgt:

**Arteriell ischämischer Infarkt (AIS):** *Eingeschlossen:* hämorrhagische Infarkte

*Ausgeschlossen:* primär cerebrale Blutung, Frühgeborene < 36 SSW

Klinische und bildgebende Kriterien müssen wie folgt erfüllt sein:

- 1) Akut beginnender, fokaler neurologischer Ausfall für mind. 20 Minuten  
**mit** korrespondierendem ischämischem Infarkt in CT oder MRI
- 2) Bei Neugeborenen: Lethargie oder cerebraler Krampfanfall als akute Symptome  
**mit** ischämischem Infarkt im CT oder MRI
- 3) ausgeschlossen für Studie Langzeitverlauf und Wiederholungsrisiko
- 4) Hypoxisch ischämisches Event mit diffussem oder bilateralem Befall, nur falls in Bildgebung klarer fokaler Infarkt in einem definierbaren Gefässterritorium vorliegt

**Sinusvenenthrombose** Frühgeborene Kinder sind eingeschlossen

Klinisch und bildgebende Kriterien müssen wie folgt erfüllt sein

- 1) Jede transitorische neurologische Dysfunktion (eingeschlossen: Kopfschmerzen, Krämpfe, Bewusstseinsstrübung, fokale neurol. Symptome)  
**mit** Thrombose cerebraler Venen oder Sinus in MRI, MRA oder Angiographie  
CT allein wird bei eindeutigem Befund akzeptiert bei Kindern < 1 Monat

Die initiale Datensammlung wird identisch sein mit derjenigen der gesamtschweizerischen Datenerfassung. Eine ausführliche neurologische Untersuchung wird in der Akutphase und bei Austritt aus dem Spital durch den betreuenden Neuropädiater durchgeführt werden. 6 und 18 Monate nach dem akuten Ereignis wird eine weitere neurologische sowie auch entwicklungsneurologische Untersuchung durch einen Untersucher der Studienzentrale durchgeführt werden. Zu diesem Zeitpunkt wird auch eine Vervollständigung der Daten aus der Krankengeschichte des Patienten durch den Untersucher der Studienzentrale durchgeführt werden. Alle diese Kinder werden durch einen jährlich versandten Fragebogen betreffend persistierender Probleme und Wiederauftreten eines akuten Ereignis verfolgt werden.

### **Datenerfassung und Auswertung**

Die Daten werden gemäss den beiliegenden Datenerfassungsformularen durch eine Access Computerdatabase erfasst werden.

Sie werden analysiert betreffend primäre Ursache, mögliche Risikofaktoren, Lokalisation des Infarktes im Vergleich zu Ursache / Outcome, primäre MRI/MRA Befunde im Vergleich zu Outcome, Vergleich neurologischer und neuropsychologischer Befunde zu Bildgebung, Outcome im Vergleich zu medikamentöser Therapie, Outcome im Vergleich zu Rehabilitationsmassnahmen, Wiederholungsrisiko im Vergleich zu Ursache, Lokalisation, Schweregrad (primär und Outcome) sowie Alter des ersten Infarktes. Es wird eine deskriptive Statistik der verschiedenen Daten erfolgen, für die Risikofaktoren und prognostischen Faktoren wird eine Multivariantanalyse durchgeführt werden.

# Requirements Overview

## Gesamtziel:

Ziel der Arbeit ist das Ausarbeiten einer MS Access 2000-Datenbank, die für das Erfassen und Auswerten der Daten, die beim „Swiss Neuropaediatric Stroke Registry SNPSR“-Projekt gesammelt werden genutzt werden kann.

## Beschreibung der Teilziele:

### Programmiersprache:

- Da bereits ein Teil auf MS Access 2000 gemacht worden ist, ist es am Sinnvollsten bei diesem zu bleiben

### Datenstruktur:

- Die Datenstruktur ist fertig und muss nur noch in Ausnahmefällen angepasst werden.

### Datentrennung:

- Im Moment ist das GUI als auch die Daten in der gleichen Datei abgelegt. Dies sollte behoben werden.

### Bereich Dateneingabe:

- Die Dateneingabe ist zwar rudimentär vorhanden aber sie sollte noch besser werden. Dabei müssen die Wünsche von den Usern berücksichtigt werden.

### Bereich Datenausgabe (Reports):

- Sämtliche Daten eines Patienten müssen auf eine einfache Art druckbar sein.
- Für wichtige Werte sollte auch eine Teilauswertung über alle Patienten hinweg möglich sein

### Statistik / Abfragen:

- Sobald die wichtigsten Daten erfasst sind, muss das Programm noch genau zu definierende Abfragen und Statistiken berechnen und ausgeben können. Detailinformationen siehe Appendix A

### Schnittstellen:

- Das Programm sollte eine Möglichkeit bieten, dass alle Daten sowie auch nur Teile daraus automatisch in eine Exceldatei exportiert werden können.

### Projektstart:

1. März 2003

### Projektende:

31. Juli 2003

Da ich eventuell im Juli im Militär bin, so würde sich das Projekt automatisch um 3 Wochen verlängern.

### Kontaktpersonen:

Betreuerin Seite Inselspital: Fr. PD Dr. Maja Steinlin, Fr. Dr. Julia Pavlovic

Betreuer Seite IAM: Prof. O. Nierstrasz

# Architektur und Designübersicht

## Allgemeine Bemerkungen zum Programm:

Das Programm wurde mit und für MS Access 2000 entwickelt. Dies bedingt, dass jeder Benutzer eine MS Access 2000 Lizenz benötigt.

Das Programm wird mit dem öffnen der Stroke2000.mdb gestartet.

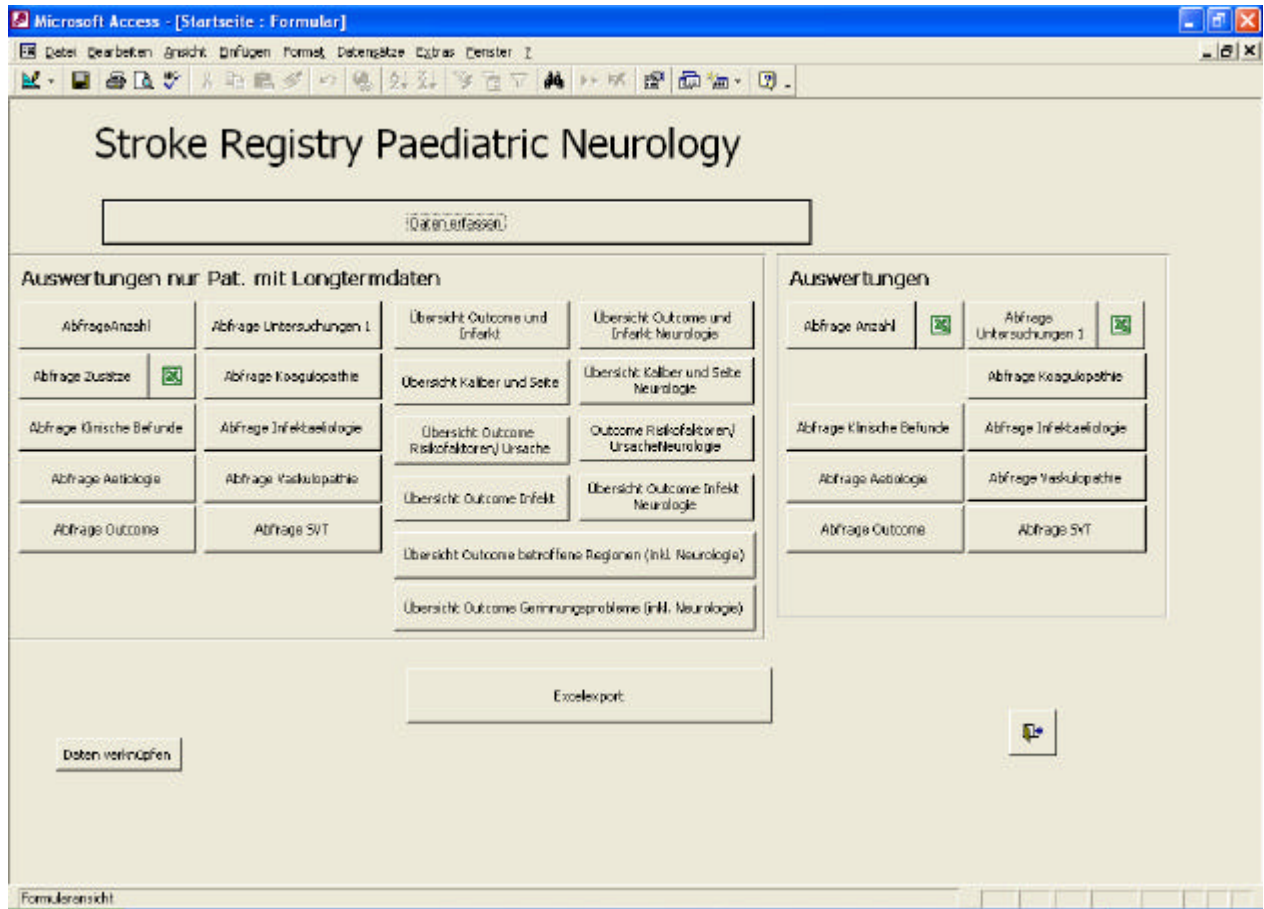
## Architektur und Designübersicht

Das ganze Programm besteht aus genau 2 Dateien. Sie heissen Stroke2000.mdb und Stroke 2000 Daten.mdb. In der letztgenannten Datei werden sämtliche Daten erfasst. Die andere ist für den Rest d.h. die Logik, die Reports und das GUI verantwortlich. Dies ermöglicht den Benutzer die Daten weiterzuleiten ohne dass die zu verschickende Datei sehr gross ist.

Grundsätzlich ist die Architektur sehr einfach. Mit den beiden Ausnahmen Excelexport und Berichtsvorschau ist der zugrundeliegende Code direkt im GUI eingebaut. Nicht-Access-Programmierer findet das ohne Zweifel sehr merkwürdig aber es ist einfach das Konzept von MS Access. Die beiden Ausnahmen werden übrigens in der Entwicklerdokumentation Seite 8 und 9 genauer beschrieben (Ablaufdiagramm).

Für genauere Informationen bezüglich der einzelnen Komponenten bediene man sich der Entwicklerdokumentation. Dort findet man auch ein ER-Diagramm der Datenbank.

# Entwicklerdokumentation Stroke Datenbank



## 0. Inhaltsverzeichnis

Entwicklerdokumentation Stroke Datenbank.....	1
0. Inhaltsverzeichnis .....	1
1. Benötigte Infrastruktur .....	2
1.2. Benötigte Hardware .....	2
2. Beschreibung / Bemerkungen zu den einzelnen Komponenten .....	2
2.1 Tabellen.....	2
2.2. Abfragen .....	3
2.3 Formulare .....	3
2.4 Berichte .....	3
2.5 Seiten .....	3
2.6 Makros.....	3
2.7 Module und ihre Klassenmethoden .....	3
3. Wichtige Aspekte von Code welche sich innerhalb der GUI befinden .....	6
3.1 Wichtige Methoden .....	6
4. ER-Diagramm.....	7
5. Ablaufdiagramme.....	8
5.1 Excelexport.....	8
5.2 Laden einer Abfrage als Bericht .....	9

## **1. Benötigte Infrastruktur**

### **1.1. Benötigte Plattform**

Microsoft Access 2000 (oder ev. höher)

### **1.2. Benötigte Hardware**

Sobald Microsoft Access 2000 einwandfrei läuft, funktioniert auch die Datenbank problemlos

## **2. Beschreibung / Bemerkungen zu den einzelnen Komponenten**

### **2.1 Tabellen**

#### Daten1, Daten2, Daten3, Daten4

In diesen 4 Tabellen, welche als einzige Komponenten unter »Stroke2000 Daten.mdb« abgespeichert sind, werden sämtliche Daten gespeichert, die unter »Daten erfassen« erfasst werden. Sie sind alle mit 1:1 mittels der Tabelle 1 verknüpft. Diese Verknüpfung wird beim setzen der Studien-ID erstellt.

#### Daten1Strings, Daten2Strings

Hier werden alle Boolean-Felder aus der Tabellen Daten1 bzw. Daten2 als Strings mit dem Feldtitelname oder ähnlich abgespeichert. Dies geschieht jeweils vor dem Ausdrucken eines Berichtes. Genaueres zu dieser Übersetzung siehe unter Modul BooleanToString.

#### Exportliste

Hier definiert der Benutzer welche Felder er für seinen Excelexport einbeziehen will oder nicht. Nach erfolgtem eingeben, werden die Werte ausgelesen und unter der Tabelle Teile abgespeichert. Genaueres dazu siehe unter Formular Exportliste (→ Code)

#### Teile

Sämtliche Gruppen, welche für den Export definiert wurden, werden hier abgespeichert. Die einzelnen Gruppen werden dabei als Query (→ String) abgespeichert und zwar nur der Teil zwischen SELECT und FROM. Das ganze Query wird dann beim effektiven Export gebildet. Genaueres dazu unter Modul Export

#### Steuerdaten

Diese Tabelle ist nicht mehr als eine Globaler Variablenspeicher, für das aktivieren bzw. deaktivieren des Filter um zwischen Alle-Patienten oder Nur-Longterm-Patienten umschalten zu können. Alle Unterabfragen sind auf diese Tabelle angewiesen. Das setzen dieses Wertes wird mit der Methode setFilter im Formular Hauptseite gesetzt.

#### T\_Pfad

Hier wird festgehalten, welche Tabellen von wo verknüpft werden sind/sollen. In diesem Projekt handelt es sich hierbei um die Tabellen Daten1, Daten2, Daten3 und Daten4. Dazu wird sie verwendet um beim Excelexport das Standardverzeichnis zu bestimmen.

#### Alle restlichen 51 Tabellen

Diese Tabellen füllen jeweils die einzelnen Werte der Kombinationsfelder. Teilweise werden sie mehrmals verwendet.

## 2.2. Abfragen

### Formular1, Formular2, Formular3, Formular4

Diese Abfragen werden benötigt, um alle Daten der jeweiligen Patienten einerseits in den gleichnamigen Formulare darstellen und andererseits sie auch ausdrucken zu können.

### DB-Export

Diese Abfrage, welche jeweils bei einem Excelexport neu erstellt wird, dient als letzten Teilschritt vor dem effektiven Export. Genau diese Daten die hier gewählt werden, werden schlussendlich exportiert.

### Alle restlichen 55 Abfragen

Diese Abfragen werden jeweils für die einzelnen Abfragenberichte verwendet.

## 2.3 Formulare

### Startseite

= 1 Seite die erscheint. Diese Seite verbirgt viel Code. Details siehe weiter unten.

### Formular1, Formular2, Formular3, Formular4

Dies sind die 4 Formulare welche für die Datenerfassung benötigt werden. Vorallem Formular1 verbirgt viel Code. Details siehe weiter unten.

### Teile, Exportliste

Beide Formulare werden für den Excelexport benötigt

### Alle restlichen 25 Formulare

Alle PopUp-Fenster unter Daten erfassen sind natürlich ein einzelnes Formular

## 2.4 Berichte

Hier es nicht notwendig alle einzelne Berichte zu kommentieren. Viele von diesen Berichten sind sowieso Unterberichte. Zudem ist nirgendwo irgendwelcher Code versteckt. Sobald die Berichte nichttriviale Abfragen verlangten, wurde die Abfrage separat gespeichert.

## 2.5 Seiten

Wurden keine verwendet

## 2.6 Makros

### M\_attache und M\_Detache

Beides sind Steuermakros welche die Ent- bzw. Verknüpfung der Maske mit den Daten regelt.

## 2.7 Module und ihre Klassenmethoden

### BooleanToString

Hier sind sämtliche Methoden zusammengefasst, welche die Übersetzung von Daten1 und Daten2 nach Daten1String und Daten2String machen.

+ BooleanToString()



Prüft ob genug Recordsets in Daten1String und Daten2String vorhanden sind, und erstellt gegebenenfalls neue. Nummeriert die Recordsets mit den identischen Nummern aus Daten1 und Daten2. Startet dann am Schluss BooleanToString1 und BooleanToString2. Zusätzlich wird noch die Funktion getMaxID(tbChoice as String) aus dem gleichen Modul gebraucht.

Kein Rückgabewert

- BooleanToString1()

Übersetzt alle Boolean aus Daten1 in Daten1Strings. Alle Werte werden dabei mit in Strings umgewandelt. Dabei wird jeweils in der Regel der Feldtitel und ein Komma hineingeschrieben, wenn der dortige Wert Wahr ist. Da es diverse Ausnahmen gab, war es mir leider nicht möglich diese Methode einfacher zu gestalten.

Am Schluss der Methode werden noch diverse Strings auch mehr oder weniger 1 zu 1 hinüberschrieben. Dies ist zwar Speicherplatzmässig sinnlos aber es macht dann das Leben bei den diversen Abfragen einfacher.

Die Methode prüft nicht, ob genug Recordsets in Daten1String zur Verfügung sind!

Kein Rückgabewert

- BooleanToString2()

Dito BooleanToString1

- getMaxID(tbChoice as String) as Integer

Rückgabewert: Gibt die grösste ID der Tabelle tbChoice zurück.

Die Tabelle muss dabei eine Feld names ID haben, damit diese Funktion einwandfrei läuft. Ausnahme: Daten2Strings. Bei dieser Tabelle wird der höchste Wert von IDTeil2 zurückgegeben.

## Export

+ ExcelExport()

Setzt den SQL-Statement für den Export zusammen. Sucht in der Tabelle Teile, die Feldnamen aller Gruppen welche gewählt worden sind.

Ruft dann die Methode SaveQuery(strSQL As String) auf, damit dieser Query gespeichert wird. Sobald dies geschehen ist, ruft es schliesslich noch die Methode TranferToExcel() auf, welche dann den Export ausführt.

kein Rückgabewert

+ SaveQuery(strSQL As String)

Speichert das Query (=strSQL) als Abfrage „Stroke DB-Export“ ab. Der Name der Abfrage wird durch die Konstante TABLENAME bestimmt. Falls diese Abfrage bereits existiert wird sie ohne Vorwarnung überschrieben.

+ TranferToExcel()

Benötigt Tabelle t\_Pfad um das Standardverzeichnis zu bestimmen. Exportiert dann die Tabelle „Stroke DB-Export“ (= Konstante TABLENAME) nach dem via der Methode DateiSpeichern (aus dem Modul mod\_DateiDialog) bestimmten Pfad und Dateinamen.

Kein Rückgabewert

## FieldsCounter

Diese Modul fasst viele Methoden, welche für die Bestimmung der Excelexportgruppen benötigt werden.

+ FieldsCounter(query As String) as Integer

Zählt die Anzahl Felder die ein beliebiges Query hat. Diese Methode wird für die Exportliste benötigt

Eingabewert: String mit dem zu untersuchenden Query

Rückgabotyp Integer mit Anzahl Felder

+ getFields(query As String, numberOfFields As Integer) as String  
Gibt das in numberOfFields gewählte Feld mit angehängtem Komma des Querys  
"query" als String zurück. Benötigt getFieldsNr1(query As String) und  
getOtherFields(query As String, numberOfFields As Integer)

- getFieldsNr1(query As String) as String  
Hilfsfunktion für getFields  
Funktionsbeschreibung in getFields

- getOtherFields(query As String, numberOfFields As Integer): String  
Hilfsfunktion für getFields  
Funktionsbeschreibung in getFields

### Reporthandler

Da ich in den Berichten keinen Code wollte, musste ich dieses Modul erstellen. Mit Ausnahme der Methode eliminate welche von einem String einfach die letzten 2 Zeichen eliminiert. D.h in der Regel das », « am Schluss der Aufzählung. Ansonsten greifen die Methoden meistens auf die Tabellen Daten1Strings und Daten2Strings zu. Es muss deshalb sichergestellt werden, dass diese 2 Tabellen zuvor aktualisiert worden sind.

### mod DateiDialog

Diese Methode ist eine Standardmethode, welche vom Internet heruntergeladen worden ist und somit auch nicht von mir entwickelt worden ist. Sie macht einen DLL-Aufruf für das Standard-Windows Speichern- und Ladefenster.

+ DateiOeffnen(Verzeichnis As String, Fenstertitel As String) as String  
Wird von diesem Projekt nicht verwendet

+ DateiSpeichern(Verzeichnis As String, Fenstertitel As String) as String  
Diese Methode wird für den Excel-Export verwendet.  
Parameter: Verzeichnis = Welches ist der Standardpfad, Fenstertitel = Wie soll das Fenster benannt werden  
Rückgabewert: Pfad eines (neues) \*.xls-File als String

+ DatenVerknuepfen(Verzeichnis As String, Fenstertitel As String) as String  
Diese Methode wird für Verknüpfung der Daten mit der Maske verwendet.  
Parameter: Verzeichnis = Welches ist der Standardpfad, Fenstertitel = Wie soll das Fenster benannt werden  
Rückgabewert: Pfad einer \*.mdb-Datei als String

### M Attache, M Detache und M Ouvre

Diese 3 Module werden allesamt zur automatischen Verknüpfung der Daten mit der Maske verwendet. Der Aufruf geschieht im Formular Startseite

### **3. Wichtige Aspekte von Code welche sich innerhalb der GUI befinden**

#### **3.1 Wichtige Methoden**

##### Startseite

- datenVerknüpfen\_Click()

Startet den „Filechooser“ und startet anschliessend die beiden Makros M\_Detache und M\_Attache damit zuerst die allfällige Entknüpfung mit anschliessender Verknüpfung realisiert wird. Der gewählte Pfad wird zudem in der Tabelle t\_Pfad gespeichert

+ ExcelExport(Queryname As String, File As String, showPath As Boolean)

Exportiert ein bestimmtes Query

Parameter:

Queryname: Name der Abfrage, welche exportiert werden soll.

File: Wohin soll die Abfrage exportiert werden

ShowPath: Soll am Schluss eine Nachricht erscheinen, welche den erfolgten Export bestätigt und den Speicherort nennt.

+ nurLongtermabfragen(berichtname As String)

lädt den mit Hilfe der Methode getReport den in berichtname gewählten Bericht, wobei nur die Datensätze angezeigt werden, welche Longterminformationen enthalten (= alle Daten mit 1 erfolgtem IQ-Test)

Setzt mittels der Methode setFilter den Queryfilter (siehe Tabelle Steuerdaten) auf True

+ berichtMitAllePatienten(stDocName As String)

lädt den mit Hilfe der Methode getReport den in berichtname gewählten Bericht Setzt mittels der Methode setFilter den Queryfilter (siehe Tabelle Steuerdaten) auf False

+ setFilter(setFilter As Boolean)

Setzt in der Tabelle Steuerdaten den gewählten Wert ein.

+ getReport(stDocName As String, whereCondition As String)

Lädt den mit stDocName gewählten Bericht unter der Filterberücksichtigung der in whereCondition gewählt wurde.

##### Formular1

+ activate()

Setzt alle „Details“-Knöpfe auf Visible oder Invisible (je nachdem ob das dazugehörige Kästchen True oder False ist.

Mithilfe der Methoden offenCounter() und offenNames() wird zudem das Übersichtsfeld zum Erfassungstatus aktualisiert.

Diese Methode wird beim Laden, beim Datensatzwechsel und nach jedem Wechsel bei irgendwelchem Detailknopf aufgerufen

+ offenCounter() as Integer

Zählt wie viele Teile noch offen sind und gibt diesen Wert zurück

+ offenNames() as String

Gibt als String alle Teile zurück, welche noch offen sind.

+ RecordSave()

Da Ms Access kein commit kennt, wurde dieser Umstand mit dieser Methode behoben

#### Formular1, Formular2, Formular3, Formular4

+ Form\_Load()

Sucht die mit Openargs mitgegebene Datensatznummer und lädt diese dann. Falls kein Openargs-Wert vorhanden ist, wird der 1. Datensatz geladen.

Die Methode schliesst zudem alle anderen FormularX-Formulare

+ goto2Part\_Click() bzw. Goto3Part\_Click() etc.

Lädt das jeweils gewählte Formular und gibt dabei die aktuelle Datensatznummer als Openargs-Wert mit.

#### Exportliste

+ changeSortierung\_Click()

Wechselt den Sortierreihenfolge der Anzeige von Datenbankreihenfolge in alphabetischer Reihenfolge und wieder zurück.

- close\_Click()

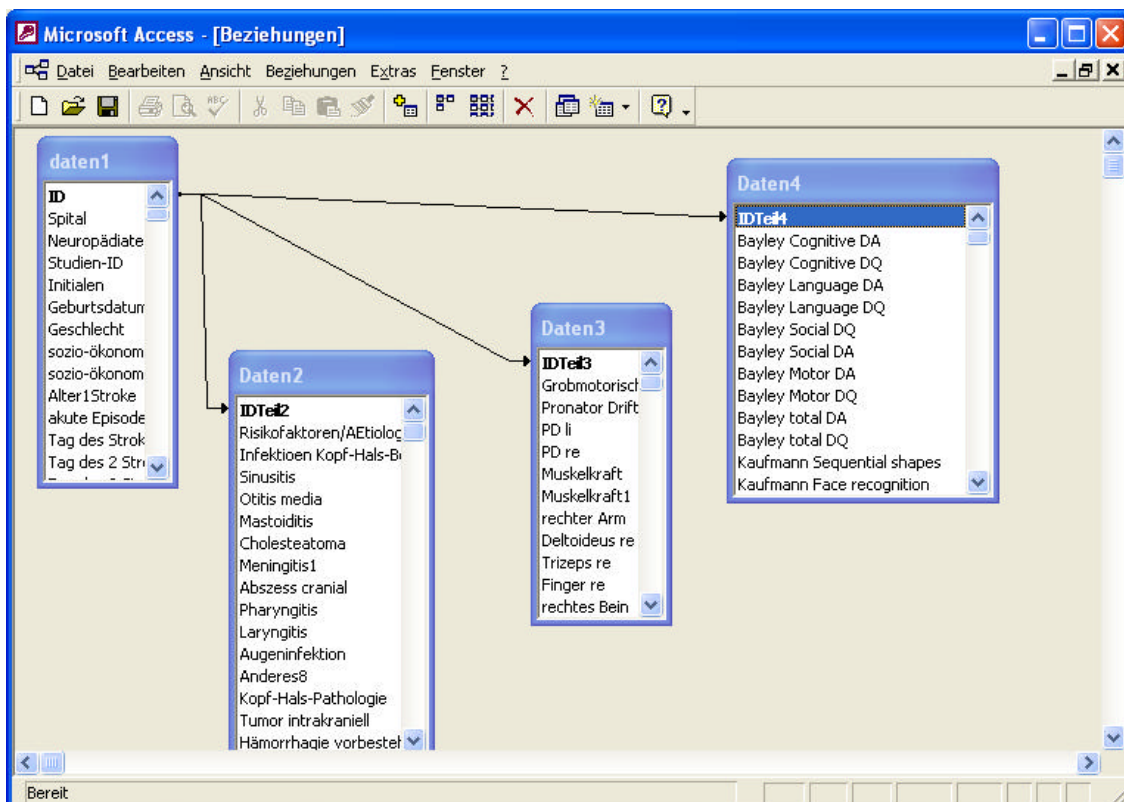
Speichert die Auswahl als Teilquery in die Tabelle Teile und schliesst danach das Formular

- Form\_Load()

Lädt das Formular und lädt die mit Openargs mitgegebene Teilquery, sodass danach die Auswahlstruktur wieder rekonstruiert werden kann.

## 4. ER-Diagramm

Wie man hier sieht, ist die Beziehungsstruktur sehr einfach. Alle restlichen Tabellen sind höchstens adHoc-Verknüpft (Daten1Strings und Daten2Strings) oder ganz unverknüpft (=alle anderen)



## 5. Ablaufdiagramme

### 5.1 Excelexport

Startseite	Teile	Exportliste	Fieldscounter	Benutzeraktionen
				klickt auf Excel Export
Lädt Excel Export				
				will ein bereits bestehende Gruppe modifizieren
→	Lädt Teile bearbeiten (Übergabe von der Gewählten Gruppe an: Exportliste			
	→	Lade Queryliste von der Tabelle Teile.		
		→	Bestimme mit FieldsCounter(query) Anzahl vorhandene Felder	
		Setze alle Export-Felder auf False		
		→	Gebe Feldname des 1. Feldes zurück GetFields(nr)	
		Suche das entsprechende Feld und setze das Export-Feld auf True		
		Dito alle weiteren Felder		
				ändert Exportfelder....
				schliesst das Formular
		Bildet mit allen gewählten Exportfelder einen Teilquery und speichert es in der Tabelle Teile ab.		
				Benutzer definiert die Obergruppen und klickt anschliessend auf Daten ins Excel exportieren
	Testet ob nicht zuviel Felder gewählt sind. Wenn nein: Führe Methode Excelexport aus			

Excelexport()	SaveQuery(strSQL As String)	TransferToExcel()	DateiSpeichern(.....)	Benutzeraktionen
Bilde den vollständigen Query				
→	Speicher den Query als Abfrage ab			
→		Bestimme Standardexportpfad aus t_Pfad und lade DateiSpeichern		
		→	Offne Speicherfenster	
				Bestimmt den Filenamen und Speicherort
		Führt den Export aus		

## 5.2 Laden einer Abfrage als Bericht

Bsp. Laden von „Abfrage Aetiologie“ Nur Longtermdaten

Abfrage_Anzahl_Filter_Click()	nurLongtermabfragen(berichtname)	setFilter(value)	GetReport(stDocName, whereCondition)	Benutzeraktionen
				Klick auf Abfrage Aetiologie, nur Longterm.
Startet: nurLongtermabfragen("Abfrage Aetiologie")				
→	Startet setFilter(true)			
	→	Setzt das Steuerfeld in der Steuerdaten-Tabelle auf true		
	Startet: booleanToString()			

Siehe Einschub ganz unten

	Startet: GetReport und gibt den dort def. Longtermfilter und den Namen des gewünschten Berichtes mit.			
	→		Oeffnet Bericht(Abfrage Aetiologie) mit Longtermfilter	

„Berichtloader“	Familienanamnese(ID)	RisikofaktorenVorStroke(ID)	RisikofaktorenBeiStroke(ID)
Beginne beim 1. Datensatz Lade: Familienanamnese([ID])			
→	Gib die Familienanamnese als String zurück		
Lade: RisikofaktorenVorStroke([ID])			
→		Gib die RisikofaktorenVorStroke als String zurück	
Lade: RisikofaktorenBeiStroke([ID])			
→			Gib die RisikofaktorenBeiStroke als String zurück
Hole die restlichen Infos aus der Tabelle schreibe diese Werte und gehe zum nächsten Datensatz			

Etc.

booleanToString()	booleanToString1()	booleanToString2()
Erstellt mit Hilfe von getMaxID(dbName) gegebenenfalls neue Recordsets für Daten1Strings und Daten2Strings		
Startet: booleanToString1()		
→	„Uebersetzt“ alle Booleans von Daten1 in Strings nach Daten1Strings	
Startet: booleanToString2()		
→		„Uebersetzt“ alle Booleans von Daten2 in Strings nach Daten2Strings

# Benutzerdokumentation Stroke Datenbank

## 1. Inhaltsverzeichnis

Benutzerdokumentation Stroke Datenbank.....	1
1. Inhaltsverzeichnis .....	1
2. Ziel und Zweck des Programmes.....	1
3. Installationshinweise.....	1
8. Fertig.....	2
4. Dateneingabe .....	2
4.1 Allgemeines .....	2
4.2 „Offen“-Felder.....	3
5. Standardauswertungen .....	3
6. Individuelle Excelexporte.....	3
6.1 Allgemeine Anleitung .....	3
6.2 Technische Hinweise.....	3
7. Kontaktadresse.....	4

## 2. Ziel und Zweck des Programmes

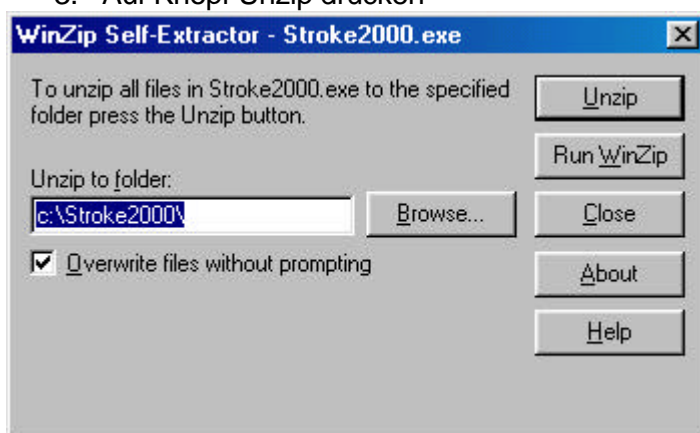
Das Programm dient der Datenerfassung und Auswertung von Daten die bei der Studie über Stroke bei Kinder anfallen.

## 3. Installationshinweise

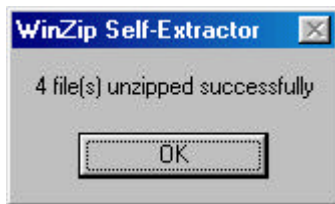
Das Programm benötigt unbedingt eine normale Version von MS Access 2000. Mit neueren Accessversionen sollte es auch laufen, doch das Programm wurde nur mit Access 2000 getestet.

Installationsanleitung:

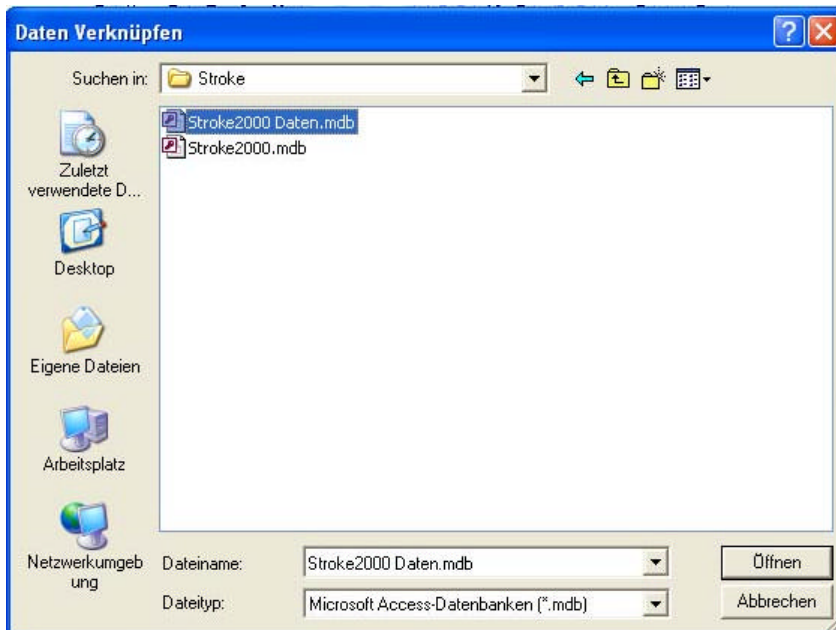
1. Installationsdatei Stroke2000.exe starten.
2. Bei Bedarf Verzeichnis ändern (WARNUNG: Alle allfälligen Dateien werden dort überschrieben)
3. Auf Knopf Unzip drücken



4. Auf Knopf OK drücken



5. Auf Knopf Close drücken (Gleiches Fenster wie bei Knopf Unzip)
6. Im Verzeichnis wo man gewählt hat, das Programm mittels Stroke2000.mdb starten
7. Im Verzeichnis wo man gewählt hat, das Programm mittels Stroke2000 Daten.mdb verknüpfen



8. Fertig
9. Falls sie wollen, können sie noch eine Verknüpfung mit der Startdatei Stroke2000.mdb erstellen. Sollten sie es Standardverzeichnis c:\stroke2000 gewählt haben, so können sie einfach den sich dort befindlichen Link dazu verwenden.

Es empfiehlt sich jetzt sich eine aktuelle Version der Daten zu besorgen. Es reicht hier, dass sie dabei nur die Datei Stroke2000 Daten.mdb ersetzen (Access zuvor schliessen)

## 4. Dateneingabe

### 4.1 Allgemeines

Da es sich um eine Access-Datenbank mit einem auf Access aufgebauten GUI (Graphics User Interface) handelt verzichte ich hier auf eine allgemeine Anleitung zur Dateneingabe. Falls sie sich unsicher fühlen, konsultieren sie bitte ein Accessbuch für Benutzer.

Die ersten Benutzerrückmeldungen haben gezeigt, dass vor allem die Detailsfenster zu im ersten Moment zu Verwirrung geführt haben. Sobald sie aber die Idee dabei gesehen haben, war es anschliessend kein Problem mehr. Im Grunde genommen handelt es sich bei den Detailsknöpfen um Wenn/Dann-Abfragen. Bsp: Ein Kind hat „Sys Zeichen und Symptome“. Man klickt also auf das Häkchen und es erscheint dann automatisch das Pop-Up-Fenster mit den Details. Falls das Kind keine „Sys Zeichen und Symptome“ hat, braucht man sich ja auch nicht um diese Details zu kümmern.

Passt aber auf, dass ihr nicht versehentlich diese Häkchen setzt oder entfernt. Dies ist schnell passiert.



## 4.2 „Offen“-Felder

Sinn und Zweck der „offen“-Felder ist folgendes:

Praktische Erfahrungen haben gezeigt, dass die einzelnen Teile der Patientendaten häufig nicht alle zusammen eintreffen. Damit man diese Daten dennoch bereits erfassen kann wurden diese „offen“-Felder erstellt. Wann immer noch irgendwo ein Teil fehlt kann man das Feld anklicken, damit man später noch weiss, was man noch nicht erfasst ist. Eine Auflistung der offenen Teile wird dann automatisch auf der 1. Seite ganz oben abgebildet

atric Neurology (Bei einem neuen Datensatz immer zuerst die Studien-ID eingeben!!!!)

Bemerkungen zur Erfassung: Anz. offene Teile: 2 (Koagulopathien, Koagulopathien Labor)

Import aus alter DB

Generell gilt:  
 = Ja, bzw. ist Vorhanden

Geburtsdatum: 30.08.2000  
Geschlecht:  männlich  weiblich  
Herkunft: caucasian

ID	ID Teil 2	ID Teil 3	ID Teil 4
23	23	23	23

## 5. Standardauswertungen

Auf der Hauptseite finden sie eine Reihe von Standardauswertungen. Diese werden jedes Mal neu berechnet. Das heisst, sämtliche Änderungen und Erweiterungen der Daten werden jedes Mal neu berücksichtigt (=dynamische Abfragen).

## 6. Individuelle Excelexporte

### 6.1 Allgemeine Anleitung

Falls sie eine spezielle Auswertung benötigen, die das Programm direkt nicht kann, haben sie die Möglichkeit die dafür benötigten Daten in ein Excel-Liste zu exportieren. Um das zu erreichen müssen sie folgendes machen:

1. Programm wie gewohnt starten (falls nicht bereits geschehen)
2. In der Hauptansicht den Knopf „Excelexport“ drücken.
3. Jetzt sehen sie eine Liste von bereits definierten Datengruppen. Durch das Auswählen der Gruppen mittels dem Häkchen können sie eine neue Übergruppe zusammenstellen. Falls sie mit den Teilgruppen nicht zufrieden sind können sie eine bestehende verändern oder auch gerade eine neue Definieren.
4. Sobald sie alles haben, können sie ihre Übergruppe mit dem Knopf „Daten ins Excel exportieren“ exportieren.
5. Das Programm fragt dann noch, wo sie es speichern wollen und wie die Datei heissen soll.
6. Fertig

### 6.2 Technische Hinweise

- Er führt zu einen Fehler, wenn sie zweimal in die gleiche Datei exportieren möchten, solange sie diese mit MS Excel gleichzeitig noch geöffnet haben.
- Zudem ist es nicht möglich mehr als 255 Felder gleichzeitig zu exportieren.
- Falls die Datei, in welche sie den Export machen wollen, schon existiert, so wird diese ohne Warnung überschrieben.

## **7. Kontaktadresse**

Falls sie noch Fragen zum Programm haben oder einen Bug entdeckt haben, können sie sich an mich wenden. Meine Adresse lautet:

Reto Zurbuchen  
Gartenstr. 8  
2558 Aegerten  
076 / 532 51 91  
rzurbuchen@swissonline.ch

# Fazit zum Projekt „Stroke Datenbank“

## **Positiv:**

### **Positive Aspekte betreffend Organisation**

- Sehr gute Zusammenarbeit zwischen den einzelnen Projektmitarbeiter. Es machte nicht nur die Arbeit viel einfacher, sondern man lernte sehr viel vom anderen.
- Verlässlichkeit der Projektteilnehmer

### **Positive Aspekte betreffend Entwicklung der Software**

- Es gab kaum Ideen von den Benutzer die ich nicht umsetzen konnte.
- Obschon ich am Anfang befürchtete, dass das Projekt ein unendliches erstellen von langweiligen und immer ähnlich ablaufenden Abfragen ist, wurde es im Schluss auch noch technisch anspruchsvoll. Der Excelexport (=kleiner Parser) mit der DLL-Anbindung war in dieser Sicht das Highlight des Projektes

## **Negativ:**

### **Aufgetauchte Probleme betreffend Organisation**

- Das Projektende verzögerte sich am Schluss um 1 Woche. Grund: Ferienabwesenheiten diverser Personen. Dies kann man zwar nicht ändern, aber durch die Einbeziehung dieser Tatsache in der Planung hätte man diese Verzögerung ev. vermeiden können indem man wichtige Sitzungen vorverlegt hätte.
- Das Konzept der „Demo Days“ konnte nur bedingt realisiert werden. Grund: Da ich laufend den Benutzer Updates zur Verfügung stellte, kamen auch laufend wieder Verbesserungswünsche zurück. Das bewusste Treffen für die Abnahme der einzelnen Teile wurde deshalb fast zu einer Alibiübung, da der Benutzer ja die neuesten Tools bereits kannte und einsetzte.
- Es ist sehr schwierig bei so grosser Zusammenarbeit, eine korrekte Stundenabrechnung zu machen. Ich kann schliesslich nicht alle tel. aufschreiben.

### **Aufgetauchte Probleme betreffend Entwicklung Software**

- Gegeben war folgende Aufgabenstellung: Erstelle 9 klar def. Abfragen als Bericht. Pro Bericht waren es rund 100 Felder die berücksichtigt werden müssen. Bei den Booleanfelder drucke nur diejenigen aus die true sind. Kurz: Ich musste eigentlich hunderte „if true then print ...“ Codezeilen schreiben. Da viele Felder aber mehrmals benötigt werden, war ich sehr schnell zum Schluss gekommen, das ich das einmal zentral lösen muss. D.h. ich muss jeweils temporäre Tabellen erstellen die nur Strings haben, damit ich einfach die einzelnen Gruppen zusammenstellen kann (→ tb!spalteA & tb!spalteB & etc.)  
Soweit so gut: Wie sieht nun diese Übersetzungsmethode aus? Ich fand damals keine bessere Lösung als alle einzeln mit if .. then zu übersetzen. Grund: Ich konnte ja schon jeweils den Feldnamen auslesen (obwohl sehr mühsam), aber dieser ist nicht immer sehr praktisch. Bsp: Bei den Manifestationen. Es hat ca. 40 Felder mit Manifestationen. Also heissen diese Felder immer etwasManifestation. Wenn ich diese Felder jetzt brauchen würde, hätte ich beim Ausdruck bis zu 40 Mal das Wort

Manifestation. So konnte ich jetzt aber dies jeweils löschen und beim Ausdruck einfach einmal Manifestationen: einfügen. Das konnte sich dann schon eher sehen lassen.

Warum schrieb ich damals: Heute kam mir die Idee, dass man das ganze noch mit einer zusätzlichen Wrappertabelle lösen konnte. Um diese aber zu erstellen braucht es sicherlich weitere rund 10 Stunden. Selbst das Auslesen der Feldnamen ist leider nicht ganz einfach.

## ***Allgemeiner Kommentar zu MS Access***

Die Vorteile welche für Access sprechen liegen auf der Hand. Ich kenne keine Entwicklungsumgebung indem man schneller etwas wie diese Aufgabenstellung realisiert. Gute Entwickler-GUIs, brauchbare Assistenten, einfache Sprache, „kein“ Compilieren

Leider gibt es auch Nachteile die für eine andere Sprache sprechen

- Solange man keine sehr teure Developerlizenz hat, braucht jeder Benutzer eine Access-Lizenz
- Stabilität von Access lässt manchmal zu wünschen übrig
- Tests erstellen ist fast ein Ding der Unmöglichkeit.
- Kein CVS oder ähnliches.
- Beschränkte Einsatzmöglichkeit von Access.
- Falls die Daten auf ein Netzlaufwerk liegen und es sich hierbei um keinen SQL-Server handelt, ist der Traffic gigantisch.
- MS Access unterstützt zwar OO-Programmierung, aber dies macht selten Sinn.

## Appendix

Liste aller zu erstellenden Berichten:

- Anzahl der Untersuchten Kinder in einer Grobübersicht
- Ü. (= Übersicht) der Resultate der 1. Untersuchungen
- Abfrage Kuogulopathie
- Abfrage Vaskulopathie
- Abfrage Infektiologie
- Abfrage SVT
- Abfrage Outcome
- Abfrage Aetiologie
- Abfrage Klinische Befunde
- Abfrage Zusätze → Resultate IQ-Tests etc.
- Ü. Outcome – Infarkt \*)
- Ü. Outcome – Kaliber und Seite \*)
- Ü. Outcome – Risikofaktoren und Ursache \*)
- Ü. Outcome – Infekt \*)
- Ü. Outcome – Betroffene Regionen \*)
- Ü. Outcome – Gerinnungsprobleme \*)

\*) hier wurden je ein Bericht auf Grundlage der IQ-Tests als auch der neurologischen Untersuchungen erstellt

Bei den meisten Berichten kann der Benutzer zudem wählen, ob alle Patienten ausgewertet werden oder nur von denen, bei der eine „Longterm“-Untersuchung vorliegt.