

Dokumentation
zu

Informatik Projekt von

Pani M. Tetik, 1997

Datenbank für
„Clinical Study Notification Forms“ (BAG/ISPM)

Auftrag / Betreuung:

Software Composition Group - Universität Bern

Prof. Dr. O. Nierstrasz
Dr. Matthias P. Günter
Fr. Nicole Portmann

Inhaltsverzeichnis

1 Übersicht	3
1.1 Projektziel	3
1.2 Entwicklungsstand des Projekts	3
1.3 Vom BAG nachträglich gewünschte Erweiterungen	3
1.4 Involvierte Personen	3
2 Benutzerinstruktion, Bedienungsanleitung	4
3 Verzeichnisstruktur der Software „SNF“	5
4 Technische Beschreibung	6
4.1 Problemanalyse	6
4.1.1 Problemstellung	6
4.1.2 Ist-Analyse	6
4.1.3 Sollkonzept	6
4.2 Aufgabendefinition	7
4.2.1 Funktionalität der Software	7
4.2.2 Benutzerschnittstelle	7
4.2.3 Dokumentation	7
4.2.4 Systembasis	7
4.2.5 Anforderungen an die Tests	7
4.2.6 Gliederung des Projekts	7
4.3 Technische Realisierung	8
4.3.1 Designspezifikation	8
4.3.2 Klassendesign	14
4.3.3 Objekthierarchie, Objekte und ihre Eigenschaften und Methoden	14
4.3.4 Konsistenztests	28
4.4 Mögliche Erweiterungen (Dynamisches Pflichtenheft)	34
4.5 Erfahrungen, Danksagung	35
5 Anhang	36
5.1 Pflichtenheft	36
5.1.1 Dokumentation	36
5.1.2 Software	36
5.2 Abklärung der einzusetzenden Entwicklungsumgebung für die Datenbankprojekte „Impfstoffnebenwirkungen“ und „Notifikationsformulare“	39
5.3 Hilfesystem der Software	40
5.4 Screen shots	45
5.5 Beispiele für Programmcode der Software „SNF“	53
5.5.1 Methode click() der Befehlsschaltfläche <i>cmd_forw</i> im Formularobjekt <i>auswert.scx</i>	53
5.5.2 Methode click() der Befehlsschaltfläche <i>cmd_such</i> im Formularobjekt <i>suchen.scx</i>	57
6 Literaturverzeichnis	61

1 Übersicht

1.1 Projektziel

Das Ziel des Projektes ist die Erstellung einer Datenbank, welche die Handhabung der Klinischen Studiennotifikationsformulare des BAG (Clinical Study Notification Forms) mit Hilfe der EDV erleichtert.

Dazu wird das Programm „SNF“ erstellt, welches auf Rechnern (PCs) mit Betriebssystem Windows NT 4, Windows 95 oder Windows 3.x lauffähig ist.

1.2 Entwicklungsstand des Projekts

Folgende Teile der Software wurden mit Anlehnung an die Spezifikation realisiert:

- Festlegung der Datenbankstruktur
- Festlegung der Verzeichnisstruktur für die Dateien des Projekts
- Definition der Datenbanktabellen (Felddefinitionen, Abhängigkeiten)
- Form der Interaktion des Programms mit dem Anwender
- Elemente der grafischen Benutzeroberfläche (GUI)
- Abfolge der Bildschirmmasken
- Eingabe von neuen Studien
- Editieren von erfassten Studiendaten
- Import von Daten, die mittels Email geliefert werden
- Tests auf Konsistenz und Vollständigkeit der eingegebenen Daten
- Speicherung der getesteten Daten in den entsprechenden Tabellen
- Datenbankabfragen nach bestimmten Kriterien
- Drucken verschiedener Typen von Auswertungen
- Benutzeranleitung in Form einer kontextsensitiven Hilfe im Programm (wurde in der Spezifikation nicht verlangt, aber trotzdem implementiert)

1.3 Vom BAG nachträglich gewünschte Erweiterungen

Ausdrucken von Studiendaten in der Form der derzeit verwendeten, sechsseitigen Studiennotifikationsformularen. Diese Funktionalität wurde noch nicht implementiert.

1.4 Involvierte Personen

Prof. O. Nierstrasz	Software Composition Group, IAM, Universität Bern
Dr. M. P. Günter	Betreuung des Projektes, Assistenz von Prof. O. Nierstrasz
Fr. N. Portmann	Betreuung des Projektes, Abklärung des Datenimports via Email
Dr. Th. Weber	Benutzer der Software im BAG
Hr. P. M. Tetik	Realisierung des Projektes

2 Benutzerinstruktion, Bedienungsanleitung

Die Benutzerinstruktion für die Installation der Software ist hinfällig: Es wird ein Satz von 3.5-Zoll Disketten mit allen notwendigen Dateien (inklusive Installationsprogramm) ausgeliefert.

Die Bedienungsanleitung für die Software liegt in der Form einer kontextsensitiven Online-Hilfe vor. Zusätzlich wird noch ein Hilfetext als MSWord-Dokument (WinWord 6) in das Hauptverzeichnis (C:\SNF\) des Programms „SNF“ unter dem Namen „snfhelp.doc“ abgelegt.

3 Verzeichnisstruktur der Software „SNF“

Das Hauptverzeichnis der Software wird zum Beispiel auf dem Laufwerk C: erstellt und heisst SNF.

Darin enthalten sind folgende Unterverzeichnisse:

Verzeichnis	Inhalt
• Classes	Klassendefinitionen
• Dbfs	Tabellen der Datenbank „snfdb1“ und die freien Tabellen „errors.dbf“ und „snfhelp.dbf“ (inklusive ihre zugehörigen Indextabellen)
• emptyDBs	Kopien der Dateien aus dem Verzeichnis Dbfs jedoch ohne Datensätze (ausser die oben erwähnten freien Tabellen)
• Forms	Definitionen der Formular-Objekte für die grafische Oberfläche und ihre zugehörigen Dateien
• Picts	Für die grafische Oberfläche verwendete Bilder (icons)
• Prgs	Programmdateien
• Reports	Definitionen der Bericht-Objekte für den Ausdruck der Auswertungen
• Imports	MS-Excel-Dateien, die per Email geliefert wurden
• Temp	Temporäre Datenbank und ihre Tabellen

Im weiteren sind folgende Dateien enthalten:

Dateiname	Funktion
Config.fpw, Foxuser.fxp, Foxuser.fpt	Konfigurationsdateien für die FoxPro-Umgebung
snf.exe	Das Programm SNF
snfdb1.*	Dateien der Datenbankdefinition
snf02.*	Projektdateien
snf_help.doc	Hilfetext zur Software
pkzip.exe, pkunzip.exe	Programme für die Archivierung
Olddb.ZIP	Archivkopie des Datenbestandes vor der letzten Neuerstellung der Datenbank

4 Technische Beschreibung

4.1 Problemanalyse

4.1.1 Problemstellung

Das Institut für Sozial- und Präventivmedizin soll eine Datenbankapplikation für die elektronische Verwaltung und Auswertung der „Clinical Study Notification Forms“ für Blutprodukte, Impfstoffe und Gentherapien realisieren.

Die Software soll unter anderem die Archivierung der Daten erleichtern.

4.1.2 Ist-Analyse

Zur Zeit werden drei einander ähnliche Notifikationsformulare zu den einzelnen Studienobjekttypen (Impfstoffe, Blutprodukte und Gentherapien) in Papierform verwendet.

Dabei werden relativ grosse Datenmengen pro Studie erfasst.

Deren Handhabung (Archivieren, Erstellen von Auswertungen, Suchen nach bestimmten Studien) soll jetzt mit Hilfe des Computers erleichtert werden.

Für das BAG wurde letztes Jahr ein Datenbankprogramm für Impfstoffnebenwirkungen von Frau N. Portmann realisiert.

4.1.3 Sollkonzept

4.1.3.a Übersicht über die geplante Lösung

Als Resultat des Projekts ist ein Computerprogramm vorgesehen, das auf den gängigen Betriebssystemen (MSWindows95, MSWindowsNT) einsetzbar ist und die nachfolgend beschriebenen Anforderungen des BAG erfüllt.

Des weiteren soll während der Implementierung versucht werden, das Programm so wartungs- und erweiterungsfreundlich wie möglich zu gestalten.

In diesem Zusammenhang wurde, abweichend vom Grobkonzept, die Realisierung mit Hilfe des Datenbanksystems MS-Visual FoxPro 3.0, anstelle von FoxPro 2.6, beschlossen (vgl. Anhang: Kapitel 5.2, Seite 39)

4.1.3.b Rahmenbedingungen

- Das Datenbankprogramm wird auf einem Compaq Rechner mit Pentium 133 Prozessor eingesetzt werden (Einzelplatzlösung).
- Für den Datenbackup werden Disketten, Optical Disks oder Tapes eingesetzt.
- Als Datenbanksystem ist MS-Visual FoxPro 3.0 vorgesehen.

4.1.3.c Anforderungskatalog

(s. auch Anhang: Kapitel 5.1, Seite 36)

- Programm für die Automatisierung der Erfassung, der Auswertung und der Archivierung der Notifikationsformulare (Study Notification Forms) und für die Vereinfachung der Suche nach Informationen in den erfassten Daten
- Benutzerhandbuch und Dokumentation zum Programm
- Schulung der Benutzer

4.2 *Aufgabendefinition*

4.2.1 Funktionalität der Software

- Die Daten der Studien müssen manuell und über Email ins System aufgenommen werden können.
- Die Daten der erfassten Studien müssen mutiert werden können.
- Es müssen auf einfache Art Auswertungen über die erfassten Studien erstellt und gedruckt werden können.
- Die Möglichkeit, bestimmte Informationen in der Datenbank der Studien zu finden, muss gegeben sein.
- Archivierung der Daten (zum Beispiel am Ende eines Jahres).

Im Anhang kann das von Hr. Dr. Th. Weber (BAG) am 21. Februar 1997 bestätigte Pflichtenheft nachgelesen werden.

4.2.2 Benutzerschnittstelle

Die Benutzerschnittstelle soll so weit wie möglich derjenigen des Projektes „Impfstoffnebenwirkungen“ von Frau N. Portmann entsprechen.

4.2.3 Dokumentation

- Projektdokumentation (dieses Dokument)
- Benutzeranleitung (Datei: snf_help.doc)

4.2.4 Systembasis

- Computersystem mit INTEL Pentium133 Prozessor
- WindowsNT 4.0
- Netzwerkdrucker (Laserdrucker von Hewlett Packard)

4.2.5 Anforderungen an die Tests

In den Tests muss die Korrektheit der implementierten Funktionen geprüft werden.

Vor allem aber muss das Speichern der Daten in den richtigen Tabellen der Datenbank überprüft werden.

Vom Endbenutzer wird erwartet, dass er in einer ersten Phase des Einsatzes der Software die Erfüllung der gewünschten Funktionen testet.

4.2.6 Gliederung des Projekts

Das Projekt kann entsprechend den verlangten Funktionen gegliedert werden in

- Entwurf der Datenbank (Entity-Relationship-Model)
- Manuelle Dateneingabe
- Editieren von erfassten Daten
- Datenimport aus Email
- Suchen nach Studien (für den Benutzer transparente Datenbankabfragen mit SQL)
- Erstellen von Auswertungen (für den Benutzer transparente Datenbankabfragen mit SQL)
- Entwurf der grafischen Benutzeroberfläche
- Archivierung und Wiederherstellung der Daten, Erstellen einer neuen Datenbank
- Benutzerführung mit Warn- und Fehlermeldungen

4.3 Technische Realisierung

4.3.1 Designspezifikation

4.3.1.a Spezifikation der Datenbasis

Verwendete Abkürzungen für die Datentypen der Tabellenfelder und ihre Bedeutung (Legende):

- C: Alphanumerisches Zeichen
- N: Numerisches Zeichen
- L: Logikwert
- D: Datumswert
- MEMO: MEMO-Feld

Die Zahl hinter der Typenbezeichnung gibt die Länge des Feldes an.

Tabellenname: STUDY
Primärschlüssel: study_ind (ref_nbr)

Feldname	Typ
ref_nbr	C7
std_type	C2
dt_code	C50
dt_phase	NI
dt_titel	MEMO
ds_pros	L
ds_retro	L
ds_contr	L
ds_open	L
ds_blind	L
ds_dbblind	L
ds_rando	L
ds_parall	L
ds_cross	L
ds_multic	L
ds_nbrc	N3
ds_othert	C50
ds_startm	N2
ds_starty	N4
ds_endm	N2
ds_endy	N4

Tabellenname: SPONSOR
Primärschlüssel: spons_ind (sp_id)

Feldname	Typ
sp_id	C8
sp_comp	C50

Tabellenname: MANUFACT
Primärschlüssel: manufa_ind (m_id)

Feldname	Typ
m_id	C8
m_comp	C50

Tabellenname: CRO
Primärschlüssel: cro_ind (cro_id)

Feldname	Typ
cro_id	C8
cro_comp	C50

Tabellenname: PERSON
Primärschlüssel: person_ind (p_id)
Fremdschlüssel: spons_ind (sp_id)

Feldname	Typ
p_id	C8
p_name	C100
sp_id	C8

Tabellenname: HOSPITAL
Primärschlüssel: hosp_ind (h_id)

Feldname	Typ
h_id	C8
h_name	C70
h_divis	C50

Tabellenname: INVESTIG
Primärschlüssel: invest_ind (i_id)
Fremdschlüssel: hosp_ind (h_id)

Feldname	Typ
i_id	C8
i_name	C100
i_divis	C100
h_id	C8

Tabellenname: CO_INV
Primärschlüssel: co_inv_ind (co_id)
Fremdschlüssel: hos_ind (h_id)

Feldname	Typ
co_id	C8
co_name	C100
co_divis	C100
h_id	C8

Tabellenname: ERC
Primärschlüssel: erc_ind (e_id)
Fremdschlüssel: hosp_ind (h_id)

Feldname	Typ
e_id	C8
e_name	C100
chair_id	C8
stat_id	C8
h_id	C8

Tabellenname: ERCMEMB
Primärschlüssel: ercmem_ind (mem_id)
Fremdschlüssel: erc_ind (e_id)

Feldname	Typ
mem_id	C8
mem_name	C100
mem_prof	C50
mem_sex	C1
mem_chair	L
mem_stat	L
e_id	C8

Tabellenname: DECISION
Fremdschlüssel: erc_ind (e_id)

Feldname	Typ
dec_date	D8
dec_deci	N1
dec_approv	MEMO
e_id	C8

Tabellenname: PHONE
Fremdschlüssel: phon_ind (id)

Feldname	Typ
id	C8
nbr	C20
direct	L

Tabellenname: FAX
Fremdschlüssel: fax_ind (id)

Feldname	Typ
id	C8
nbr	C20
direct	L

Tabellenname: ADDRESS
Fremdschlüssel: addr_ind (id)

Feldname	Typ
id	C8
ref_nbr	C7
addr1	C80
addr2	C80
addr3	C80
state	C50

Tabellenname: STD_CRO

Feldname	Typ
ref_nbr	C7
cro_id	C8

Tabellenname: GT

Primärschlüssel: gen_ind (id)

Fremdschlüssel: std_ind (ref_nbr)

Feldname	Typ
id	N7
ref_nbr	C7
study_flag	L
trdname	C100
codenbr	C20
bagnbr	C20
lotnbr	C50
spec	MEMO
sc	L
im	L
iv	L
po	L
rect	L
intranas	L
other	L
otherappl	C20
save_prec	MEMO
prodcomp	MEMO
vector	MEMO

Tabellenname: VACCINE

Primärschlüssel: vacc_ind (id)

Fremdschlüssel: std_ind (ref_nbr)

Feldname	Typ
id	N7
ref_nbr	C7
study_flag	L
trdname	C100
codenbr	C20
bagnbr	C20
lotnbr	C50
sc	L
im	L
iv	L
po	L
rect	L
intranas	L
other	L
otherappl	C20

Tabellenname: STD_HOSP

Feldname	Typ
ref_nbr	C7
h_id	C8

Tabellenname: BP

Primärschlüssel: bp_ind (id)

Fremdschlüssel: std_id (ref_nbr)

Feldname	Typ
id	N7
ref_nbr	C7
study_flag	L
trdname	C100
codenbr	C20
bagnbr	C20
lotnbr	C50
type	N3
sc	L
im	L
iv	L
other	L
otherappl	C3
adjuvants	MEMO
vimeth	MEMO
actcomp	MEMO

Tabellenname: ACTCOMPV

Primärschlüssel: comp_ind (comp_id)

Fremdschlüssel: vacc_ind (id)

Feldname	Typ
comp_id	N9
id	N7
ac_name	C50
ac_state	N1
ac_type	N1
ac_otype	C50
ac_recomb	N1

Tabellenname: STD_SPON

Feldname	Typ
ref_nbr	C7
sp_id	C8

Tabellenname: STD_MANU

Feldname	Typ
ref_nbr	C7
m_id	C8

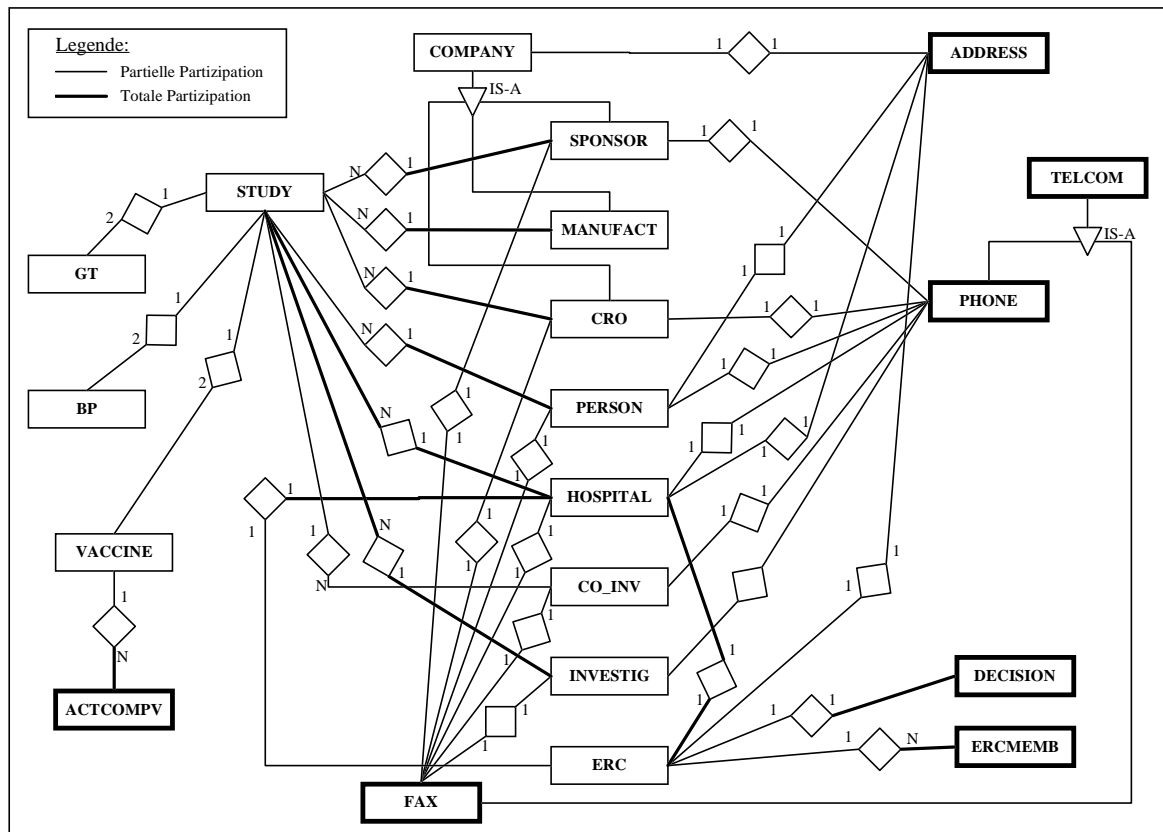
Tabellenname: STD_PERS

Feldname	Typ
ref_nbr	C7
p_id	C8

4.3.1.b Entity-Relationship-Diagram

Erläuterung zu den abgekürzten Entitätsnamen:

- GT Gene Therapy
- BP Blood Product
- ACTCOMPV Active Component of Vaccine
- CRO Clinical Research Organisation
- ERC Ethical Review Committee
- CO_INV Co-Investigator



4.3.1.c Spezifikation der Dokumentation

Das Benutzerhandbuch wird als integriertes, kontextsensitives Hilfesystem implementiert. Der Benutzer kann mit der Taste <F1> die Hilfe aufrufen. Der Text wird aber auch als MSWord-Text im Verzeichnisbaum des Projekts abgelegt (s. Anhang: Kapitel 5.3, Seite 40).

4.3.1.d Spezifikation der Systembasis

- Betriebssystem MSWindows95 oder MSWindowsNT (ab Version 3.51)
- Monitor mit eingestellter Auflösung von 800x600 oder höher
- Sinnvolle Leistung ab INTEL Pentium 133
- Gewünschtes Backup-System (Disketten, MO, ZIP, JAZ)

4.3.1.e Spezifikation der Benutzeroberfläche

Damit das Programm so benutzerfreundlich wie möglich bedient werden kann, wird versucht, das Aussehen der Bildschirmmasken für die Eingabe der Daten entsprechend den sechs Seiten der bestehenden Formulare zu gestalten.

Abweichend vom GUI des Projektes von Fr. Portmann wird darauf geachtet, dass die Benutzer, welche die Notifikationsformulare des BAG kennen, das Programm auch ohne ausführliche Schulung benutzen können.

Um eine intuitive Bedienung des Programms zu gewährleisten, sind Befehlsschaltflächen vorgesehen, die auf jeder Eingabemaske bei gleicher Beschriftung und Position gleiche Aktionen hervorrufen.

Die Benutzer werden relativ starr durch das Programm geführt:

- Es gibt immer nur ein aktives Fenster
- Ein Fenster kann nur über eines der Befehlsschaltflächen verlassen werden (→ sogenannte „modale Fenster“)
- Es gibt keine Menüs am oberen Rand der Fenster

Fehlermeldungen werden durchwegs mit einem roten, Warnmeldungen mit einem gelben Hinweisfenster angezeigt.

Warnmeldungen können auf zwei Arten quittiert werden:

- Wenn die Schaltfläche „Zurück“ geklickt wird, springt die Einfügemarke in das (Text-) Eingabefeld, wo die fehlenden Informationen eingegeben werden können
- Falls die Schaltfläche „Schon gut“ gewählt wird, wird das in der Aktivierungsfolge nächste Feld angesprungen.

Fehlermeldungen enthalten nur die Schaltfläche „OK“, welches nach dem Lesen und verstehen der Nachricht betätigt werden sollte.

Auf der ersten Seite (Formularobjekt: *main_fcts.scx*) wird die gewünschte Funktion ausgewählt:

- **Neue** Studie erfassen
- Vorhandene Studien **mutieren/editieren**
- Studien **suchen**
- **Import** von Daten aus E-Mail-Attachments
- **Auswertungen**
- **Administration** (Archivieren, Wiederherstellen, ...)
- **Beenden**

Auf den Bildschirmmasken der Dateneingabe sind folgende, *gemeinsame* Bedienelemente vorgesehen:

Button	Aktion(en)
Zurück	<ul style="list-style-type: none"> • Werte der Aktuellen Seite zwischenspeichern • eine Seite zurück blättern (Die Funktion dieser Befehlsschaltfläche wurde nicht realisiert . Der Benutzer kann also erst eine Korrektur an den Eingaben der vorhergehenden Eingabemaske anbringen, wenn er im Hauptmenü die Funktion „Studiendaten editieren“ wählt.)
Weiter	<ul style="list-style-type: none"> • Konsistenzprüfungen für die Eingaben dieser Seite • Falls die Tests keine Fehler finden, wird zur nächsten Seite gewechselt, sonst werden Fehler-/Warnmeldungen eingeblendet
Abbrechen	<ul style="list-style-type: none"> • Warnung und Rückfrage • Falls der Benutzer den Befehl bestätigt: Abbruch der Aktionen und zurück zur ersten Maske (Auswahl der Funktion)
Ton ein/aus	<ul style="list-style-type: none"> • Ein- beziehungsweise ausschalten des Signaltons, der jeweils bei Warnungen, Fehlermeldungen oder beim Erreichen des Ende eines Texteingabefeldes ertönt.
Info...	<ul style="list-style-type: none"> • Einblenden der Informationen zum Programm „SNF“ (Copyright, Kontaktmöglichkeit zum technischen Support)

Die Bildschirmmaske für die Suchfunktion enthält für alle möglichen Suchkriterien (vgl. Kapitel 5.1 Pflichtenheft, Seite 36) ein Eingabefeld und eine Checkbox.

Das Eingabefeld soll mit der gewünschten Zeichenfolge beschrieben werden. Dabei könne Sterne (*) und Fragezeichen (?) als Platzhalter für mehrere Zeichen beziehungsweise ein Zeichen eingesetzt werden. Das eingegebene Kriterium wird für die Suche erst verwendet, wenn die zugehörige Checkbox aktiviert wird.

Mehrere Kriterien können kombiniert werden.

In der Funktion Auswertungen werden der gewünschte Auswertungstyp mit den *radio buttons* (Optionsfeldgruppe) ausgewählt. Vor dem Ausdruck wird eine Seitenansicht eingeblendet und vor dem endgültigen Ausdruck wird der vom Betriebssystem her bekannte Druckerdialog angezeigt, wo man unter anderem das Drucken des Reports verhindern kann.

4.3.1.f Datenimport aus EMAIL

Der Mechanismus für den Import von Studiendaten aus Email-Attachments wurde zu einem grossen Teil von Frau N. Portmann entworfen (Idee: Daten in einer MS-Excel-Datei als Attachment an Email).

Sie hat eine Excel-Datei in Form eines Formulars entworfen, worin die Daten am PC eingetragen und dann per EMAIL ans BAG verschickt werden sollen.

Diese Datei wird durch einen Parser ausgewertet. Die Informationen gelangen in eine temporäre Datenbank, welche die gleiche Struktur aufweist wie die Datenbank „snfdb1“ des Programms „SNF“.

Diese Daten werden dem Benutzer mit Hilfe der Bildschirmmasken, die er schon von der Eingabe von Hand und vom Editieren der Studiendaten her kennt, angezeigt und mit jeder Betätigung der Befehlsschaltfläche „Weiter“ in die richtige Datenbank übernommen.

4.3.1.g Archivierung

Unter der Hauptfunktion „Administration“ werden die Dienste für das Archivieren, das Wiederherstellen der Datenbank und für das Anlegen einer neuen, leeren Datenbank angeboten. Unter den administrativen Funktionen wird konkret Folgendes verstanden:

Archivieren	Die Tabellen der Datenbank werden komprimiert und in einer Datei gespeichert. Den Namen der Datei und das Verzeichnis bzw. den Datenträger bestimmt der Benutzer.
Wiederherstellen	Die Tabellen der Datenbank werden gelöscht und durch diejenigen einer früher archivierten Archivdatei ersetzt.
Neue Datenbank	Die Tabellen der Datenbank werden zuerst vorsorglich in die Datei „Olddb.ZIP“ kopiert und dann gelöscht. Danach werden die Tabellen aus dem Verzeichnis „C:\snf\emptyDBs\“ in das Verzeichnis „C:\snf\DBFS\“ kopiert. Der alte Zustand der Datenbank kann somit durch die Funktion „Wiederherstellen“ (Archivdatei „Olddb.ZIP“) zurückgeholt werden. Falls eine alte Datei „Olddb.ZIP“ schon existierte, wird sie bei diesem Vorgang überschrieben.

Die Screenshots der beschriebenen Eingabemasken sind im Anhang abgebildet.

4.3.2 Klassendesign

Die Klassen, die in diesem Projekt zum Einsatz gelangen, werden aus den Visual FoxPro 3.0 Basisklassen abgeleitet.

Es sind dies die Klassen „snfformclass“ und „snfformclass2“ in der Klassenbibliothek „snfclasses.vcx“ (snf steht für **S**tudy **N**otification **F**orms).

Die Klasse „snfformclass“ als sogenannte Container-Klasse enthält alle Steuer- und sonstige Elemente mit ihren für dieses Projekt festgelegten Eigenschaften (Text, Stil, Anordnung). Diese Klasse liefert die Grundlage für die meisten Eingabemasken des Programms.

Die Klasse „snfformclass2“ dient als gemeinsame Basisklasse für die 6 (untereinander sehr ähnlichen) Eingabe-/ Editiermasken für die Studien- und Vergleichsprodukte, welche auf den letzten zwei Seiten der sechsseitigen Formularbögen des BAG erfasst werden. Sie wird aus der Klasse „snfformclass“ abgeleitet.

4.3.3 Objekthierarchie, Objekte und ihre Eigenschaften und Methoden

In diesem Abschnitt werden die für die Bedienung und Steuerung relevanten Eigenschaften und Methoden der einzelnen Formularobjekte beschrieben.

4.3.3.a Allgemeine Bemerkungen

Die Formularobjekte *seite1.scx*, *seite2.scx*, *seite3.scx*, *seite4.scx*, *seite5mem.scx* und *seite4b.scx* werden von der Klasse *snfformclass* der Klassendefinition *snfformclasses.vcx* abgeleitet.

Dadurch wird das einheitliche Aussehen und die Bedienung der Formulare gewährleistet.

Die Anordnung der Befehlsschaltflächen am unteren Rand der Formulare, die Position des Titels auf jeder Seite, die Standardtexteigenschaften und die visuellen Eigenschaften der übrigen Objekte sind somit für alle oben erwähnten Formularobjekte vorgegeben.

Darüber hinaus wird das Verhalten der Befehlsschaltflächen „Ton einschalten/ausschalten“ und „Info...“ in der Klassendefinition implementiert.

Ähnliche Überlegungen führten zum Entwurf der Klasse *snfformclass2*, welche als Grundlage für die Formulare *seite5va.scx*, *seite5bp.scx* und *seite5gt.scx* dient.

4.3.3.b Objekte und ihre Methoden und Eigenschaften

Formular-Objekt: intro.scx“

Zweck

Begrüßung des Benutzers.

Methoden

init(): Initialisierung des Timer-Objekts (Intro.Timer1)

Eigenschaften

c: Intensität der Farben R, G und B

count: Zähler für Timer-Events von Timer1 (Anzahl Intervalle zu 30ms)

Enthaltene Objekte

Timer1

Methoden

TimerEvent(): Steuerung der Animation, Aufruf des nächsten Formulars (*main_fcts.scx*)

Formularobjekt: info.scx“ (vgl. Abbildung 1)

Zweck

Informationen zum Programm einblenden.

Methoden

keine

Eigenschaften

keine

Enthaltene Objekte

Command1

Methoden

click(): Löscht das Formulars *info.scx* aus dem Speicher.

Formularobjekt: meldung.scx

Zweck

Formular für die Anzeige verschiedener Meldungen (z.B.: „Keine Daten gefunden.“)

Methoden

init(): Je nach Einstellung ertönt ein Warnton oder nicht.

Der als Parameter übergebene Text wird in das Formular plaziert.

Eigenschaften

keine

Enthaltene Objekte

btn_ok

Methoden

valid(): Löscht das Formular *meldung.scx* aus dem Speicher.

Formularobjekt: *warnung.scx* (vgl. Abbildung 6)

Zweck

Formular für die Anzeige verschiedener Warnungen („Gelbe Karte“).

Methoden

init(): Je nach Einstellung ertönt ein Warnton oder nicht.
Mit dem als Parameter übergebene Fehlercode wird in der Tabelle „ERRORS“ der zugehörige Fehlertext herausgelesen und an der entsprechenden Position auf dem Formular *warnung.scx* plziert.

Eigenschaften

keine

Enthaltene Objekte

btn_zur

Methoden

valid(): Setzt die globale Variable *antw* auf .F. und löscht das Formular *meldung.scx* aus dem Speicher.
Im aufrufenden Formular wird dadurch das Objekt, welches die Warnung hervorgerufen hat, wieder aktiviert und farblich hervorgehoben.

btn_ok

Methoden

valid(): Setzt die globale Variable *antw* auf .T. und löscht das Formular *meldung.scx* aus dem Speicher.

Formularobjekt: *fehler.scx* (vgl. Abbildung 2)

Zweck

Formular für die Anzeige verschiedener Fehlermeldungen („Rote Karte“).

Methoden

init(): Je nach Einstellung ertönt ein Warnton oder nicht.
Mit dem als Parameter übergebene Fehlercode wird in der Tabelle „ERRORS“ der zugehörige Fehlertext herausgelesen und an der entsprechenden Position auf dem Formular *fehler.scx* plziert.

Eigenschaften

keine

Enthaltene Objekte

btn_ok

Methoden

valid(): Löscht das Formular *fehler.scx* aus dem Speicher.

Formularobjekt: *main_fcts.scx* (vgl. Abbildung 3)

Zweck

Befehlsschaltflächen für die Auswahl der Hauptfunktionen des Programms („Neue Studie erfassen“, „Studiendaten ändern“, „Studiendaten importieren“, „Datenbank durchsuchen“, „Auswertungen erstellen“, „Beenden“).

Methoden

show(): HelpTopic ID einstellen

Eigenschaften

Keine.

Enthaltene Objekte

btn_ersch

Methoden

click(): Aufruf des Formularobjekts *eingabe.scx* (Festlegung des Studienobjektyps: VA, BP, GT) und Aufruf des Formularobjekts *seite1.scx* mit Studienobjektyp und „1“ (neue Studie) als Parameter.

btn_edit

Methoden

click(): Aufruf des Formularobjekts *suchen.scx* mit Parameter „F.“ (suchen um zu Editieren).

btn_such

Methoden

click(): Aufruf des Formularobjekts *suchen.scx*. mit Parameter „T.“ (nur suchen und Resultate anzeigen).

btn_auswert

Methoden

click(): Aufruf des Formularobjekts *auswert.scx*. (Auswertungen erstellen/drucken)

btn_auswert

Methoden

click(): Aufruf des Formularobjekts *zipunzip.scx*. (Administration: Archivierung, etc.)

btn_end

Methoden

click(): Einleitung des Programmendes, Aufruf des Codes *cleanup.prg*.

Timer1

Methoden

timer(): Steuerung der Farbanimation eines Textobjekts.

Formularobjekt: eingabe.scx (vgl. Abbildung 4)

Zweck

Festlegung des Typs des Studienobjekts (VAccine, BloodProduct oder GeneTherapy)

Methoden

show(): Festlegung der ID des Hilfekontexts für diese Eingabemaske.

unload(): Rückgabe des Wertes der Optionsfeldgruppe (1, 2 oder 3 für VA, BP oder GT) an das aufrufende Formularobjekt.

Eigenschaften

wert: Speichert den Wert der Optionsfeldgruppe.

Enthaltene Objekte

btn_ok

Methoden

click(): Übernimmt den Wert der gewählten Option in die Eigenschaft *wert*.

valid(): Zeigt eine Fehlermeldung mit Hilfe des Formularobjekts *fehler.scx* an oder gibt das Formular *eingabe.scx* frei (und übergibt somit die Kontrolle an das aufrufende Formularobjekt).

Formularobjekt: seite1.scx (vgl. Abbildung 5)

Zweck

Erfassung der detaillierten Daten der Studie (Details of the Study)

Methoden

- init():** Übernahme der Parameterwerte aus der Aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen, Berechnung einer neuen Referenznummer für die Studie, Festlegung der ID des Hilfekontexts für diese Eingabemaske.
- inputTest():** Tests für die eingegebenen Daten (Konsistenz, Vollständigkeit), Vorbereitung der Daten für die Speicherung in der Tabelle „STUDY“, Speichern der Daten in der Tabelle „STUDY“.

Eigenschaften

- calc_ref:** Speichert die berechnete Referenznummer der neuen Studie, um bei Bedarf auf diese zurückzugreifen.
- fkt:** Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“)
- stdtyp:** Enthält den Typ des Studienobjekts (1=VA, 2=BP oder 3=GT)

Enthaltene Objekte

cmd_forw

Methoden

- click():** Ruft die Methode *seite1.InputTest()* auf, Aufruf des Formulars *seite2.scx* mit Parametern Referenznummer, gewählte Funktion, Anzahl der an der Studie beteiligten *Centers* und Studienobjekt-Typ (VA, BP oder GT).

cmd_esc

Methoden

- click():** Rückkehr zum Formularobjekt *main_fcts.scx*.

Formularobjekt: *seite2.scx* (vgl. Abbildung 7)

Zweck

Erfassung der Daten des Sponsors der Studie (Details of Study Sponsor)

Methoden

- init():** Übernahme der Parameterwerte aus der Aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen, Festlegung der ID des Hilfekontexts für diese Eingabemaske
- inputTest():** Tests für die eingegebenen Daten (Konsistenz, Vollständigkeit), Vorbereitung der Daten für die Speicherung in den entsprechenden Tabellen, Speichern der Daten in den Tabellen „ADDRESS“, „FAX“, „PHONE“, „CRO“, „SPONSOR“, „MANUFACT“, „PERSON“, „STD_CRO“, „STD_SPON“, „STD_MANU“, „STD_PERS“.

Eigenschaften

- center_counter:** Enthält jeweils die Nummer des „Centers“, dessen Daten gerade erfasst werden (Zählvariable der FOR-Schleife).
- fkt:** Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).
- centers:** Enthält die Anzahl der an der Studie beteiligten „Center“.
- refnr:** Referenznummer der aktuellen Studie.
- stdtyp:** Typ des Studienobjekts (VA, BP, GT).

Enthaltene Objekte

cmd_forw

Methoden

- click():** Aufruf der Methode *seite2.InputTest()*, Ruft in einer FOR-Schleife das Formular *seite3.scx* mit Parametern „gewählte Funktion“, *centers* und *center_counter* auf (Erfassung der Daten zu allen an der Studie beteiligten *Centers*). Falls nicht „Abbrechen“ gewählt wurde, werden, je nach Wert der Eigenschaft *stdtyp*, die Formulare *seite5va.scx*, *seite5bp.scx* oder *seite5gt.scx*, je einmal mit drittem Parameter *.F.* und

einmal mit .T. aufgerufen. Die zwei anderen Parameter sind Referenznummer und gewählte Funktion („Neue Studie“ oder „Studie editieren“).

Am Schluss wird das Formular *main_fcts.scx* aufgerufen.

cmd_esc

Methoden

click(): Rückkehr zum Formularobjekt *main_fcts.scx*.

Formularobjekt: *seite3.scx* (vgl. Abbildung 9)

Zweck

Erfassung der Daten der untersuchenden Stelle (Details of Investigator)

Methoden

init(): Übernahme der Parameterwerte aus der aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen, Initialisierung der Arrays für die Zwischenspeicherung der Daten zu den Co-Investigators und deren Telefonnummern, Ermittlung der zur Zeit letzten ID-Nummer in der Tabelle „HOSPITAL“, Initialisierung der Zähler für die Indizes der Arrays Festlegung der ID des Hilfkontexts für diese Eingabemaske. Je nach Wert des ersten Parameters („Neue Studie“ oder „Studie editieren“) werden die Eigenschaften der Steuerelemente *btn_more* und *Command2* (Befehlsschaltflächen) und des Kombinationsfeldobjekt *conamelist* entsprechend gesetzt.

error(): Abfangen der Fehlermeldung "Zieltabelle wird bereits in einer Beziehung benutzt" (VFP-Fehlercode: 1147) infolge des Befehls **SELECT hospital** in der Methode *seite3.inputTest()* (cf. *seite3.btn_more.Click(l_value)*).

inputTest(): Tests für die eingegebenen Daten (Konsistenz, Vollständigkeit), Vorbereitung der Daten für die Speicherung in den entsprechenden Tabellen, Berechnung einer neuen ID-Nummer für das aktuelle *Center* (falls nicht schon bestimmt), Speichern der Daten in den Tabellen „ADDRESS“, „FAX“, „PHONE“, „INVESTIG“, „CO_INV“, „HOSPITAL“, „STD_HOSP“.

Eigenschaften

coinv_nbr: Enthält die Anzahl der Co-Investigators, deren Daten bisher eingegeben wurden.

hosp_id: Enthält die ID-Nummer des aktuellen *Center*'s.

inv_id: ID des Investigators.

tel_cnt: Zähler für Telefonnummern der Co-Investigators (Indexwerte für Array „coTelArr“).

more: Enthält logischen Wert, der angibt, ob die Methode *seite3.btn_more.Click()* durch eine Benutzeraktion oder programmgesteuert aufgerufen wurde.

fkt: Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).

centers: Enthält die Anzahl der an der Studie beteiligten „Center“.

edited: .T. oder .F., je nachdem, ob in der Funktion „Studie Editieren“ die Liste der Co-Investigators verändert wurde.

Enthaltene Objekte

btn_more(l_value)

Methoden

- click(): Berechnung einer neuen ID für das aktuelle *Center* (falls nicht schon bestimmt), Berechnung von je einer neuen ID für den Co-Investigator und, falls angegeben, für dessen Telefonnummer,
Im Fall der Erfassung einer *neuen Studie*: Zwischenspeicherung der berechneten ID's und der Angaben aus den Eingabefeldern für „Co_Investigators/Staff Members ...“ in den Arrays *coInvArr* und *coTelArr*.
Im Fall des Editierens einer Studie: Direkte Speicherung der Änderung der Co-Investigator-Angaben in den Tabellen „PHONE“ und „CO_INV“.
- error(): Abfangen der Fehlermeldung "Zieltabelle wird bereits in einer Beziehung benutzt" (VFP-Fehlercode: 1147) infolge des Befehls **SELECT hospital** in der Methode *seite3.btn_more.Click(l_value)*.

cmd_forw

Methoden

- click(): Speichern der Daten des zuletzt eingegebenen Co-Investigators (falls vorhanden), Aufruf der Methode *seite3.InputTest()*,
Aufruf des Formularobjekts *seite4.scx* mit Parametern „gewählte Funktion“ (*fkt*), Bezeichnung des aktuellen *Centers* (*hosp.Value*) und ID-Nummer des aktuellen „Center“'s (*hosp_id*), falls nicht „Abbrechen“ gewählt wurde.

cmd_esc

Methoden

- click(): Rückkehr zum Formularobjekt *main_fcts.scx*.

Command2

Methoden

- click(): Auslesen der Daten aus den Tabellen „PHONE“ und „CO_INV“ und Vorbereitung der Editierfelder für die entsprechenden Daten.
Ausschalten (Enable=.F.) der Befehlsschaltfläche *cmd_forw*, damit nicht fälschlicherweise darauf anstelle der Befehlsschaltfläche *btn_more* (mit neuer Beschriftung „Speichern“) geklickt wird.

Formularobjekt: *seite4.scx* (vgl. Abbildung 11)

Zweck

Erfassung der Daten der ethischen Kommission der untersuchenden Stelle (Details of Ethical Review Committee ERC).

Methoden

- init(): Übernahme der Parameterwerte aus der aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen,
Initialisierung des Arrays für die Zwischenspeicherung der Daten zu den ERC-Mitgliedern (für die Funktion „Neue Studie“),
Ermittlung der zur Zeit letzten ID-Nummer in der Tabelle „ERCMEMB“,
Initialisierung der Zähler für den Index des Arrays,
Festlegung der ID des Hilfef Kontexts für diese Eingabemaske.
Im Editiermodus werden ähnlich wie in Formular *seite3.scx* die Daten der ERC-Mitglieder eingeblendet. Diese können dann mit den Befehlsschaltflächen *btn_more* und *Command2* (nur sichtbar wenn nötig) ergänzt oder editiert werden.
- inputTest(): Tests für die eingegebenen Daten (Konsistenz, Vollständigkeit),
Vorbereitung der Daten für die Speicherung in den entsprechenden Tabellen,
Berechnung einer neuen ID-Nummer für die ethische Kommission (falls nicht schon bestimmt),
Aufruf des Formularobjekts *seite4b.scx* für die Eingabe des Berichtes der ethischen Kommission (Ethical Committee Approval), falls dieser vergessen wurde,
Zwischenspeicherung der Angaben zum Vorsitzenden und zum Verantwortlichen für die

Statistik im Array *membArr*, falls diese vorliegen,
Speichern der vorbereiteten und sonstiger Daten in den Tabellen „ADDRESS“, „ERC“,
„DECISION“, „ERCMEMB“.

Eigenschaften

- appr_txt*: Enthält den Text des „Approvals“.
- hosp_id*: Enthält die ID-Nummer des aktuellen *Center*'s.
- erc_id*: Enthält die ID der ethischen Kommission des aktuellen *Center*'s.
- chair*: Enthält die ID des ERC-Mitglieds, das die Funktion des Vorsitzenden inne hat.
- stat*: Enthält die ID des ERC-Mitglieds, das die Funktion des Statistikers inne hat.
- more*: Enthält logischen Wert, der angibt, ob die Methode *seite4.more_memb.Click(l_value)* durch eine Benutzeraktion oder programmgesteuert aufgerufen wurde.
- pkt*: Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).
- members*: Enthält die Anzahl der Mitglieder des ERC (ausgenommen Vorsitzender und Verantwortlicher für die Statistik)
- edited*: .T. oder .F., je nachdem, ob in der Funktion „Studie Editieren“ die Liste der ERC-Mitglieder verändert wurde.

Enthaltene Objekte

more_memb(l_value)

Methoden

- click()*: Im Fall „Neue Studie“: Berechnung einer neuen ID für das ERC (falls nicht schon bestimmt), Berechnung einer neuen ID für das ERC-Mitglied, Zwischenspeicherung der berechneten ID's und der Angaben aus den Eingabefeldern für „Other Members“ im Array *membArr*.
Im Fall „Studie Editieren“: Bereitstellen des Texteingabefeldes für ein neues Mitglied.

btn_eca

Methoden

- click()*: Aufruf des Editierfensters *seite4b.scx* für den Text des *Ethical Committee Approval* (ECA) mit dem Inhalt der Eigenschaft *appr_txt* als Parameter.
Setzt den Fokus auf die Befehlsschaltflächenobjekt *cmd_forw*.

cmd_forw

Methoden

- click()*: Speichern der Daten des zuletzt eingegebenen ERC-Mitglieds (falls vorhanden), Aufruf der Methode *seite4.InputTest()*.

cmd_esc

Methoden

- click()*: Rückkehr zum Formularobjekt *main_fcts.scx*.

Command2

Methoden

- click()*: Auslesen der Daten aus der Tabelle „ERCMEMB“ und Vorbereitung der Editierfelder für die entsprechenden Daten.
Ausschalten (Enable=.F.) der Befehlsschaltfläche *cmd_forw*, damit nicht fälschlicherweise darauf anstelle der Befehlsschaltfläche *more_memb* (mit neuer Beschriftung „Speichern“) geklickt wird.

Formularobjekt: *seite5va.scx* (vgl. Abbildung 13)

Zweck

Erfassung der Daten zum Studienimpfstoff und Vergleichsimpfstoff.

Methoden

- init()*: Übernahme der Parameterwerte aus der aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen,

Initialisierung des Arrays für die Zwischenspeicherung der Daten zu den aktiven Komponenten des Impfstoffes (für die Funktion „Neue Studie“),
Ermittlung der zur Zeit letzten ID-Nummer in der Tabelle „VACCINE“,
Initialisierung des Zählers für den Index des Arrays,
Festlegung der ID des Hilfekontexts für diese Eingabemaske.

Im Editiermodus werden ähnlich wie in Formular *seite3.scx* die Daten der aktiven Komponenten eingeblendet. Diese können dann mit den Befehlsschaltflächen *more_comp* und *Command2* (nur sichtbar wenn nötig) ergänzt oder editiert werden.

in_tst_std(): Tests für den ersten Teil der eingegebenen Daten zum *Studienimpfstoff* (Konsistenz, Vollständigkeit)

Setzen des globalen Flags *all_done* gemäss Testergebnis.

in_tst_comp(): Tests für den ersten Teil der eingegebenen Daten zum *Vergleichsimpfstoff* (Konsistenz, Vollständigkeit)

Setzen des globalen Flags *all_done* gemäss Testergebnis.

cmp_tst_std(): Tests für die Daten der aktuellen aktiven Komponente des *Studienimpfstoffs*.

cmp_tst_comp(): Tests für die Daten der aktuellen aktiven Komponente des *Vergleichsimpfstoffs*.

Eigenschaften

flag: .T., für *Studienimpfstoff*, .F. für *Vergleichsimpfstoff*

refnr: Referenznummer der aktuellen Studie.

comps: Anzahl der aktiven Komponenten.

more: Enthält logischen Wert, der angibt, ob die Methode *seite5va.more_comp.Click(l_value)* durch eine Benutzeraktion oder programmgesteuert aufgerufen wurde.

fkt: Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).

edited: .T. oder .F., je nachdem, ob in der Funktion „Studie Editieren“ die Liste der aktiven Komponenten verändert wurde.

orig_fkt: Speichert den Wert der Eigenschaft *fkt*, falls die Daten der Studie in einer früheren Sitzung nicht vollständig eingegeben waren und jetzt in der Funktion „Studie editieren“ ergänzt werden, so dass die Funktionalität der Objekte der Software mit dem Wert *fkt*=1 („Neue Studie“) wiederbenutzt werden können.

Enthaltene Objekte

more_comp(l_value)

Methoden

click(): Im Fall „Neue Studie“: Berechnung einer neuen ID für den Impfstoff (falls nicht schon bestimmt),
Berechnung einer neuen ID für die aktive Komponente,
Zwischenspeicherung der berechneten ID's und der Angaben aus den Eingabefeldern für „Active Component“ im Array *actCompsArr*.
Im Fall „Studie Editieren“: Bereitstellen des Texteingabefeldes für eine neue aktive Komponente.

cmd_forw

Methoden

click(): Speichern der Daten des zuletzt eingegebenen aktiven Komponente (falls vorhanden),
Schlusstest durch Aufruf der Methoden zum testen der Daten (je nach gewählter Funktion und Zustand der Flags): *seite5va.in_tst_std()*, *seite5va.in_tst_comp()*, *seite5va.cmp_tst_std()*, *seite5va.cmp_tst_comp()* und Speichern der Daten.

cmd_esc

Methoden

click(): Rückkehr zum Formularobjekt *main_fcts.scx*.

Command2

Methoden

click(): Auslesen der Daten aus der Tabellen „ACT_COMP“ und Vorbereitung der Editierfelder für die entsprechenden Daten.
Ausschalten (Enable=.F.) der Befehlsschaltfläche *cmd_forw*, damit nicht fälschlicherweise

darauf anstelle der Befehlsschaltfläche *more_comp* (mit neuer Beschriftung „Speichern“) geklickt wird.

Formularobjekt: *seite5bp.scx*

Zweck

Erfassung der Daten zum Studien-Blutprodukt und Vergleichs-Blutprodukt.

Methoden

- init()*: Übernahme der Parameterwerte aus der aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen, Ermittlung der zur Zeit letzten ID-Nummer in der Tabelle „BP“, Festlegung der ID des Hilfekontexts für diese Eingabemaske.
- in_tst_std()*: Tests für die eingegebenen Daten zum *Studienblutprodukt* (Konsistenz, Vollständigkeit), setzen des globalen Flags *all_done* gemäss Testergebnis.
- in_tst_comp()*: Tests für die eingegebenen Daten zum *Vergleichsblutprodukt* (Konsistenz, Vollständigkeit), setzen des globalen Flags *all_done* gemäss Testergebnis.

Eigenschaften

- flag*: .T., für *Studienblutprodukt*, .F. für *Vergleichsblutprodukt*
- refnr*: Referenznummer der aktuellen Studie.
- comps_txt*: Enthält den Text, der die *aktiven Komponenten* des Blutproduktes beschreibt, wird in der Tabelle „BP“ in einem MEMO-Feld gespeichert.
- adjuv_txt*: Enthält den Text, der die *Adjuvants* des Blutproduktes beschreibt (wird in der Tabelle „BP“ in einem MEMO-Feld gespeichert).
- vimeth_txt*: Enthält den Text, der die *Virus Inactivation Methods* des Blutproduktes beschreibt (wird in der Tabelle „BP“ in einem MEMO-Feld gespeichert).
- fkt*: Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).
- id*: Enthält die ID des gerade bearbeiteten Blutproduktes.
- orig_fkt*: Speichert den Wert der Eigenschaft *fkt*, falls die Daten der Studie in einer früheren Sitzung nicht vollständig eingegeben waren und jetzt in der Funktion „Studie editieren“ ergänzt werden, so dass die Funktionalität der Objekte der Software mit dem Wert *fkt*=1 („Neue Studie“) wiederbenutzt werden können.

Enthaltene Objekte

btn_actcomp

Methoden

- click()*: Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „BP“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *btn_adjuvants*.

btn_adjuvants

Methoden

- click()*: Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „BP“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *btn_vimeth*.

btn_vimeth

Methoden

- click()*: Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „BP“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *cmd_forw*.

cmd_forw

Methoden

click(): Schlusstest durch Aufruf der Methoden zum testen der Daten (je nach gewählter Funktion und Zustand der Flags): *seite5bp.in_tst_std()*, *seite5bp.in_tst_comp* und Speichern der Daten.

cmd_esc

Methoden

click(): Rückkehr zum Formularobjekt *main_fcts.scx*.

Formularobjekt: *seite5gt.scx* (vgl. Abbildung 15)

Zweck

Erfassung der Daten zum Studienprodukt (Gentherapie) und Vergleichsprodukt.

Methoden

init(): Übernahme der Parameterwerte aus der aufrufenden Methode in eigene Eigenschaften, Initialisierung, der globalen und Memory-Variablen, Ermittlung der zur Zeit letzten ID-Nummer in der Tabelle „GT“, Festlegung der ID des Hilfekontexts für diese Eingabemaske.

in_tst_std(): Tests für die eingegebenen Daten zur *Studiengentherapie* (Konsistenz, Vollständigkeit), setzen des globalen Flags *all_done* gemäss Testergebnis.

in_tst_comp(): Tests für die eingegebenen Daten zur *Vergleichsgentherapie* (Konsistenz, Vollständigkeit), setzen des globalen Flags *all_done* gemäss Testergebnis.

Eigenschaften

flag: .T., für *Studienprodukt*, .F. für *Vergleichsprodukt*.

refnr: Referenznummer der aktuellen Studie.

comps_txt: Enthält den Text, der die *Komponenten* der Gentherapie beschreibt, wird in der Tabelle „GT“ in einem MEMO-Feld gespeichert.

vector_txt: Enthält den Text, der den *Vector* der Gentherapie beschreibt, wird in der Tabelle „GT“ in einem MEMO-Feld gespeichert.

safe_prec_txt: Enthält den Text, der die *Safety and Precautions* des Blutproduktes beschreibt, wird in der Tabelle „BP“ in einem MEMO-Feld gespeichert.

fkt: Enthält die gewählte Funktion (1=„Neue Studie“ oder 2=„Studiendaten Editieren“).

id: Enthält die ID der gerade bearbeiteten Gentherapie.

orig_fkt: Speichert den Wert der Eigenschaft *fkt*, falls die Daten der Studie in einer früheren Sitzung nicht vollständig eingegeben waren und jetzt in der Funktion „Studie editieren“ ergänzt werden, so dass die Funktionalität der Objekte der Software mit dem Wert *fkt*=1 („Neue Studie“) wiederbenutzt werden können.

Enthaltene Objekte

btn_prodcomp

Methoden

click(): Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „GT“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *btn_vector*.

btn_vector

Methoden

click(): Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „GT“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *btn_safe_prec*.

btn_safe_prec

Methoden

click(): Aktiviert das Formular *seite5mem.scx*.
Im Fall „Studie Editieren“ wird als Parameter der Text, den das entsprechende MEMO-Feld der Tabelle „GT“ enthält, übergeben, sonst ein leerer String.
Setzt den Fokus auf die Befehlsschaltfläche *cmd_forw*.

cmd_forw

Methoden

click(): Schlusstest durch Aufruf der Methoden zum testen der Daten (je nach gewählter Funktion und Zustand der Flags): *seite5gt.in_tst_std()*, *seite5gt.in_tst_comp* und Speichern der Daten.

cmd_esc

Methoden

click(): Rückkehr zum Formularobjekt *main_fcts.scx*.

Formularobjekt: *seite4b.scx* (vgl. Abbildung 12)

Zweck

Erfassung des Textes zum ECA (Ethical Committee Approval)

Methoden

init(): Übernahme des Parameterwertes (Text) aus der aufrufenden Methode in die Eigenschaften *approval_txt*.

Eigenschaften

keine

Enthaltene Objekte

cmd_forw

Methoden

click(): Löscht das Formular *seite4b.scx* aus dem Speicher.

Formularobjekt: *seite5mem.scx*

Zweck

Erfassung von Texten, die zu den verschiedenen Textinformationen der drei Formularobjekte *seite5va.scx*, *seite5bp.scx* und *seite5gt.scx* gehören.

Methoden

init(): Übernahme des Parameterwertes (Text) aus der aufrufenden Methode in die Eigenschaften *txt*.

Eigenschaften

keine

Enthaltene Objekte

cmd_forw

Methoden

click(): Löscht das Formular *seite5mem.scx* aus dem Speicher.

Formularobjekt: *suchen.scx* (vgl. Abbildung 8)

Zweck

Festlegen der Suchkriterien und starten der Suche nach Studien.

Methoden

init(): Festlegung der ID des Hilfef Kontexts für diese Eingabemaske.

Eigenschaften

search: Flag, das anzeigt, ob nur gesucht wird (.T.) oder nach dem Suchen das Editieren der Daten zur gewählten Studie gestartet wird (.F.).

Enthaltene Objekte

cmd_such

Methoden

- click(): SQL-Statements für die Suche nach den Studien gemäss den gewählten Kriterien.
Falls die Tabelle mit den Resultaten nicht leer ist, wird das Formularobjekt *showres.scx* mit dem Namen dieser Tabelle (MASTER_R) aufgerufen. Der zweite Parameter ist das Flag *search*.

Formularobjekt: *showres.scx* (vgl. Abbildung 14)

Zweck

Anzeigen der gefundenen Studien. Falls „Studiendaten editieren“ aus den Hauptfunktionen gewählt wurde, kann eine Studie in der Tabelle ausgewählt werden und dann mittels der Befehlsschaltfläche *cmd_forw* (jetzt mit Aufschrift „Editieren“) das Formularobjekt *seite1.scx* aufgerufen werden.

Methoden

- init(): Eigenschaften der enthaltenen Steuerelemente gemäss den Parametern festlegen.
show(): Festlegung der ID des Hilfekontexts für diese Eingabemaske.

Eigenschaften

- search: Flag, das anzeigt, ob nur gesucht wird (.T.) oder nach dem Suchen das Editieren der Daten zur gewählten Studie gestartet wird (.F.).
rn: Enthält die RecordNumber der Ausgewählten Studie in der Tabelle.

Enthaltene Objekte

cmd_forw

Methoden

- click(): Ruft das Formularobjekt *seite1.scx* mit den Parametern Referenznummer, Studententyp (VA, BP oder GT) der ausgewählten Studie und „2“ (für „Studiendaten editieren“) auf.
when(): Gibt erst .T. zurück, wenn mit der Maus in ein Zeile der Tabelle geklickt wurde, und so eine Studie zum editieren ausgewählt wurde.

cmd_back()

Methoden

- click(): Schliesst das Formular *showres.scx* und aktiviert das vorherige Formularobjekt (*suchen.scx*), wo man die Suchkriterien verändern kann.

cmd_esc

Methoden

- click(): Schliesst das Formular *showres.scx* und aktiviert das Formularobjekt *main_fcts.scx*, wo man dann eine andere Hauptfunktion auswählen kann.

Formularobjekt: *auswert.scx* (vgl. Abbildung 10)

Zweck

Auswahl der gewünschten Auswertung und Bereitstellung der gewünschten Daten in Tabellen, die den Bericht-Objekten *ausw1.frx*, *ausw2.frx*, ..., *ausw9.frx* zwecks Voransicht auf dem Bildschirm und dann zum Drucken übergeben werden.

Methoden

- show(): Festlegung der ID des Hilfekontexts für diese Eingabemaske.

Eigenschaften

- txt_param: Text, der als Parameter für die aufgerufenen Berichtobjekte verwendet wird (vor allem für das Berichtobjekt *ausw2.frx*).

Enthaltene Objekte

ausw typ

Zweck

Auswahl der gewünschten Auswertung mit Hilfe dieser Optionsfeldgruppe (*radio button group*)

cmd_forw()

Methoden

- when(): Gibt den booleschen Wert .T. zurück, wenn eine Auswahl getroffen wurde, das heisst, die Methode click() dieser Befehlsschaltfläche wird erst aktiviert, falls der Benutzer eine Auswahl für den Berichtstyp getroffen hat.
- click(): Enthält die SQL-Statements, um die Tabellen für die Berichtobjekte bereitzustellen. Die Resultate werden in CURSOR-Tabellen geschrieben (physikalisch im RAM). Danach wird das zugehörige Berichtobjekt aufgerufen, welches dann die Daten in diesen Tabellen herausliest und weiterbearbeitet (Anzeigen und Drucken).

cmd_esc

Methoden

- click(): Schliesst das Formular *auswert.scx* und aktiviert das Formularobjekt *main_fcts.scx*, wo man dann eine andere Hauptfunktion auswählen kann.

Formularobjekt: zipunzip.scx (vgl. Abbildung 16)

Zweck

Formular für die Auswahl der Funktionen „Archivieren“, „Wiederherstellen“ und „Neue Datenbank erstellen“ und Ausführung der gewünschten Funktionen.

Methoden

- Show(): Je nach Einstellung ertönt ein Warnton oder nicht.
Festlegung der ID des Hilfekontexts für diese Eingabemaske.

Eigenschaften

keine

Enthaltene Objekte

btn_zip

Methoden

- click(): Ruft den Betriebssystemdialog für das Speichern von Dateien auf, um dem Anwender einen Namen, den Datenträger und das Verzeichnis für die zu erstellende Archiv-Datei eingeben zu lassen.
Diese Informationen werden dem Programm „pkzip.exe“ als Parameter übergeben, welches die Tabellen der Datenbank (im Verzeichnis C:\snf\DBFS\) in eine Archiv-Datei im ZIP-Format in komprimierter Form speichert.

btn_unzip

Methoden

- click(): Ruft den Betriebssystemdialog für das Öffnen von Dateien auf, damit der Anwender die Archiv-Datei auswählen kann, die wiederhergestellt werden soll.
Der Dateiname wird als Parameter dem Programm „pkunzip.exe“ übergeben.

btn_newdb

Methoden

- click(): Zuerst werden die Tabellen der Datenbank in das ZIP-File „Olddb.ZIP“ gespeichert. Dann wird das bestehende Verzeichnis „C:\snf\DBFS\“ mit den Tabellen der Datenbank durch ein leeres ersetzt. Aus dem Verzeichnis „C:\snf\emptyDBs\“ werden die leeren Tabellen in das neu erstellte Verzeichnis \DBFS\ kopiert.

4.3.4 Konsistenztests

Die nachfolgend beschriebenen Konsistenztests dienen zur Kontrolle der eingegebenen Informationen in den einzelnen Datenfeldern der Eingabemasken.

Dabei werden die von den Benutzern in den Eingabemasken eingetragenen Werte so weit wie möglich auf logische Korrektheit und auf Vollständigkeit geprüft.

Die Tests erzeugen Warnungen und Fehlermeldungen, die von der Benutzerin oder vom Benutzer quittiert werden müssen.

Warnungen machen die Benutzer im Allgemeinen darauf aufmerksam, dass bestimmte Datenfelder nicht beschrieben wurden.

Fehlermeldungen werden erzeugt, wenn logisch überprüfbare, sinnlose Eingaben gemacht wurden oder wenn obligatorische Angaben übergangen wurden. Das Umschalten auf die nächste Formularseite ist nur dann möglich, wenn alle obligatorischen Angaben in den entsprechenden Datenfeldern vollständig sind.

4.3.4.a Legende für die nachfolgenden Tabellen:

Warnungen:

Typ	Text
W2	Sie haben in dieser Auswahl nichts markiert
W4	Sie haben hier keine Option ausgewählt
W6	Sie haben keine Co-Investigators eingegeben
W7	Diese Angaben sind nicht vollständig
W8	Sie haben keine weiteren ERC-Mitglieder eingegeben
W9	Sie haben noch keinen Text für das ECA eingegeben
W10	Sie haben keine Komponenten angegeben.
W11	Sie haben bisher nur eine Komponente eingegeben.
W12	Sie haben bisher keine Angaben zu "Adjuvants" gemacht.
W13	Sie haben bisher keine Angaben zu "Virus Inactivation Methods" gemacht.
W14	Sie haben bisher keine Angaben zu "Vector" gemacht.
W15	Sie haben bisher keine Angaben zu "Safety/Precautions" gemacht.

Fehlermeldungen:

Typ	Text
E1	Sie haben in diesem Feld nichts eingegeben
E2	Aus dieser Auswahl muss mindestens ein Feld markiert werden
E3	Diese Angaben können nicht stimmen
E4	Wählen Sie eine Option aus der Auswahl
E5	Monatsangaben liegen zwischen 1 und 12
E6	Falls das aktuelle Jahr kleiner als 2000 ist und Sie wollen eine Jahreszahl im 21. Jahrhundert eingeben, so muss das Jahr vollständig angegeben werden (mit Tausender- und Hunderterziffer).
E7	Sie haben diese Felder nicht vollständig ausgefüllt
E8	Sie müssen mindestens eine Komponente eingeben

4.3.4.b Tests

Tests für Study Details (Seite 1)

Datenfeld	Datentyp	Test	Aktion
DT_CODE	char	Leer ?	W 1
DT_PHASE	check boxes	Nichts ausgewählt	W 2
DT_TITLE	memo	Leer ?	E 1
DS_PROS DS_RETRO DS_CONTR DS_OPEN DS_BLIND DS_DBLIND DS_RANDO DS_PARALL DS_CROSS DS_MULTIC DS_OTHERT	check boxes	Ist neben der Angabe, ob es sich bei der Studie um eine „single“ oder „multi center study“ handelt, mindestens eines der anderen Optionen geklickt oder ist DS_OTHER beschrieben ? Sich logisch ausschliessende Studiendesigntypen sind in radio button group-Steuer-elemente zusammengefasst (eine Studie kann z. B. nicht gleichzeitig ein „open“ und ein „blind“ design haben).	E 2
DS_NBRC	numerisch	Falls „multi center“ ausgewählt: ist DS_NBR > 1 ?	E 1
DS_STARTM DS_STARTY DS_ENDM DS_ENDY	numerisch	Leer ? Ist „end year“ < „start year“ Falls „end year“ = „start year“: Ist „end month“ > „start year“ ?	W1 E3

Tests für Details of Study Sponsor (Seite 2)

Datenfeld	Datentyp	Test	Aktion
SP_COMP	char	Leer ?	W3
SP_ADDR		Falls SP_COMP nicht leer: ist SP_ADDR leer ?	W1
SP_TEL	char	Falls SP_COMP nicht leer: ist SP_TEL leer ?	W1
SP_FAX	char	Falls SP_COMP nicht leer: ist SP_FAX leer ?	W1
P_NAME	char	Leer ?	W1
P_ADDR	char	Falls P_NAME nicht leer: ist P_ADDR leer ?	W1
P_TEL	char	Falls P_NAME nicht leer: ist P_TEL leer ?	W1
P_FAX	char	Falls P_NAME nicht leer: ist P_FAX leer ?	W1

Tests für Details of Investigator (Seite 3)

(Falls unter Study Design „multicenter“ gewählt wurde, sind für jedes Center folgende Tests durchzuführen:)

Datenfeld	Datentyp	Test	Aktion
H_NAME H_ADDR	char	Leer ?	E1
H_TEL H_FAX	char	Leer ?	W1
INV_NAME INV_DIVIS	char	Leer ?	E1
INV_TEL INV_FAX	char	Leer ?	W1
CO_NAME	char	Leer ?	W1
CO_DIVIS	char	Falls CO_NAME nicht leer: Ist CO_DIVIS leer ?	W1
CO_TEL	char	Falls CO_NAME nicht leer: Ist CO_TEL leer ?	W1

Tests für Details of ERC (Seite 4)

(Falls unter Study Design „multicenter“ gewählt wurde, sind für jedes Center folgende Tests durchzuführen:)

Datenfeld	Datentyp	Test	Aktion
ERC_NAME ERC_ADDR	char	Leer ?	E1
CHAIRNAME	char	Leer ?	W1
CHAIRPROF	char	Falls CHAIRNAME nicht leer: Ist CHAIRPROF leer ?	W1
CHAIRSEX	2 radio butt.	Falls CHAIRNAME nicht leer: Nichts ausgewählt ?	W2
STATNAME	char	Leer ?	W1
STATPROF	char	Falls STATNAME nicht leer: Ist STATPROF leer ?	W1
STATSEX	2 radio buttons	Falls STATNAME nicht leer: Nichts ausgewählt ?	W2
DEC_DATE	char	Leer ?	E1
DEC_DEC	int	Leer ?	E2
<i>Folgende Angaben werden in einer separaten Tabelle erfasst</i>			
NAME	char	Leer ?	W1
PROF	char	Falls NAME nicht leer: ist PROF leer ?	W1
SEX	2 radio buttons	Falls NAME nicht leer: SEX Nichts ausgewählt ?	W2

Tests für Vaccine Identification: Study Vaccine

Datenfeld	Datentyp	Test	Aktion
TRDNAME CODENBR BAGNBR	char	Leer ?	W1
LOTNBR	char	Leer ?	E1
APPL	check boxes	Ist mindestens eines geklickt ?	E2
APPL_OTHER	char	Falls APPL_OTHER geklickt: APPL_OTHER leer ?	E1
<i>Folgender Angabensatz wird in einer separaten Tabelle erfasst, dabei muss mindestens eine aktive Komponente eingegeben werden</i>			
AC_NAME	char	Falls erste aktive Komponente: Leer ? Sonst	E1 W1
AC_STATE	3 radio buttons	Falls erste aktive Komponente: Nichts ausgewählt ? Sonst: Falls AC_NAME nicht leer: Nichts ausgewählt ?	E4 W4
AC_TYPE	3 radio buttons	Falls erste aktive Komponente: Nichts ausgewählt ? Sonst: Falls AC_NAME nicht leer: Nichts ausgewählt	E4 W4
AC_OTYPE	char	Falls erste Komponente: Falls Option 3 von AC_TYPE gesetzt: Leer? Sonst: Falls Option 3 von AC_TYPE gesetzt: Leer ?	E1 W1
AC_RECOMB	3 radio buttons	Nichts ausgewählt ?	W2

Tests für Vaccine Identification: Comparison Vaccine

Datenfeld	Datentyp	Test	Aktion
TRDNAME	char	Leer ?	W1
CODENBR BAGNBR LOTNBR	char	Falls TRDNAME nicht leer: leer ?	W1
APPL	check boxes	Falls TRDNAME nicht leer: ist mind. eines geklickt ?	W2
APPL_OTHER	char	Falls APPL_OTHER geklickt: Ist APPL_OTHER leer ?	W1
<i>Folgender Angabensatz wird in einer separaten Tabelle erfasst, dabei muss mindestens eine aktive Komponente eingegeben werden, falls für TRDNAME eine Eingabe gemacht wurde</i>			
AC_NAME	char	Falls TRDNAME nicht leer: Leer ?	W1
AC_STATE	3 radio buttons	Falls TRDNAME nicht leer: Nichts ausgewählt ?	W4
AC_TYPE	3 radio buttons	Falls TRDNAME nicht leer: Nichts ausgewählt ?	W4
AC_OTYPE	char	Falls Option 3 von AC_TYPE gesetzt: Leer ?	W1
AC_RECOMB	3 radio buttons	Falls TRDNAME nicht leer: Nichts ausgewählt ?	W2

Tests für Blood Product Identification: Study Product

Datenfeld	Datentyp	Test	Aktion
TRDNAME CODENBR BAGNBR	char	Leer ?	W1
LOTNBR	char	Leer ?	E1
APPL	check box gr.	Ist mindestens eines geklickt ?	E2
APPL_OTHER	char	Falls APPL_OTHER geklickt: APPL_OTHER leer ?	E1
AC_COMP	memo	Leer ?	E1
ADJUVANTS	memo	Leer ?	W1
VIMETH	memo	Leer ?	W1

Tests für Blood Product Identification: Comparison Product

Datenfeld	Datentyp	Test	Aktion
TRDNAME CODENBR BAGNBR	char	Leer ?	W1
LOTNBR	char	Falls TRDNAME nicht leer: Leer ?	W1
APPL	check box gr.	Falls TRDNAME nicht leer: ist mind. eines geklickt ?	W2
APPL_OTHER	char	Falls APPL_OTHER geklickt: ist APPL_OTHER leer ?	E1
AC_COMP	memo	Falls TRDNAME nicht leer: Leer ?	E1
ADJUVANTS	memo	Falls TRDNAME nicht leer: Leer ?	W1
VIMETH	memo	Falls TRDNAME nicht leer: Leer ?	W1

Tests für (GT-) Product Identification: Study Product

Datenfeld	Datentyp	Test	Aktion
TRDNAME CODENBR BAGNBR	char	Leer ?	W1
LOTNBR	char	Leer ?	E1
APPL	check box gr.	Ist mindestens eines geklickt ?	E2
APPL_OTHER	char	Falls APPL_OTHER geklickt: ist APPL_OTHER leer ?	E1
PROD_COMP	memo	Leer ?	E1
VECTOR	memo	Leer ?	W1
SAFE_PREC	memo	Leer ?	W1

Tests für Blood Product Identification: Comparison Product

Datenfeld	Datentyp	Test	Aktion
TRDNAME CODENBR BAGNBR	char	leer ?	W1
LOTNBR	char	Falls TRDNAME nicht leer: Leer ?	W1
APPL	check box gr.	Falls TRDNAME nicht leer: Ist mind. eines geklickt ?	W2
APPL_OTHER	char	Falls APPL_OTHER geklickt: Ist APPL_OTHER leer ?	E1
PROD_COMP	memo	Falls TRDNAME nicht leer: Leer ?	E1
VECTOR	memo	Falls TRDNAME nicht leer: Leer ?	W1
SAFE_PREC	memo	Falls TRDNAME nicht leer: Leer ?	W1

4.4 Mögliche Erweiterungen (Dynamisches Pflichtenheft)

Es wäre denkbar, das Programm mit folgenden Features auszubauen:

Mögliche Erweiterungen	Vorkehrungen
Hilfesystem in anderen Sprachen als nur Deutsch.	Der Text der kontextsensitiven Online-Hilfe wurde in einer freien FoxPro-Tabelle gespeichert. Der Zugriff auf den richtigen Hilfetext passiert über die sogenannten HelpContextIDs, die für jedes VisualFoxPro-Objekt festgelegt werden kann. Um den Text zu jedem Thema in einer neuen Sprache anzubieten, muss die freie Tabelle „snfhelp.dbf“ zuerst um eine weitere Spalte (Typ: MEMO-Feld) erweitert werden. Diese wird dann mit dem entsprechenden Hilfetext in der neuen Sprache beschrieben. So könnte je nach Sprachauswahl des Benutzers (zum Beispiele beim Programmstart) die entsprechende Spalte der Tabelle aktiviert werden.
Einfaches Blättern in den Formularen (zum Beispiel in chronologischer Reihenfolge)	
Ausdrucken der Formulare in der Art, wie sie heute in Gebrauch sind (6 Seiten, A4)	Es müssten „reports“ (Berichte) generiert werden, die in gedruckter Form das Aussehen der ursprünglichen Formulare haben.
Erweiterung der Fehlerbehandlung mit wachsenden Anforderungen (z.B. auch andere Sprachen)	Die Texte zu den einzelnen Fehlern / Warnungen werden in der freien Tabelle „errors.dbf“ verwaltet. Diese werden jeweils über einen Fehlerschlüssel, der dem Formularobjekt „fehler.scx“ als Parameter übergeben wird, aus der Tabelle „errors.dbf“ herausgelesen. Für neue Fehler-/ Warnmeldungen kann diese Tabelle mit weiteren Datensätzen ergänzt werden. Um den Fehlertext in anderen Sprachen anzubieten, kann ähnlich wie beim Hilfesystem vorgegangen werden.
Implementierung der Funktionalität des Befehlsschaltknopfes „Zurück“	In der Definition der Klasse „snfformclass“ sind die Eigenschaften dieses Steuerelements vorgegeben. Da jedoch dessen Funktion nicht programmiert wurde bleibt die Eigenschaft „Enabled“ die ganze Zeit auf dem Wert <i>false</i> . Die Idee war anfangs, dass mit diesem Schaltknopf der Benutzer bei der Ersterfassung der Daten auf der aktuellen Eingabemaske, eine Seite zurückblättern könnte. Dabei müssten die eingegebenen Werte zwecks Komfortsteigerung und Ergonomie temporär zwischengespeichert und wiederhergestellt werden. Der Aufwand für den Mechanismus, der vor allem bei mehrfachem Betätigen dieses Knopfes nötig wäre, schien mir jedoch für diese erste Version der Software „SNF“ übertrieben.
Ausbau zum Mehrbenutzersystem	Die Daten werden nach der Eingabe in die Datenbank geschrieben, so dass die Datensätze oder Tabellen nicht über längere Zeit gegenüber Lese- oder Schreibzugriffe anderer Benutzer geschützt und damit blockiert werden müssen. Somit könnten mehrere Benutzer gleichzeitig mit den zentralen Studiendaten arbeiten.

4.5 Erfahrungen, Danksagung

Die Folgen und die weitere Entwicklung, die aus dem Einsatz dieser Software entstehen könnten, kann ich nicht genau abschätzen.

Die Arbeit an diesem Informatik-Projekt fand ich für mich persönlich sehr sinnvoll.

Die Umsetzung des theoretischen Wissens über Datenbanksysteme und über die Techniken des Software Engineering aus den Vorlesungen des Grundstudiums in Informatik in ein praxisorientiertes, grösseres Projekt, war sehr interessant.

Eine weitere Erkenntnis war die Bestärkung der Meinung, dass ein guter Dialog zwischen den Endbenutzern (in diesem Falle Mediziner) und ausführenden Organen (mit der Realisierung beauftragte Informatiker) eine der wichtigsten Voraussetzung für die allgemeine Zufriedenheit am Resultat darstellt. In dieser Hinsicht war die Zusammenarbeit, wie ich sie erlebt habe, sehr positiv.

Einige Schwierigkeiten während der Realisierung der Software entstanden bei der Implementation der grafischen Oberfläche und der Endbenutzernavigation.

Vor allem die vielen grafischen Hilfsmittel und Möglichkeiten der Entwicklungsumgebung für die Darstellung (Bilder, Farben, usw.) verleiten zum Einbau von unnötigen optischen Elementen. Ich habe versucht, diese Komponenten dosiert einzusetzen, um keine unübersichtlichen oder verwirrenden Bildschirmmasken zu entwerfen.

In diesem Zusammenhang wären meiner Ansicht nach Kenntnisse über Ergonomie und auch Psychologie sehr hilfreich.

Weiter erwähnenswert erscheint mir die Tatsache, dass auch grosse Softwarehersteller ihre Versprechen nicht immer einhalten können.

Wegen gewissen Rahmenbedingungen der Entwicklungsumgebung (zum Beispiel die Begrenzung der Länge eines SQL-Statements auf 254 Zeichen in Microsoft's Visual FoxPro, welche übrigens nicht an der erwarteten Stelle dokumentiert war) und wegen oft unbrauchbaren, auf Syntaxfehler folgenden Fehlermeldungen, war die Produktivität etwas eingeschränkt.

Die oft in der Werbung der Softwarehersteller zitierten Produktivitätshilfsmittel, sogenannte *Wizards* (Assistenten), stellten sich als Entwicklungshilfe in der Praxis als nicht brauchbar heraus.

Hingegen war die Zusammenarbeit mit den am Projekt direkt beteiligten Personen sehr konstruktiv. Ihnen gilt auch mein spezieller Dank.

5 Anhang

5.1 *Pflichtenheft*

(Bereinigte Fassung gemäss Bemerkungen von Dr. Th. Weber, 21. Februar 1997)

5.1.1 Dokumentation

Als Dokumentation wird ein Benutzerhandbuch verlangt. Der Text soll in einem Format abgelegt werden, das mit MS-Word 6.0 gelesen werden kann.

Die Information zur Software soll eine Kontaktadresse für den Support enthalten.

5.1.2 Software

5.1.2.a Allgemeines

Als Entwicklungsumgebung wird MS FoxPro Version 2.6 (englisch) zur Verfügung gestellt. Die zu erfassenden Daten sollen auch via Email „automatisch“ erfasst werden können.

5.1.2.b Funktionalität

Folgende Hauptfunktionen sind erforderlich und werden weiter unten genauer beschrieben:

- Erfassung von neuen Studien
- Änderungen/Ergänzungen an den Daten von schon erfassten Studien
- Suchen nach Studien mit verschiedenen Suchkriterien und Ausdrucken von Suchergebnissen in Form von Listen (Anzeigen, Drucken)
- Auswertungen (Anzeigen, Drucken)

Erfassung von neuen Studien

Beim Erfassen von neuen Studien sind die nachfolgend aufgeführten Eingaben obligatorisch. Wenn zu einem mit • markierten Feld (in der nachfolgenden Aufzählung) die Angabe fehlt, wird eine entsprechende Fehlermeldung ausgegeben und eine Eingabe verlangt.

Auf Seite 1 des Formulars (Details of the Study):

- **Referenznummer** (wird vom System vorgegeben und kann nicht editiert werden)
- **Study Title**
- Mindestens eine Auswahl aus **Study Design** (prospective, retrospective, controlled, open, blind, double-blind, randomised, parallel-group, cross-over, single-center, multi-center, other)

Auf Seite 3 des Formulars (Details of Investigator):

Für Hospital, Institut/clinic, practice:

- **Hospital**
- **Address**

Für responsible investigator:

- **Name**
- **Division**

Auf Seite 4 des Formulars (Details of Ethical Review Committee (ERC))

Für responsible ethical review committee:

- **Name**
- **Address**

Für ERC decision

- **Genau eine Auswahl** aus positive decision, positive decision with recommendations, positive decision with conditions, negative decision with grounds („grounds“ ersetzen durch „reasons“)

Auf Seite 5 des Formulars

Speziell für Vaccine Identification: Study Vaccine

- **Lot-number**
- Mindestens eine Auswahl aus **Application Route** (s.c., i.m., i.v., p.o., rectal, intranasal, other)
- Mindestens eine Angabe zu **Active components** (Name der Komponente)
 - Genau eine Auswahl aus killed, living und attenuated
 - Genau eine Auswahl aus bacterial, viral und other

Speziell für Blood Product Identification: Study Product

- **Lot-number**
- Mindestens eine Auswahl aus (i.v., i.m., s.c., other)

Speziell für Product Identification: Study Product (Gene-Therapy)

- **Lot-number**
- Mindestens eine Auswahl aus **Application route** (s.c., i.m., i.v., p.o., rectal, intranasal, other)

Warnungen beim Erfassen von neuen Studien

Eine *Warnmeldung* wird angezeigt, wenn folgende, mit • markierte Informationen, fehlen:

Auf Seite 1 des Formulars (Details of the Study):

- **Study Identification Code**
- **Study Phase**
Falls hier mehr als eine Auswahl angewählt wird, soll eine Fehlermeldung angezeigt werden, ausser, wenn „Phase III“ und dann „B“ gewählt wurde. In diesem letzten Fall muss auch „Phase II“ angewählt sein.
- **Planned Study Enrollement**

Auf Seite 2 des Formulars (Detail of Study Sponsor):

Für Sponsor:

- **Company**
- **Address, Telephone und Facsimile**, falls für Company eine Angabe gemacht wurde

Für Person responsible for study:

- **Name**, falls für Company eine Angabe gemacht wurde
- **Address, Telephone und Facsimile**, falls für Name eine Angabe gemacht wurde

Auf Seite 3 des Formulars (Detail of Study Investigator):

Für Hospital und Responsible Investigator:

- **Telephone**
- **Facsimile**

Für Co-Investigators/Staff:

- **Name**
- **Division** und **Telephone**, falls für Name eine Angabe gemacht wurde

Auf Seite 4 des Formulars (Details of Ethical Review Committee):

Für Members of ERC who voted for this study:

- **Name** (chairman)
- **Name**
- **profession** und **sex**, nur falls Einträge für Name gemacht wurden

Auf Seite 5 des Formulars:

- **Trade name**
- **Code number**
- **BAG-number**

Speziell für Vaccine Identification: Study Vaccine

Für Active components:

Für die erste, obligatorische Komponente:

- Eine **Auswahl** aus recombinant antigen, recombinant microorganism und DNA

Für zweite und weitere Komponenten:

- **Name** der Komponente
- Eine **Auswahl** je aus
 - killed/living/attenuated,
 - bacterial/viral/other
 - recombinant antigen/recombinant microorganism/DNAfalls eine Angabe für den Namen der Komponente gemacht wurde

Speziell für Vaccine Identification: Comparison Vaccine

- **Trade name**
- **Andere Angaben** auf dieser Seite, falls für Trade name eine Angabe gemacht wurde

Die Bedingungen für die speziellen Seiten für *Blood Product Identification* und *Product Identification (Gene-Therapy)* entsprechen denen der Vaccine Identification.

Änderungen/Ergänzungen an den Daten von schon erfassten Studien

Hier gelten die gleichen Bedingungen wie schon bei der Erfassung neuer Studien, d.h. sobald eine Änderung vorgenommen wird, die die oben erwähnten Bedingungen hinsichtlich obligatorischer Eingaben oder Warnmeldungen nicht erfüllt, soll der Benutzer oder die Benutzerin darauf aufmerksam gemacht werden.

Suchen nach Studien

Die Suchkriterien, gemäss welchen bestimmte Studien gesucht werden sollen sind die folgenden:

- Study Identification Code
- Company (Sponsor)
- Company (Manufacturer)
- Company (Clinical Research Organization)
- Name (Person responsible for study)
- Hospital (Hospital, institute / clinic, practice)
- Name (Responsible investigator)
- Name (Responsible ERC)
- Trade name
- Code number
- BAG-number
- Lot-number
- Referenznummer

Am Ende der Suche soll eine Liste von Referenznummern der gefundenen Studien angezeigt werden, die dann auf Wunsch gedruckt werden kann.

Als Resultat der Suche mit der Referenznummer als Kriterium soll eine Liste mit folgenden Informationen gedruckt werden können: study title, study Identification Code, company, investigator name.

5.2 Abklärung der einzusetzenden Entwicklungsumgebung für die Datenbankprojekte „Impfstoffnebenwirkungen“ und „Notifikationsformulare“

Die zur Verfügung stehende Datenbanksoftware ist Microsoft FoxPro in den Versionen

- FoxPro 2.6 für DOS
- FoxPro 2.6 für Windows
- Visual FoxPro 3.0

Für die Versionen FoxPro 2.6 für DOS und Visual FoxPro 3.0 Professional existieren die sogenannten **Distribution Kits**. Damit können selbständig lauffähige „*.exe“ - **Programme** generiert werden.

Die neuste, zur Verfügung stehende Version der FoxPro-Reihe (Visual FoxPro 3.0) ist zugleich auch die flexibelste von den drei Versionen.

Grund dafür ist vor allem die Möglichkeit der objektorientierten Programmierung.

Zudem ist die komplexere, aber funktionale, **GUI** ausgereifter als die der älteren Versionen und steigert Dank der übersichtlicheren Darstellung die **Wartbarkeit**.

In bezug auf die **Stabilität** sind wohl alle drei Versionen einander ebenbürtig, wobei die Version 3.0 auf der grössten Betriebssystemplattform nutzbar ist. Diese ist auf Windows 3.x, Windows 95 und Windows NT lauffähig.

Die **Konversion** von Anwendungen (wichtig für das Projekt „VAE“ von Frau Portmann), die mit der Version 2.6 erstellt wurden, in die Version 3.0 erfolgt im Prinzip automatisch. Dies bedingt

jedoch, dass die Anwendungen, die mit der Version 2.6 erstellt wurden, als Projekte vorliegen (und nicht als einzelne, kompilierte Programmdateien).

Das FoxPro2.6-Projekt muss die zum Projekt gehörenden Dateien in ihrer ursprünglichen Form enthalten. Das heisst zum Beispiel, dass eine Bildschirmmaske (screen, form) als Datei der ursprünglichen Form (*.scx) vorliegt, und nicht schon als vom FoxPro-Screen-Wizzard generiertes Programm im Datenformat „*.spr“ oder gar als kompiliertes Programm in einer Datei mit Extension „.spx“.

Als ein Nachteil ist die geringere *Ausführungsgeschwindigkeit* der einzelnen Aktionen der Visual-Version während der **Entwicklung** zu erwähnen (vor allem auf einem Rechner mit i486-Prozessor). Dies ist womöglich eine Auswirkung des grösseren Overheads infolge der OO-Funktionalität.

Dieses Manko ist jedoch beim Einsatz eines Rechners der Pentiumklasse nicht von Bedeutung. Die grössere Funktionsvielfalt, welche die Version 3.0 von FoxPro zur Verfügung stellt, mit der auch eher zeitaufwendige Programmieraufgaben (zum Beispiel Implementierung eines Online-Hilfesystems) in kurzer Zeit erledigt werden können, macht diesen Nachteil zu einem grossen Teil wieder wett.

Ein weiterer Vorteil der *Visual-Version* ist die Verwaltung der zusammengehörenden Tabellen als *Datenbank* und nicht wie in den früheren Versionen als *Ansammlung von Tabellen*. Dadurch wird neu auch ein einfacher Mechanismus zur Überwachung der Datenintegrität bereitgestellt.

5.3 *Hilfesystem der Software*

Hauptmenü des Datenbankprogramms für Study Notification Forms	Hier können Sie die gewünschte Funktion wählen, die ausgeführt werden soll. Die Beschreibungen auf den Auswahlkosten sollten selbsterklärend sein. Die gewünschte Funktion kann auch durch den in den Tastenbeschriftungen markierten Buchstaben (gleichzeitiges Drücken mit der <ALT>-Taste, links neben der Leerschlagtaste) aktiviert werden. Beispiel: Um das Programm zu beenden <ul style="list-style-type: none"> • klicken Sie die unterste Taste mit der Aufschrift "Beenden" mit der Maus oder • drücken und halten Sie die <ALT>-Taste und drücken die Taste .
Eingabe des Typs der Studie	Hier muss eine von drei Möglichkeiten ausgewählt werden, bevor es weitergeht.
Fehlermeldung ("Rote Karte")	Irgendwo wird eine Eingabe von Ihnen verlangt, die obligatorisch ist, das heisst, ohne diese Information macht es keinen Sinn, die Studie in die Datenbank aufzunehmen. Versuchen Sie die Fehlermeldung zu verstehen und klicken Sie erst dann auf die Befehlstaste "OK".
Erfassung von neuen Studien	In diesem Teil des Programms können Sie die Daten zu einer Studie erfassen. Die Informationen, die zu einer Studie gehören, sind auf sechs Seiten thematisch verteilt.

<p>STUDY DETAILS (Seite 1 des Formulars)</p>	<p>Auf dieser Seite werden die Details zur Studie erfasst / editiert.</p> <p>In den Feldern 'Referenznummer', 'Study Title' und 'Study Design' MÜSSEN Informationen eingegeben werden. Alle anderen Feldern KÖNNEN leer bleiben.</p> <p>Wenn Letzteres zutrifft, wird für jedes leere Feld ein Hinweis eingeblendet, der wie folgt quitiert werden muss: "Schon gut...": Das leere Feld bleibt leer und wird vom Programm nicht mehr beanstandet. "Zurück": Die Einfügemarke wird auf das betreffende Feld gesetzt, so dass die vergessene Eingabe nachgeholt werden kann.</p> <p>N.B.: Die Referenznummer der neuen Studie wird vom Programm berechnet und vorgegeben, kann aber verändert werden. Jeder Studie muss eine eindeutige Referenznummer zugewiesen werden, um eine datenbanktechnisch korrekte Verwaltung zu gewährleisten.</p>
<p>Detail of Sponsor (Seite 2 des Formulars)</p>	<p>Auf dieser Seite werden die Informationen zum Sponsor erfasst.</p> <p>Alle Feldern können leer bleiben. Wenn die Angaben zu 'Sponsor' oder zu 'Responsible Person' fehlen oder unvollständig sind, wird jeweils eine Warnung eingeblendet, die wie folgt quitiert werden muss:</p> <ul style="list-style-type: none">• "Schon gut...": Die leeren Felder bleiben leer und werden vom Programm nicht mehr beanstandet.• "Zurück": Setzt die Einfügemarke auf das betreffende Feld, so dass die vergessene Eingabe nachgeholt werden kann.

Details of Investigator (Seite 3 des Formulars)	<p>Auf dieser Seite werden die Informationen zur Datenquelle (Spital, Klinik) erfasst.</p> <p>Die Angaben zu 'Hospital' beziehungsweise zu 'Responsible Investigator' und die zugehörigen Adressen und die Abteilungen ('Division') müssen gemacht werden. Alle anderen Feldern können leer bleiben.</p> <p>Wenn Letzteres zutrifft, wird für jedes leere Feld ein Hinweis eingeblendet, der wie folgt quittiert werden muss:</p> <ul style="list-style-type: none">• "Schon gut...": Die leeren Felder bleiben leer und werden vom Programm nicht mehr beanstandet.• "Zurück": Setzt die Einfügemarke auf das betreffende Feld, so dass die vergessene Eingabe nachgeholt werden kann. <p>Hinweis zu 'Co-Investigator': Jedes 'Study Center' kann mehrere Co-Investigators haben. Um mehr als einen Co-Investigator einzugeben, muss nach der Eingabe der Daten für einen Co-Investigator die Befehlsschalttaste "More Co-Investigators / Staff Members" geklickt werden, damit die Eingabefelder für die Angaben zum nächsten Co-Investigator vorbereitet werden.</p>
Details of ERC (Seite 4 des Formulars)	<p>Auf dieser Seite werden die Informationen zur jeweiligen Ethischen Kommission der Forschungsstellen erfasst.</p> <p>Die Informationen zur "Responsible Ethical Review Commitee" (Name und Adresse) MÜSSEN eingetragen werden.</p> <p>Aus der Auswahl für "ERC Decision" MUSS ein möglicher Entscheid selektiert werden.</p> <p>Ausserdem MUSS das Datum des Beschlusses angegeben werden.</p> <p>Alle anderen Feldern KÖNNEN leer bleiben.</p> <p>Wenn Letzteres zutrifft, wird für jedes leere Feld ein Hinweis eingeblendet, der wie folgt quittiert werden muss:</p> <ul style="list-style-type: none">• "Schon gut...": Die leeren Felder bleiben leer und werden vom Programm nicht mehr beanstandet.• "Zurück": Setzt die Einfügemarke auf das betreffende Feld, so dass die vergessene Eingabe nachgeholt werden kann. <p>Hinweis zu 'Other Members': Jedes 'ERC' kann ausser dem Vorsitzenden ('Chairman') und dem für die Statistik verantwortlichen Mitglied weitere Mitglieder haben. Um mehr als ein weiteres Mitglied einzugeben, muss nach der Eingabe der Daten für ein Mitglied ('Other Member') die Befehlstaste "More Members" geklickt werden, damit die Eingabefelder für die Angaben zum nächsten Mitglied vorbereitet werden.</p>

Approval des ERC	Hier können Bemerkungen, Empfehlungen, Bedingungen und Erläuterungen zum Entscheid der ethischen Kommission notiert werden.
Vaccine Identification (Seiten 5 und 6 des Formulars für Impfstoffe)	<p>Obligatorische Angaben für den STUDIENImpfstoff sind:</p> <ul style="list-style-type: none"> • 'Lot Number' • 'Application Route' (mindestens eine aus der Angezeigten Auswahl) • 'Active Components' (mindestens eine aktive Komponente) <p>Zur ersten aktiven Komponente müssen auch je eine Auswahl aus 'Killed/Living/Attenuated' und 'Bacterial/Viral/Other' selektiert werden.</p> <p>Mit einem Klick auf die Befehlstaste "More Active Components" werden die Daten zur aktiven Komponente gespeichert und die Eingabefelder für die nächste einzugebende Komponente vorbereitet.</p> <p>Ein VERGLEICHImpfstoff kann angegeben werden. Dazu genügt die Angabe des Handelsnamen ('Trade Name').</p>
Blood Product Identification (Seiten 5 und 6 des Formulars für Blutprodukte)	<p>Obligatorische Angaben für das Studienprodukt sind:</p> <ul style="list-style-type: none"> • 'Lot Number' • 'Application Route' (mindestens eine aus der Angezeigten Auswahl) <p>Ein Vergleichsprodukt KANN angegeben werden. Dazu genügt die Angabe des Handelsnamen ('Trade Name').</p>
Product Identification - Gene Therapy (Seiten 5 und 6 des Formulars für Gentherapien)	<p>Obligatorische Angaben für das Studienprodukt sind:</p> <ul style="list-style-type: none"> • 'Lot Number' • 'Application Route' (mindestens eine aus der Angezeigten Auswahl) <p>Ein Vergleichsprodukt KANN angegeben werden. Dazu genügt die Angabe des Handelsnamen ('Trade Name').</p>

Suchen	<p>Auf dieser Seite können die gewünschten Suchkriterien festgelegt werden.</p> <p>Für die Suche in der Datenbank werden dabei diejenigen Kriterien verwendet, die auch markiert sind.</p> <p>Die Kriterien werden mit einer logischen UND-Operation verknüpft, d. h. wenn mit mehr als einem Kriterium gesucht wird, werden diejenigen Studien angezeigt, welche alle aktivierten Bedingungen (Kriterien) erfüllen.</p> <p>Gross-/Kleinschreibung spielt keine Rolle. Um die Tabelle mit den gefundenen Referenznummern der Studien wieder zu verlassen drücken Sie bitte die Taste <ESC>.</p> <p>Beispiel: Das Feld 'Sponsor' enthalte "XYZ AG" (ohne Anführungs- und Schlussstriche!) Das Feld 'Hospital' enthalte "World Hospital". Wenn beide Kriterien aktiviert werden (mittels zugehörigen Check-Boxes), sucht das Programm aus den erfassten Studien, diejenigen heraus, in denen der Sponsor UND das Spital die entsprechenden Einträge enthalten. Falls keine Studie gefunden wurde, kann z. B. nach deaktivieren der Check-Box für 'Hospital' eine neue Suche gestartet werden, welche jetzt nur noch alle Studien mit dem Eintrag "XYZ AG" für 'Sponsor' sucht.</p> <p>Die Suchbegriffe müssen nicht vollständig eingegeben werden: bei Angabe eines unvollständigen Wortes wird nur nach diesem Fragment gesucht</p> <ul style="list-style-type: none">• anstelle <i>eines</i> Buchstabens kann ein Fragezeichen geschrieben werden• anstelle <i>einer unbestimmten Anzahl</i> Buchstaben kann ein Stern * eingesetzt werden <p>Beispiele:</p> <ul style="list-style-type: none">• Mit der Zeichenfolge "su" werden alle Einträge gefunden, die irgendwo die Zeichenfolge "su" enthalten.• Mit der Zeichenfolge "su??" werden alle Einträge gefunden, die aus insgesamt vier Zeichen bestehen und mit "su", "Su", "sU" oder "SU" anfangen.• Mit der Zeichenfolge "s*u?" werden alle Einträge gefunden, die mit einem "s" anfangen und an zweitletzter Stelle ein "u" haben.
Studiendaten ändern	<p>Diese Funktion erlaubt Ihnen, Änderungen an den bestellenden Daten vorzunehmen.</p> <p>Mit den Funktionstasten am unteren Rand des Bildschirms kann in der Datenbank geblättert werden.</p> <p>Wenn die zu verändernde Studie gefunden wurde, muss auf "Bearbeiten" geklickt werden, bevor die Daten verändert werden können.</p>

5.4 Screen shots



Abbildung 1: Formularobjekt "info.scx"

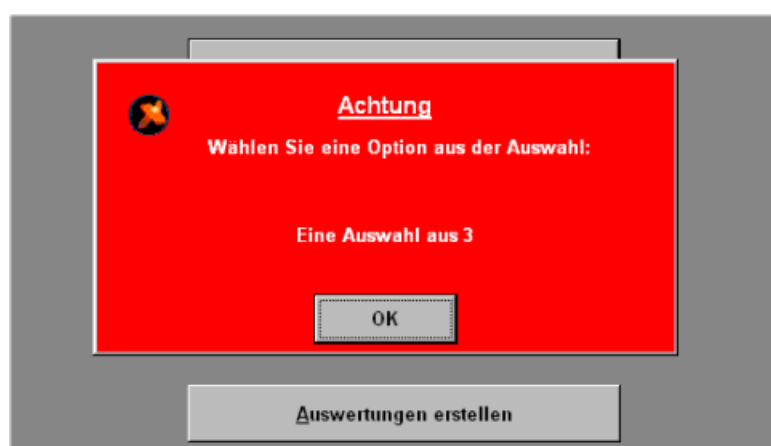


Abbildung 2: Formularobjekt "fehler.scx"

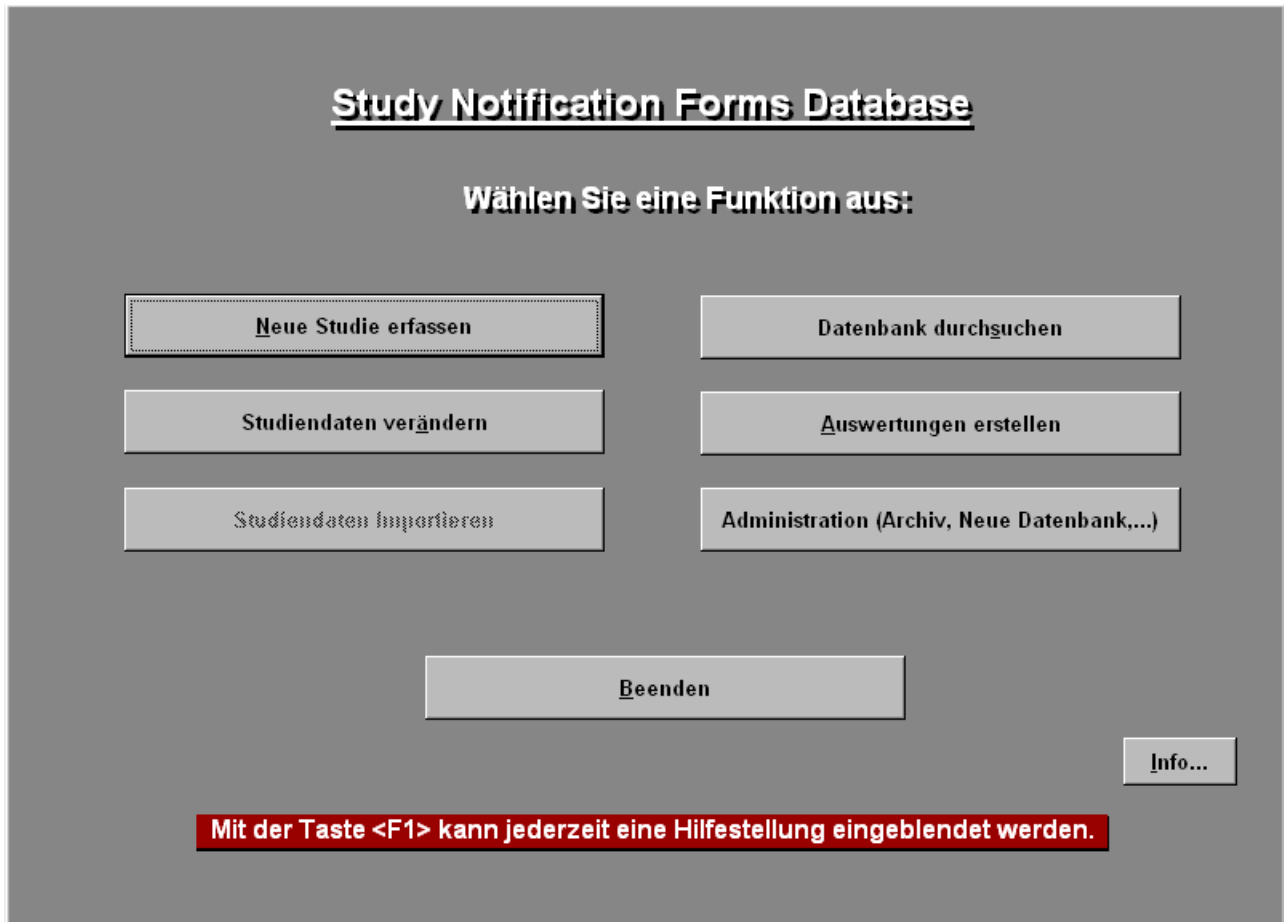


Abbildung 3: Formularobjekt "main_fcts.scx"

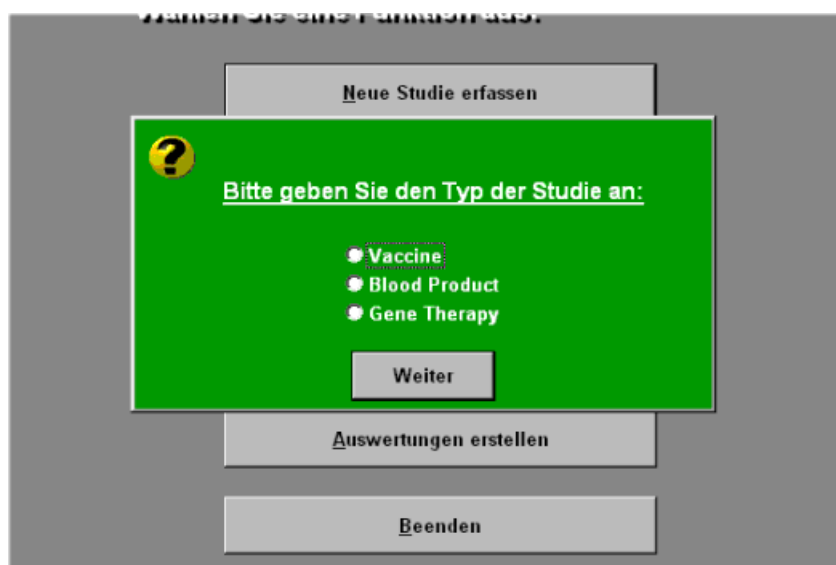


Abbildung 4: Formularobjekt "eingabe.scx"

The screenshot shows a web browser window with the title bar 'Gewählte Funktion: NEUE STUDIE ERFASSEN'. The main content area is titled 'DETAILS OF THE STUDY' and contains several input fields and sections:

- Referenznummer:** A dropdown menu set to 'VA' and a text box containing '1997001'.
- Study Identification Code:** An empty text input field.
- Study Title:** A large empty text area with a vertical scrollbar.
- Study Phase:** A group of radio buttons with options: Phase I, Phase II, Phase III, Phase IV, and PMS.
- Planned Study Enrollment:** Two sets of date pickers. The first is for 'start date' and the second for 'end date', both currently set to '0' for month and '0' for year.
- Study Design:** A group of radio buttons and checkboxes with options: prospective, retrospective, controlled, open, blind, double-blind, randomised, parallel-group, cross-over, single-center, multicenter, and other.

At the bottom of the form, there are several buttons: 'Zurück', 'Weiter', 'Abbrechen', 'Ton ausschalten', and 'Info...'.

Abbildung 5: Formularobjekt "seite1.scx"

The screenshot shows a yellow warning dialog box with a black exclamation mark icon in the top-left corner. The text inside the dialog reads:

Achtung
Sie haben in diesem Feld nichts eingegeben:
(Study Identification Code)

At the bottom of the dialog are two buttons: 'Zurück' and 'Schon gut...'. The background shows a portion of the form from the previous image, including the 'Planned Study Enrollment' and 'Study Design' sections.

Abbildung 6: Formularobjekt "warnung.scx"

Gewählte Funktion: NEUE STUDIE ERFASSEN

DETAILS OF STUDY SPONSOR

For studies without a sponsor, the investigator assumes the responsibilities of the sponsor

Sponsor Company

Addr1 State
Addr2 Telephone
Addr3 Facsimile

Manufacturer Company (if different from sponsor)

Addr1 State
Addr2
Addr3

Clinical Research Organization (CRO)

Company

Addr1 State
Addr2 Telephone
Addr3 Facsimile

Person Responsible for Study

Name State
Addr1 Phone direct
Addr2 Fax direct
Addr3

Zurück Weiter Abbrechen Ton ausschalten Info...

Abbildung 7: Formularobjekt "seite2.scx"

Studien suchen

Legen Sie hier bitte die Suchkriterien fest:

Study Identification Code

Sponsor

Manufacturer

Clinical Research Organization

Person Responsible For Study

Hospital, Institute, Clinic

Responsible Investigator

Responsible ERC

Trade Name

Code Number

BAG Number

Lot Number

Referenznummer -

Suchen Abbrechen

Abbildung 8: Formularobjekt "suchen.scx"

Gewählte Funktion: NEUE STUDIE ERFASSEN

Details of Investigator

Center 1 of 1Center(s)

Hospital, Institute / Clinc, Practice

Hospital

Division

Addr1 State

Addr2 Telephone

Addr3 Facsimile

Responsible Investigator

Name

Division

Telephone direct Facsimile direct

Co-Investigators / Staff Members Directly Involved With The Study

Name

Division

Telephone (direct)

Ton

Abbildung 9: Formularobjekt "seite3.scx"

Auswertungen

- Anzahl Studien je Studienphase
- Liste der Studien je Design-Kategorie
- Studien, die noch nicht abgeschlossen sind
- Sponsoren pro Jahr und über alle Studien
- CRO's pro Jahr und über alle Studien
- Studien, die zu einem Produkt gehören
- Produkte, die in der Studiendatenbank erfasst sind
- Studien, die zu einer Lot Number gehören
- Lot Numbers, die in der Studiendatenbank erfasst sind

Ton

Abbildung 10: Formularobjekt "auswert.scx"

Gewählte Funktion: NEUE STUDIE ERFASSEN

Details of the Ethical Review Committee (ERC)

Responsible Ethical Review Committee of study center ss

Name

Addr1

Addr2

Addr3 State

Members of ERC who Voted for this Study

Chairman

Profession sex female
 male

Person Responsible for Statistics

Name

Profession sex female
 male

Other Members

Name

Profession sex female
 male

ERC decision

Date
(ex.: 23 04 1997)

positive decision
 positive decision with recommendations
 positive decision with conditions
 negative decision with reasons

(Ethical Committee Approval)

Abbildung 11: Formularobjekt "seite4.scx"

Ethical Committee Approval (ECA)

Comments, recommendations, conditions or reasons

Wissenschaftliche Beurteilung eines Genthrapie-Protokolles durch die SKBS/CSSB

Sehr geehrter Herr Professor Äbischer

Die SKBS hat Ihren Antrag "Phase I clinical protcol for patients suffering from amyotrophic lateral"

Abbildung 12: Formularobjekt "seite4b.scx"

Gewählte Funktion: NEUE STUDIE ERFASSEN

Vaccine Identification: Study Vaccine

Trade Name

Code No.

BAG No.

Lot No.

Application Route

s.c. i.m. i.v. p.o. rectal intranasal other:

Active Components

Components Name

killed
 living
 attenuated

bacterial
 viral
 other

recombinant antigen
 recombinant microorganism
 DNA

Ton

Abbildung 13: Formularobjekt "seite5va.scx"

Resultat der Suche

Referenznummer	Identification Code
VA 1996001	MeMuRu-124 (209762/12)
VA 1996002	KV 9610
GT 1997001	

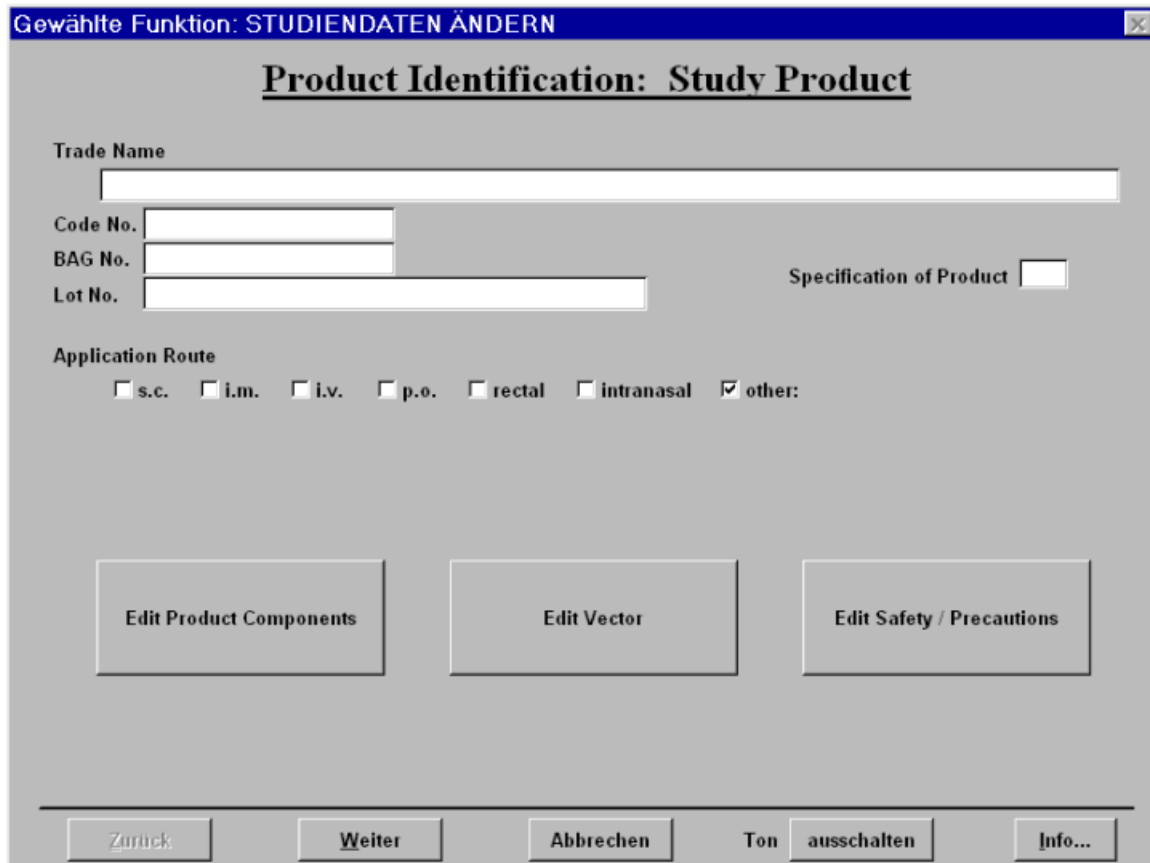
Klicken Sie die Zeile der gewünschten Studie an.

Zum Hauptmenü
Wählen Sie <Abbrechen>, um zum Hauptmenü des Programms zu gelangen.

Andere Studie suchen
Wählen Sie <Zurück>, um neue Kriterien für die Suche nach einer anderen Studie festzulegen.

Ton

Abbildung 14: Formularobjekt "showres.scx"



Gewählte Funktion: STUDIENDATEN ÄNDERN

Product Identification: Study Product

Trade Name

Code No.

BAG No.

Lot No.

Specification of Product

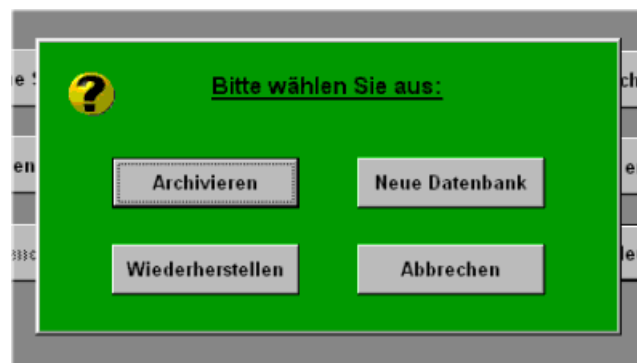
Application Route

s.c. i.m. i.v. p.o. rectal intranasal other:

Edit Product Components Edit Vector Edit Safety / Precautions

Zurück Weiter Abbrechen Ton ausschalten Info...

Abbildung 15: Formularobjekt "seite5bp.scx"



Bitte wählen Sie aus:

Archivieren Neue Datenbank

Wiederherstellen Abbrechen

Abbildung 16: Formularobjekt "zipunzip.scx"

5.5 Beispiele für Programmcode der Software „SNF“

5.5.1 Methode `click()` der Befehlsschaltfläche `cmd_forw` im Formularobjekt `auswert.scx`

```
* Code:      auswert.cmd_forw.Click()
* Author:    Sani Tetik
* Change:    October 2, 1997
*
** SQL-Statements gemaess Benutzereinstellungen ausführen.
** Anzeigen, Drucken der Resultate

WITH THISFORM
DO CASE
CASE .ausw_typ.Value=1
  **Auswertung nach Studienphasen
  SELECT dt_phase, COUNT (DISTINCT ref_nbr);
  FROM study ;
  GROUP BY dt_phase ;
  ORDER BY dt_phase ASC ;
  INTO CURSOR ausw1
  IF RECCOUNT(>)>0
    REPORT FORM ausw1 PREVIEW
    REPORT FORM ausw1 NOCONSOLE TO PRINTER PROMPT
    USE
  ELSE
    DO FORM meldung WITH "Keine Studien gefunden."
  ENDIF
CASE .ausw_typ.Value=2
  **Auswertung nach Design-Kategorie
  SELECT std_type, ref_nbr ;
  FROM study ;
  WHERE ds_pros ;
  ORDER BY std_type, ref_nbr ;
  INTO CURSOR ausw2
  .txt_param="PROSPECTIVE"
  IF RECCOUNT(>)>0
    REPORT FORM ausw2 PREVIEW
    REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
    USE
  ELSE
    DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
  ENDIF
  SELECT std_type, ref_nbr ;
  FROM study ;
  WHERE ds_retro ;
  ORDER BY std_type, ref_nbr ;
  INTO CURSOR ausw2
  .txt_param="RETROSPECTIVE"
  IF RECCOUNT(>)>0
    REPORT FORM ausw2 PREVIEW
    REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
    USE
  ELSE
    DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
  ENDIF
  SELECT std_type, ref_nbr ;
  FROM study ;
  WHERE ds_contr ;
  ORDER BY std_type, ref_nbr ;
  INTO CURSOR ausw2
  .txt_param="CONTROLLED"
  IF RECCOUNT(>)>0
    REPORT FORM ausw2 PREVIEW
    REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
    USE
  ELSE
    DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
  ENDIF
  SELECT std_type, ref_nbr ;
  FROM study ;
  WHERE ds_open ;
  ORDER BY std_type, ref_nbr ;
  INTO CURSOR ausw2
  .txt_param="OPEN"
  IF RECCOUNT(>)>0
```

```
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE ds_blind ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="BLIND"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE ds_dblind ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="DOUBLE-BLIND"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE ds_rando ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="RANDOMISED"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE ds_parall ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="PARALLEL-GROUP"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE ds_cross ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="CROSS-OVER"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
FROM study ;
WHERE !ds_multic ;
ORDER BY std_type, ref_nbr ;
INTO CURSOR ausw2
.txt_param="SINGLE-CENTER"
IF RECCOUNT(>)>0
REPORT FORM ausw2 PREVIEW
REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
```

```

ELSE
    DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
SELECT std_type, ref_nbr ;
    FROM study ;
    WHERE ds_multic ;
    ORDER BY std_type, ref_nbr ;
    INTO CURSOR ausw2
.txt_param="MULTICENTER"
IF RECCOUNT(>)>0
    REPORT FORM ausw2 PREVIEW
    REPORT FORM ausw2 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
    DO FORM meldung WITH "Keine Studien mit Design-Typ: "+.txt_param
ENDIF
CASE .ausw_typ.Value=3
*Auswertung: Studien, die noch nicht abgeschlossen sind
SELECT std_type, ref_nbr, ds_endm, ds_endy ;
    FROM study ;
    WHERE ds_endy>YEAR(DATE()) OR (ds_endy=YEAR(DATE()) AND ds_endm>MONTH(DATE())) ;
    ORDER BY ds_endy, ds_endm ;
    INTO CURSOR ausw3
IF RECCOUNT(>)>0
    REPORT FORM ausw3 PREVIEW
    REPORT FORM ausw3 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
    DO FORM meldung WITH "Alle Studien sind schon beendet."
ENDIF
CASE .ausw_typ.Value=4
*Auswertung: Sponsor
SELECT study.std_type, study.ref_nbr, sponsor.sp_comp, study.ds_starty ;
    FROM study, std_spon, sponsor ;
    WHERE study.ref_nbr=std_spon.ref_nbr AND sponsor.sp_id=std_spon.sp_id ;
    ORDER BY study.ds_starty, sponsor.sp_comp ;
    INTO CURSOR ausw4
IF RECCOUNT(>)>0
    REPORT FORM ausw4 PREVIEW
    REPORT FORM ausw4 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
    DO FORM meldung WITH "Nichts gefunden."
ENDIF
CASE .ausw_typ.Value=5
*Auswertung: CRO
SELECT study.std_type, study.ref_nbr, cro.cro_comp, study.ds_starty ;
    FROM study, std_cro, cro ;
    WHERE study.ref_nbr=std_cro.ref_nbr AND cro.cro_id=std_cro.cro_id ;
    ORDER BY study.ds_starty, cro.cro_comp ;
    INTO CURSOR ausw5
IF RECCOUNT(>)>0
    REPORT FORM ausw5 PREVIEW
    REPORT FORM ausw5 NOCONSOLE TO PRINTER PROMPT
USE
ELSE
    DO FORM meldung WITH "Nichts gefunden."
ENDIF
CASE .ausw_typ.Value=6
*Auswertung: Studien, die zu einem Trade Name gehoeren
SELECT a.std_type, a.ref_nbr, b.trdname ;
    FROM study a, vaccine b ;
    WHERE a.ref_nbr=b.ref_nbr AND b.study_flag ;
    ORDER BY b.trdname, a.ref_nbr ;
    INTO CURSOR temp1
SELECT a.std_type, a.ref_nbr, b.trdname ;
    FROM study a, bp b ;
    WHERE a.ref_nbr=b.ref_nbr AND b.study_flag ;
    ORDER BY b.trdname, a.ref_nbr ;
    INTO CURSOR temp2
SELECT a.std_type, a.ref_nbr, b.trdname ;
    FROM study a, gt b ;
    WHERE a.ref_nbr=b.ref_nbr AND b.study_flag ;
    ORDER BY b.trdname, a.ref_nbr ;
    INTO CURSOR temp3
SELECT * ;
    FROM temp1 ;
    INTO CURSOR ausw6 ;
    UNION ALL ;
    (SELECT * ;
        FROM temp2 ;

```

```
        UNION ALL ;
        (SELECT * ;
          FROM temp3))
IF RECCOUNT(>)>0
  REPORT FORM ausw6 PREVIEW
  REPORT FORM ausw6 NOCONSOLE TO PRINTER PROMPT
  USE
  SELECT temp1
  USE
  SELECT temp2
  USE
  SELECT temp3
  USE
ELSE
  DO FORM meldung WITH "Nichts gefunden."
ENDIF
CASE .ausw_typ.Value=7
  *Auswertung: Trade Name, die in DB vorkommen
  SELECT a.std_type, a.ref_nbr, b.trdname ;
  FROM study a, vaccine b;
  WHERE a.ref_nbr=b.ref_nbr ;
  ORDER BY b.trdname ;
  INTO CURSOR temp1
  SELECT a.std_type, a.ref_nbr, b.trdname ;
  FROM study a, bp b;
  WHERE a.ref_nbr=b.ref_nbr ;
  ORDER BY b.trdname ;
  INTO CURSOR temp2
  SELECT a.std_type, a.ref_nbr, b.trdname ;
  FROM study a, gt b ;
  WHERE a.ref_nbr=b.ref_nbr ;
  ORDER BY b.trdname ;
  INTO CURSOR temp3
  SELECT * ;
  FROM temp1 ;
  INTO CURSOR ausw7 ;
  UNION ;
  (SELECT * ;
    FROM temp2 ;
    UNION ;
    (SELECT * ;
      FROM temp3))
IF RECCOUNT(>)>0
  REPORT FORM ausw7 PREVIEW
  REPORT FORM ausw7 NOCONSOLE TO PRINTER PROMPT
  USE
  SELECT temp1
  USE
  SELECT temp2
  USE
  SELECT temp3
  USE
ELSE
  DO FORM meldung WITH "Nichts gefunden."
ENDIF
CASE .ausw_typ.Value=8
  *Auswertung: Studien, die zu einer Lot-Nummer gehoeren
  SELECT study.std_type, study.ref_nbr, vaccine.lotnbr ;
  FROM study , vaccine ;
  WHERE study.ref_nbr=vaccine.ref_nbr AND vaccine.study_flag ;
  ORDER BY vaccine.lotnbr, study.ref_nbr ;
  INTO CURSOR temp1
  SELECT study.std_type, study.ref_nbr, bp.lotnbr ;
  FROM study , bp ;
  WHERE study.ref_nbr=bp.ref_nbr AND bp.study_flag ;
  ORDER BY bp.lotnbr, study.ref_nbr ;
  INTO CURSOR temp2
  SELECT study.std_type, study.ref_nbr, gt.lotnbr ;
  FROM study , gt ;
  WHERE study.ref_nbr=b.ref_nbr AND gt.study_flag ;
  ORDER BY gt.lotnbr, study.ref_nbr ;
  INTO CURSOR temp3
  SELECT std_type, ref_nbr, lotnbr ;
  FROM temp1 ;
  INTO CURSOR ausw8 ;
  UNION ALL ;
  (SELECT std_type, ref_nbr, lotnbr ;
    FROM temp2 ;
    UNION ALL ;
    (SELECT std_type, ref_nbr, lotnbr ;
      FROM temp3))
```



```
IF RECCOUNT(>0
  REPORT FORM ausw8 PREVIEW
  REPORT FORM ausw8 NOCONSOLE TO PRINTER PROMPT
  USE
  SELECT temp1
  USE
  SELECT temp2
  USE
  SELECT temp3
  USE
ELSE
  DO FORM meldung WITH "Nichts gefunden."
ENDIF
CASE .ausw_typ.Value=9
  *Auswertung: Lot-Nummern, die in DB vorkommen
  SELECT study.std_type, study.ref_nbr, vaccine.lotnbr ;
  FROM study , vaccine ;
  WHERE study.ref_nbr=vaccine.ref_nbr ;
  ORDER BY vaccine.lotnbr ;
  INTO CURSOR temp1
  SELECT study.std_type, study.ref_nbr, bp.lotnbr ;
  FROM study , bp ;
  WHERE study.ref_nbr=bp.ref_nbr ;
  ORDER BY bp.lotnbr ;
  INTO CURSOR temp2
  SELECT study.std_type, study.ref_nbr, gt.lotnbr ;
  FROM study , gt ;
  WHERE study.ref_nbr=gt.ref_nbr ;
  ORDER BY gt.lotnbr ;
  INTO CURSOR temp3
  SELECT * ;
  FROM temp1 ;
  INTO CURSOR ausw9 ;
  UNION ;
  (SELECT * ;
   FROM temp2 ;
   UNION ;
   (SELECT * ;
    FROM temp3))
IF RECCOUNT(>0
  REPORT FORM ausw9 PREVIEW
  REPORT FORM ausw9 NOCONSOLE TO PRINTER PROMPT
  USE
  SELECT temp1
  USE
  SELECT temp2
  USE
  SELECT temp3
  USE
ELSE
  DO FORM meldung WITH "Nichts gefunden."
ENDIF

ENDCASE
ENDWITH
```

5.5.2 Methode `click()` der Befehlsschaltfläche `cmd_such` im Formularobjekt `suchen.scx`

```
* Code:      suchen.cmd_such.Click()
* Author:    Sani Tetik
* Change:    September 5, 1997
*
** SQL-Abfragen, Resultat der Suche anzeigen, drucken

DIMENSION resArr(13)  &&Array fuer Resultatstabellennamen (fuer 13 Kriterien)
LOCAL i, maxi        &&Index im Array resArr, Anzahl erzeugte Resultatstabellen
LOCAL give_up
give_up=.F.
i=0

**Suchen und Resultate in Tabellen speichern
susp
IF !give_up AND THISFORM.chk_idcode.Value=1
  SELECT std_type, ref_nbr, dt_code ;
  FROM study ;
  WHERE UPPER(THISFORM.kr_idcode.Value) $ UPPER(dt_code) ;
  NOCONSOLE ;
  INTO TABLE code_res
```

```
IF RECCOUNT(>0
  i=i+1
  resArr(i)="code_res"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_spons.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM sponsor a, std_spon b, study c ;
WHERE (UPPER(THISFORM.kr_spons.Value) $ UPPER(a.sp_comp)) ;
      AND (a.sp_id=b.sp_id) AND (c.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE spons_re
IF RECCOUNT(>0
  i=i+1
  resArr(i)="spons_re"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_manuf.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM manufact a, std_manu b, study c ;
WHERE (UPPER(THISFORM.kr_manuf.Value) $ UPPER(a.m_comp)) ;
      AND (a.m_id=b.m_id) AND (c.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE manuf_r
IF RECCOUNT(>0
  i=i+1
  resArr(i)="manuf_r"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_cro.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM cro a, std_cro b, study c ;
WHERE (UPPER(THISFORM.kr_cro.Value) $ UPPER(a.cro_comp)) ;
      AND (a.cro_id=b.cro_id) AND (c.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE cro_res
IF RECCOUNT(>0
  i=i+1
  resArr(i)="cro_res"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_pers.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM person a, std_pers b, study c ;
WHERE (UPPER(THISFORM.kr_pers.Value) $ UPPER(a.p_name)) ;
      AND (a.p_id=b.p_id) AND (c.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE pers_res
IF RECCOUNT(>0
  i=i+1
  resArr(i)="pers_res"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_hosp.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM hospital a, std_hosp b, study c ;
WHERE (UPPER(THISFORM.kr_hosp.Value) $ UPPER(a.h_name)) ;
      AND (a.h_id=b.h_id) AND (c.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE hosp_res
IF RECCOUNT(>0
  i=i+1
  resArr(i)="hosp_res"
ELSE
  give_up=.T.
ENDIF
ENDIF
```

```
ENDIF

IF !give_up AND THISFORM.chk_inv.Value=1
SELECT c.std_type, c.ref_nbr, c.dt_code ;
FROM investig a, std_hosp b, study c ;
WHERE (UPPER(THISFORM.kr_inv.Value) $ UPPER(a.i_name)) ;
      AND (a.h_id=b.h_id) AND (c.ref_nbr=b.ref_nbr);
NOCONSOLE ;
INTO TABLE inv_res
IF RECCOUNT(>0
  i=i+1
  resArr(i)="inv_res"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_erc.Value=1
SELECT d.std_type, d.ref_nbr, d.dt_code ;
FROM erc a, hospital b, std_hosp c, study d ;
WHERE (UPPER(THISFORM.kr_erc.Value) $ UPPER(a.e_name)) ;
      AND (a.h_id=b.h_id) AND (b.h_id=c.h_id) AND (d.ref_nbr=c.ref_nbr) ;
NOCONSOLE ;
INTO TABLE erc_res
IF RECCOUNT(>0
  i=i+1
  resArr(i)="erc_res"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_trade.Value=1
SELECT b.std_type, b.ref_nbr, b.dt_code ;
FROM vaccine a, study b ;
WHERE (UPPER(THISFORM.kr_trade.Value) $ UPPER(a.trdname)) ;
      AND (a.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE trade_re
IF RECCOUNT(>0
  i=i+1
  resArr(i)="trade_re"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_codenr.Value=1
SELECT b.std_type, b.ref_nbr, b.dt_code ;
FROM vaccine a, study b ;
WHERE (UPPER(THISFORM.kr_codenr.Value) $ UPPER(a.codenbr)) AND (a.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE codenr_r
IF RECCOUNT(>0
  i=i+1
  resArr(i)="codenr_r"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_bagnr.Value=1
SELECT b.std_type, b.ref_nbr, b.dt_code ;
FROM vaccine a, study b ;
WHERE (UPPER(THISFORM.kr_bagnr.Value) $ UPPER(a.bagnbr)) AND (a.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE bagnr_re
IF RECCOUNT(>0
  i=i+1
  resArr(i)="bagnr_re"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_lotnr.Value=1
SELECT b.std_type, b.ref_nbr, b.dt_code ;
FROM vaccine a, study b ;
WHERE (UPPER(THISFORM.kr_lotnr.Value) $ UPPER(a.lotnbr)) AND (a.ref_nbr=b.ref_nbr) ;
NOCONSOLE ;
INTO TABLE lotnr_re
```

```
IF RECCOUNT(>0
  i=i+1
  resArr(i)="lotnr_re"
ELSE
  give_up=.T.
ENDIF
ENDIF

IF !give_up AND THISFORM.chk_refnr.Value=1
SELECT std_type, ref_nbr, dt_code ;
FROM study ;
WHERE LIKE(IIF(EMPTY(THISFORM.kr_refnr1.Value), "*", UPPER(THISFORM.kr_refnr1.Value)) + ;
  IIF(EMPTY(THISFORM.kr_refnr2.Value), "*", THISFORM.kr_refnr2.Value), ;
  UPPER(std_type)+ref_nbr);
NOCONSOLE ;
INTO TABLE refnr_re
IF RECCOUNT(>0
  i=i+1
  resArr(i)="refnr_re"
ELSE
  give_up=.T.
ENDIF
ENDIF

maxi=i  &&Anzahl erzeugte Resultatstabellen

**Endresultat berechnen
IF give_up
DO FORM meldung WITH "Es wurden keine Studien gefunden, die Ihren Kriterien entsprechen."
ELSE
DO CASE
CASE maxi=1
  SELECT std_type, ref_nbr, dt_code ;
  FROM resArr(1) ;
  NOCONSOLE ;
  INTO TABLE master_r
  resArr(1)="master_r"
OTHERWISE
  FOR i=maxi TO 2 STEP -1
    SELECT DISTINCT * ;
    FROM resArr(i) a, resArr(i-1) b ;
    WHERE a.ref_nbr=b.ref_nbr ;
    NOCONSOLE ;
    INTO TABLE master_r
    DELETE FILE resArr(i-1)
    resArr(i-1)="master_r"
  ENDFOR
ENDCASE

**Resultat anzeigen
*Tabellen aus Speicher loeschen
FOR i=1 TO maxi
  SELECT resArr(i)
  use
ENDFOR
DO FORM showres WITH "master_r", THISFORM.search &&.search=.T.: nur suchen, =.F.: suchen und
editieren
ENDIF
```

6 Literaturverzeichnis

H. F. Korth, A. Silberschatz: Database System Concepts, McGRAW-HILL 1991
Prof. Dr. O. Nierstrasz: Unterlagen zur Vorlesung „Datenbanken“