Projektarbeit im Rahmen der Universität Bern

Bern, Mai 02
Nicolas Wrobel

Betreuung: Prof. O. Nierstrasz

Inhaltsverzeichnis

Projektarbeit im Rahmen der Universität Bern 1
Inhaltsverzeichnis 2
Einleitung 3
Allgemeines 3
Abgrenzung zur Dokumentation Danzeisens 3
Lebenszyklus dieser Applikation 4
Konzept 4
ISO 9001 für das ASTRA-Portfolio 5
Analyse 6
Design 7
Implementation 8
Tests 9
Systemumgebung (MS Access, VBA) 9
MS Access 9
Kurze Übersicht 9
Vorteile gegenüber anderen DBMS 10
Nachteile gegenüber anderen DBMS 10
Integration von VBA, Code-Beispiele 10
SQL, benutzte Abfragen 11
MS VBA 12
Kurze Übersicht 12
Vorteile gegenüber anderen Programmiersprachen 12
Nachteile gegenüber anderen Programmiersprachen 13
ASTRA–Portfolio Datenbank 13
Detailliertere Beschreibung der Tabelleneinträge 13
Das Mailtool 16
Beschreibung 16
Aufbau 16
Codeerklärung 17
Schlusswort 18

Einleitung

Allgemeines

Dieses Projekt fand im Rahmen des Informatikprojektes unter der Aufsicht von Prof. O. Nierstrasz im Sommer 2000 statt. Marc Danzeisen und ich haben die Möglichkeit bekommen für das ASTRA (Bundesamt für Strassen) ein Portfolio zu entwickeln.

Portfolio, *das; -s,-s (ökon.)* Portefeuille (1)
Portefeuille, *(), das; -s,-s 1. (ökon.)* (von einer Bank verwalteter) Wertpapierbestand 2. Geschäftsbereich eines Ministers 3. *(veraltet)* Brieftasche, Aktentasche

Die Aufgabe war ein Projektmanagement-Tool zu entwickeln, das von der Chefetage des ASTRAs benutzt werden sollte. Unser Gesprächspartner auf der ASTRA - Seite war Herr Thierry Berset. Mit Herrn Berset haben wir die genauen System- und Applikationsanforderungen bestimmt. Ihm haben wir auch die Prototypen und das fertige Produkt vorgestellt.

Abgrenzung zur Dokumentation Danzeisens

Da wir dieses Projekt zu Zweit erarbeitet haben, haben Marc Danzeisen und ich auch zwei verschiedene Dokumentationen geschrieben. Damit das Projekt umfassend und nicht doppelt dokumentiert wird, haben wir unsere Dokumentation folgendermassen aufgeteilt:

Danzeisen's Arbeit[1] (konzeptuell):

- ¤ Beschreibung der Konzept-Phase
    - o Analyse des Ist-Zustandes
    - o ASTRA - bedingte Voraussetzungen
    - o Anforderungen an die Portfolio – Applikation
- ¤ Beschreibung der Design-Phase
    - o Grobe / Detaillierte Funktionsbeschreibung der Applikation
- ¤ Handbuch

Diese Arbeit (technisch):

- ¤ Beschreibung des Softwareprozesses als solches
- ¤ Detaillierte Informationen bez. der Systemumgebung
- ¤ Detaillierte Informationen bez. der Implementation

---

[1] Marc Danzeisen, **"ASTRA — Portfolio,"** Technical Report, University of Bern, June 2000, Informatikprojekt. Befindet sich unter http://www.iam.unibe.ch/~scg/Archive/Projects/danzeisen.pdf

Lebenszyklus dieser Applikation

In Diesem Kapitel werde ich kurz den Lebenszyklus unserer Applikation "Portfolio" erläutern. Nach der altbekannten Wasserfall-Anschauung.
Die Idee des Programms hatte Herr Berset, der, im Bundesamt für Strassen, die Administration der laufenden und bevorstehenden Projekte vereinfachen wollte. MS Projekt war für diesen Zweck zwar passend aber zu kompliziert. Die Applikation sollte für jedermann auf Anhieb verständlich und benutzbar sein.

Konzept

So entstand also die Idee eine Applikation zu entwickeln die folgenden Bedürfnissen gerecht werden sollte:

- benutzerfreundlich
- muss, ohne grossen Aufwand, auf den Arbeitsstationen des ASTRA laufen
- zentral verwaltbar
- muss ISO 9001 – Standard erfüllen
- soll alle bisherigen Arbeitsschritte umfassen

Benutzerfreundlich heisst, das Mitarbeiter des ASTRA ohne grossen schulischen Aufwand die neue Software benutzen können. Das Programm sollte also ohne weitere Kosten eingeführt werden können. Ein weiterer Punkt ist der das je einfacher und spezifischer die Angaben zum Portfolio sind um so überschaubarer und sicherer wird die Applikation. Sicher im Sinne von Fehlerfreiheit und Robustheit.

Die Systemumgebung des ASTRA ist Microsoft-lastig. Die Serverumgebung besteht hauptsächlich aus NT-Servern (Exchange-, SQL- und verschiedene File-Server), die Workstations der Mitarbeiter sind mit NT-Workstation 4.0 eingerichtet. Es boten sich 2 Alternativen an, die Applikation ohne grossen Aufwand auf den Arbeitsstationen des ASTRA laufen zu lassen; zum ersten eine Intranet-Lösung mit ASP und einer MS-Access-Datenbank, zum zweiten eine reine MS-Access-Datenbank. Andere Datenbankmanagementsysteme hätten wir neu im ASTRA einführen müssen und eine Client/Server-Applikation sahen wir als zu aufwendig für diesen Zweck. Da wir noch keine Erfahrung mit ASP hatten und in dieser Lösungsvariante ebenfalls eine Access-Datenbank im Hintergrund hätten aufbauen müssen, entschieden wir uns die gesamte Applikation in Access zu programmieren.

"Zentral Verwaltbar" war für Herrn Berset ein wichtiger Punkt. Damit der administrative Aufwand des Programms so einfach wie möglich wird. So entstand die Idee das wir 2 verschiedene Benutzergruppen einführten, Administratoren und normale Benutzer. Die Administratoren haben jegliche Zugriffsberechtigungen auf die Datenbank, darunter fällt Projekteverwaltung (erfassen, mutieren, Änderungsanträge bearbeiten) und Benutzerverwaltung(erfassen, löschen, Passwörter ändern). Die normalen Benutzer haben nur das Recht Projekte zu erfassen, ihre erfassen Projekte zu mutieren (Änderungsanträge stellen) und ihr Login-Passwort zu ändern.

Unter den ISO 9001 – Standard[2] soll man sich folgendes vorstellen. Dies in ein Standard der von der ISO (International Organization for Standardization) erstellt

[2] http://www.iso.ch/

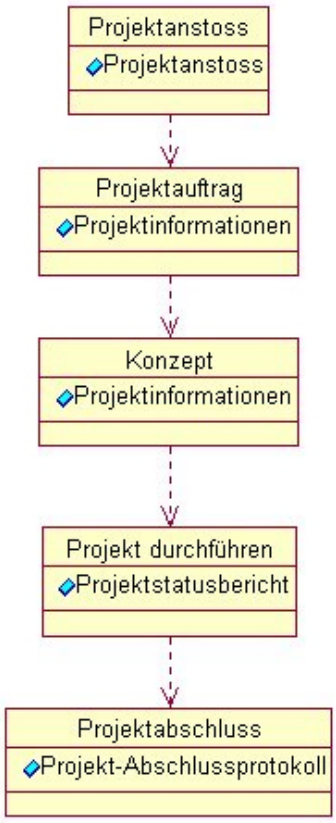
wurde um die Qualität des Produktes oder Dienstes (in unseren Fall, die Qualität der zu verwaltenden Projekte) sicherzustellen. Nun hat jeder eine andere Vorstellung was Qualität bedeutet, deswegen hat sich dieser Qualitätsstandard darauf spezialisiert den Bedürfnissen des Kunden gerecht zu werden. ISO 9001 garantiert eine standardisierte Vorgehensweise der jeweiligen Projekte. Die folgenden Punkte sind aus dem Projektleitfaden des ASTRAS:

- Die Projektvorgaben sind dokumentiert (Ziele, Termine, Kosten und Verantwortlichkeiten)
- Die Projektphasen sind dokumentiert
- Nach jeder Phase wird ein Projektreview durchgeführt, um die Fortsetzung freizugeben. Diese Tätigkeiten werden in einem Bericht dokumentiert
- Die Projektergebnisse werden geprüft, angenommen und dokumentiert (Schlussbericht)
- Der Projektleiter führt eine Abweichungsliste und behandelt sie gemäss seinen Kompetenzen
- Die Projektrückverfolgbarkeit ist gewährleistet

ISO 9001 für das ASTRA-Portfolio

1. Diese Angaben muss man bei der Erfassung neuer Projekte angeben. Änderungen können anschliessend nur von Administratoren vorgenommen werden, damit ist die Rückverfolgung der Projekte gewährleistet. Denn es werden alle Projektänderungen protokolliert. Projektänderungen umfassen die Punkte Kosten, Termine, Ressourcen und Ziele.
2. Projektphasen können bei der Erfassung neuer Projekte angegeben werden. Oder im Verlaufe eines Projektes adaptiert, resp. neu erfasst werden.
3. Es gibt die Möglichkeit sich Berichte zu den einzelnen Projekten ausdrucken zu lassen. So ist es dem Projektverantwortlichen möglich weitere Entscheide bezüglich dieses Projektes zu fällen
4. Wenn ein Projekt fertiggestellt wurde kann man Dieses archivieren. Es ist möglich Bemerkungen zu den einzelnen Projekten zu erfassen, um die Ergebnisse zu dokumentieren oder einen Schlussbericht zu schreiben.
5. Die Abweichungsliste wird implizit geführt durch die Änderungsanträge die im Laufe des Projektes entstehen. So wird festgehalten welche Punkte (Kosten, Termine, Ressourcen oder Ziele) im Verlauf des Projektes von den Vorgaben abweichen.
6. Die Projektrückverfolgbarkeit ist so geregelt, das alle Änderungsanträge in einer separaten Tabelle abgelegt werden. In der Detailansicht des Projektes werden nur die Anzahl der jeweiligen Änderungen angezeigt. Die Änderungsanträge können nur von den Administratoren verwaltet werden.

<u>"Soll alle bisherigen Arbeitsschritte umfassen"</u> soll heissen das alle Arbeitsschritte, die im Moment für die Projektleitung von Nöten sind, auch verarbeitet werden sollen. In Bersets Projektleitfaden sind verschiedene Schritte für die Ausführung eines korrektes Projektes vorgesehen, mit entsprechenden Formularen. Unsere Applikation soll diese Formulare zum Teil ersetzen.



Analyse

Die bisherigen Arbeitsschritte waren Teil unserer Ist-Analyse. Hier war uns Herr Berset eine grosse Hilfe, denn er hat für das ASTRA bereits ein Dokument("ASTRA – Projektleitfaden") verfasst das die Standardisierung der Projektabläufe gewährleisten soll.
Dieses Dokument diente uns als Vorlage für unser Projekt, das Portfolio. Die Arbeitsschritte Projektanstoss, Projektauftrag, Konzept, Projekt durchführen und Projektabschluss werden hier ausführlich dokumentiert. So haben wir ein klares Bild bekommen wie der Ablauf bisher in der ASTRA geregelt war.

Die Anforderungen an das Portfolio waren nicht von Anfang an klar definiert. In der Phase des Prototyping kamen immer neue Wünsche und Anforderungen an unser Projekt.
So kamen vor allem noch zwei weitere wichtige Punkte hinzu:

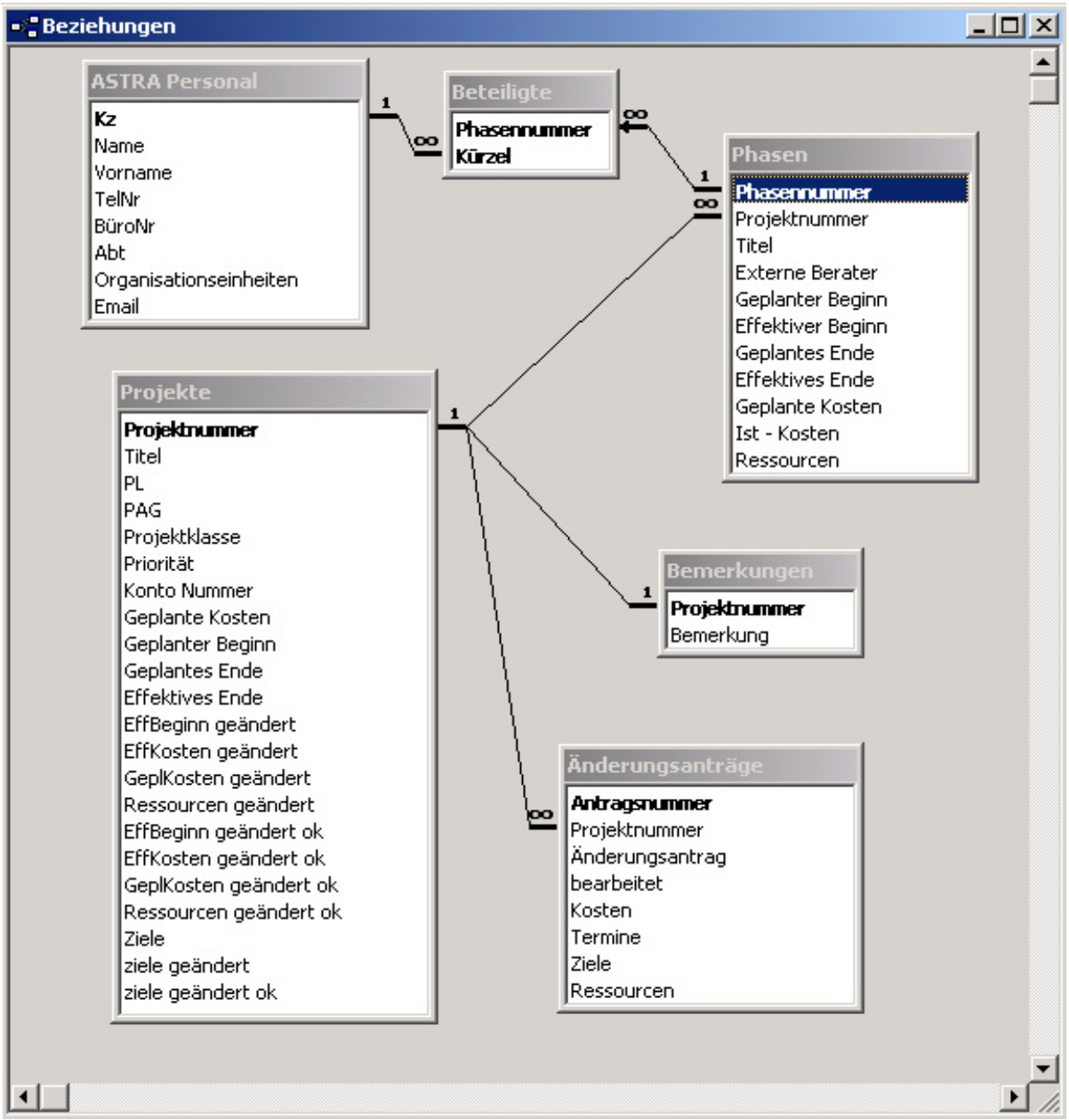
1. Automatische Email-Benachrichtigung bei Projekten die nicht Plangemäss verlaufen. Für die Implementation werde ich genauer im Kapitel "Das Mailtool" eingehen
2. Archivierung der abgeschlossenen Projekte

1. Das Problem der autom. Email-Benachrichtigung sind wir gleich angegangen wie das Portfolio. Wir konnten diese Funktionalität nicht in der MS-Datenbank implementieren, da diese Benachrichtigung in einem Bestimmten Intervall automatisch erfolgen sollte. Und nicht das ein Administrator dafür verantwortlich gemacht werden muss, das die Benachrichtigungen ausgelöst werden. So haben wir uns entschieden ein eingenständiges Programm zu entwickeln das die Datenbank

überprüft und die entsprechenden Mails versendet. Das Programm wird dann vom NT-Server direkt angestossen über den 'at'-Befehl.
2. Die Archivierung sollte sich als nicht all zu grosses Problem herausstellen. Wir haben eine zweite Datenbank erfasst mit den gleichen Tabellen wie im Portfolio. Weiter haben wir einen Mechanismus zur Verfügung gestellt, der es ermöglicht abgeschlossene Projekte aus der Portfolio-Datenbank in die neue Archiv-Datenbank zu verschieben.

Design

Angefangen hat die Design-Phase mit dem finden eines geeigneten Datenmodells. Wir hatten den "Projektleitfaden" und die Anforderungen Bersets, die uns in die richtige Richtung brachten. Wir haben ein ER-Diagramm auf Papier aufgestellt und anschliessend dieses 1:1 in MS-Access implementiert. Das Tabellen-Modell sieht nun so aus.



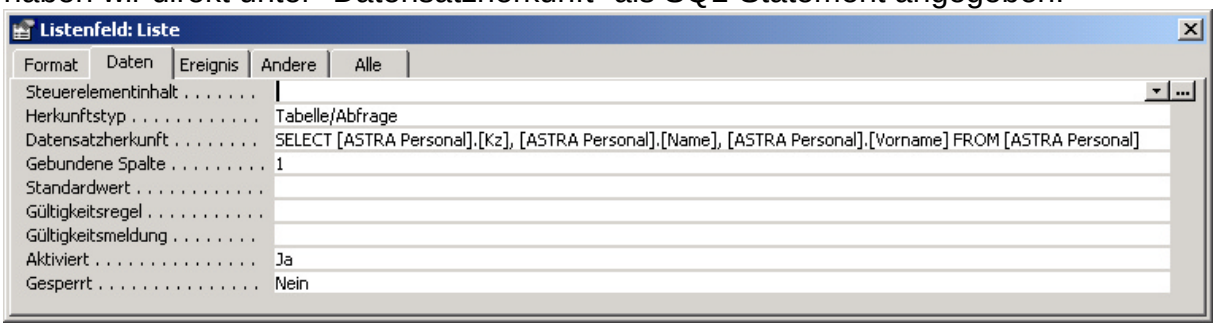
Zu den einzelnen Tabelleneinträgen gehe im Kapitel "Detailliertere Beschreibung der Tabelleneinträge" ein. Hier nur eine kurze Erläuterung.

Zentral ist die Tabelle Projekte. Zu jedem Projekt werden die Phasen in der entsprechenden Tabelle festgehalten. Zu jeder Phase können mehrere Beteiligte des ASTRA-Personals erfasst werden. Weiter hat jedes Projekt mehrere Änderungsanträge, die den Verlauf des Projektes dokumentieren.
Die Tabelle "Bemerkungen" haben wir separat erfasst damit wir im Detailbericht eines Projektes die Bemerkung dieses Projektes als Unterformular einfügen konnten, das wäre sonst als normales Feld der Tabelle "Projekte" nicht möglich.

Von MS Access aus sind die weiteren Schritte bezüglich des Designs schon vorgegeben. Dieses DBMS ist in 6 verschiedenen Teilen aufgebaut.

1. Tabellen
2. Abfragen
3. Formulare
4. Berichte
5. Makros
6. Module

1. Die Tabellen sind oben bereits erwähnt.
2. Wir haben 4 Abfragen erfasst, die über mehrere Tabellen gingen. Einfache Abfragen, wie zum Beispiel die Auflistung aller möglichen Beteiligten einer Phase, haben wir direkt unter "Datensatzherkunft" als SQL-Statement angegeben.



3. Weiter haben wir 21 Formulare verfasst die im Handbuch jeweils erklärt sind.
4. 2 Berichte stehen zur Verfügung. Eines für einen Detailausdruck eines Projektes. Das Zweite für den Ausdruck des Suchergebnisses.
5. Das einzige Macro das wir geschrieben haben dient zur automatischen Ausführung des Login-Dialogs. So wird es verunmöglicht die Datenbank zu manipulieren ohne sich einzuloggen.
6. Unser Design sieht so aus, das wir 3 VBA-Module geschrieben haben, dessen Funktionalität wir in den jeweiligen Formularen benutzen. So war es zum Beispiel möglich Eingabeüberprüfungen für ähnliche Formulare wiederzuverwenden.

Implementation

Anfangs war die Arbeitsaufteilung zwischen Marc und mir noch einfach. Konnten wir doch einfach die vorgesehenen Formulare unter uns aufteilen und diese anschliessend in ein Projekt zusammenführen. Um dieses dann Herrn Berset vorzustellen, so dass wir wieder Feedback bekamen.
Später, als das Grobgerüst mit den Formularen stand, fingen wir an Formular für Formular mit der jeweiligen Funktionalität zu versehen. Auch diese Aufgabe konnten wir noch gut aufteilen. So hatten wir anfänglich für jedes Formular ein eigenes Modul, in den die jeweilige Funktionalität stand.

Je länger wir jedoch am Portfolio arbeiteten umso gravierender wurden die Abhängigkeiten zwischen unseren Teilen. So war es kaum mehr möglich das wir beide separat programmierten, vielmehr musste einer von uns immer auf den anderen warten, bis dieser sein neues Teilmodul fertiggestellt hatte.
Die Implementation des Mail-Tools konnte ich unabhängig von Portfolio durchführen. Denn dazu benötigte ich ausschliesslich die Definition der Tabellen. Die Tabelle „Projekte“ musste ich noch erweitern mit den 8 Flags „xxx geändert [ok]“[3] damit die richtigen Mails generiert werden können.

Tests

Dieses Kapitel wird eher kurz ausfallen, denn wir haben unsere Teilmodule und die Funktionalität der gesamten Applikation manuell ausgetestet. Wir haben eine Hand voll Testdaten eingegeben und anschliessend mit diesen die neue Funktionalität ausgetestet.
Da die Applikation unser Erachtens nicht so gross und komplex ist, das wir ausgiebigere Tests hätten implementieren sollten, entschlossen wir uns für die einfachste Variante. Dies erwies sich auch als die Richtige.

Systemumgebung (MS Access, VBA)

MS Access

Kurze Übersicht

MS Access ist ein Teil des Microsoft Office Pakets. Aus diesem Grund ist es relativ leicht möglich Daten aus der Portfolio-DB in andere Office-Applikationen zu übertragen. Z. B. WinWord oder Excel.
Eine Access-Datenbank kann sechs verschiedene Typen von Komponenten (Objekten) enthalten: Tabellen, Abfragen, Formulare, Berichte, Makros und Module.
Tabelle: Organisation der Daten.
Abfrage: Mit einer Abfrage können Datensätze einer oder mehrerer Tabellen angezeigt werden, die bestimmte Kriterien erfüllen. Zu erwähnen ist hier die hilfreiche visuelle Erstellungsmöglichkeit, diese erzeugt eine SQL-Abfrage die man ohne weiteres in den Modulen oder in Datensatzherkunftsfeldern in Formularen benutzen kann.
Formular: Ein Formular dient zur Ansicht, Eingabe und Bearbeitung von Daten aus Tabellen (und Abfragen).
Bericht: Die Ausgabe von Daten aus Tabellen oder Abfragen auf Drucker geschieht meist als Bericht. Der Aufbau eines Berichtes (Layout) muss zuvor definiert werden.
Makro: Immer wiederkehrende Datenbankarbeiten können mit Makros automatisiert werden. Hier wird eine Folge von Tastatur-Befehlen oder auch von Mausklicken gespeichert und kann dann auf andere Datenbanken oder Tabellen mit ähnlicher Struktur angewandt werden.

[3] siehe Tabellendefinition

Modul: Lösungen für spezielle Erfordernisse können mit einer eigenen Programmiersprache (hier Visual Basic) als sog. Modul erstellt werden.
Access verwaltet alle Objekte dieser sechs Objekttypen in einer einzigen Datei (Microsoft-Database-Datei, Dateinamenerweiterung MDB) mit einer spezifischen Datenstruktur.

Vorteile gegenüber anderen DBMS

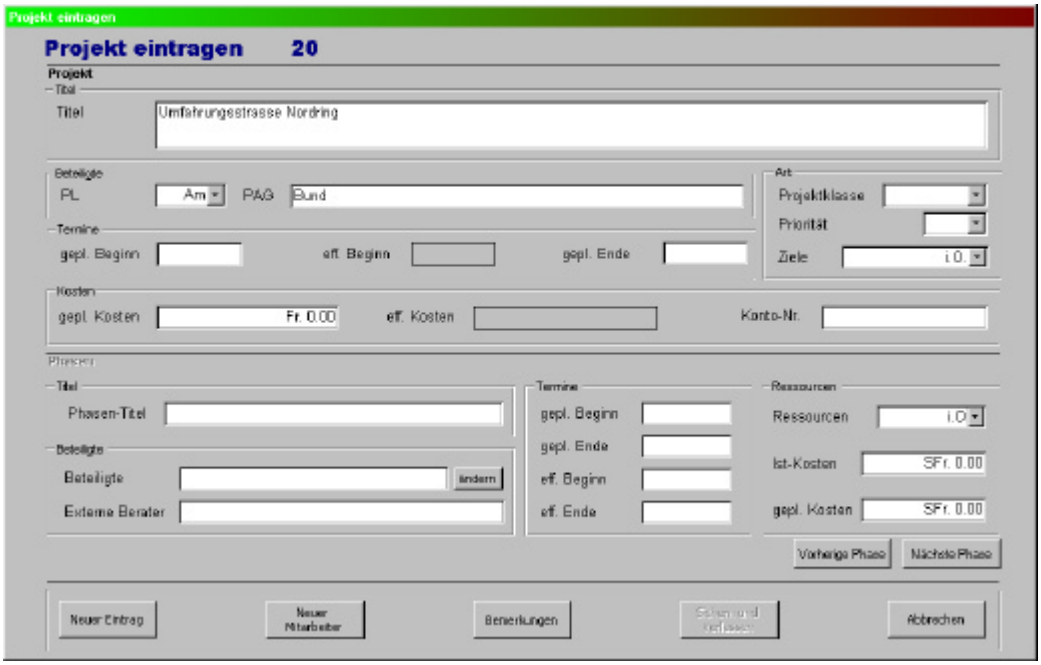
Die integrierte Programmiersprache VBA für Access ist ein sehr praktisches Tool um nicht-triviale Probleme in einer DB-Anwendung zu lösen.
Das Tool Access ist sehr intuitiv und einfach bedienbar. Sämtliche Eingaben werden durch Wizards und einfach Formulare durchgenommen. Dies kann sich aber auch als Nachteil herausstellen, da man so in gewissen Bereichen limitierter ist.
Access ist SQL-fähig. SQL-Statements können sehr einfach mit dem Abfrage-Assistenten erstellt werden. Der so generierte SQL-Text kann noch entsprechend angepasst werden.
Es ist möglich Access als reinen Client zu verwenden. Und die Datenbank und DB-Anfragen auf einen SQL-Server auszulagern, so kann man den Netzverkehr reduzieren. Access generiert die Anfrage, diese wird an den Server geschickt. Der Server verarbeitet diese Anfrage und schickt die Daten der Antwort an Access zurück.

Nachteile gegenüber anderen DBMS

Oracle oder Sybase Datenbanksysteme sind konzipiert für grosse Datenmengen und entsprechend grosse Anzahl an Anfragen auf die Tabellen. Access als DBMS ist für den privaten oder Smallbuissness-Bereich sehr gut einsetzbar. Übersteigt aber die DB die Grenzen Access' ist es ratsam sich einen MS SQL-Server zu beschaffen, der dieser Aufgabe gewachsen ist.

Integration von VBA, Code-Beispiele

Ein kleines Beispiel des VBA-Codes für folgendes Formular



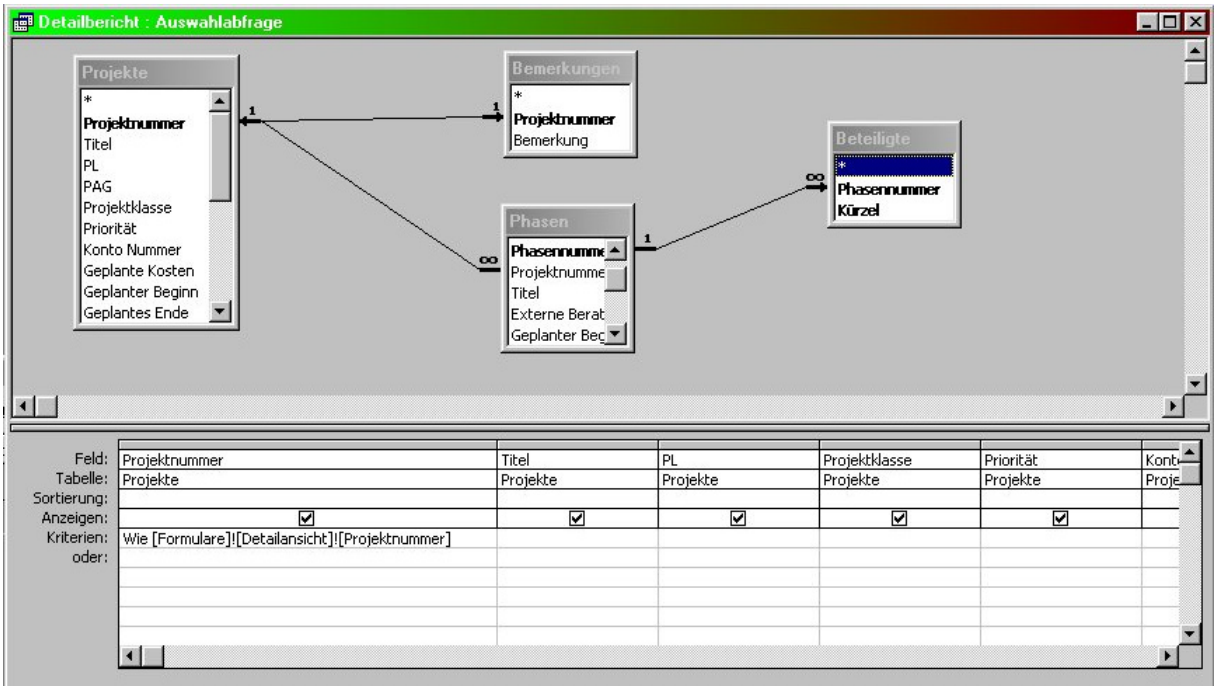
Beim Verlassen des Formularfeldes Beteiligte✍ PL muss überprüft werden ob der aktuelle Benutzer dazu berechtigt ist. Die gleiche Routine wird auch im Projekt aktualisieren aufgerufen.
Die Routine "testInput" kontrolliert ob die restlichen Felder richtig gesetzt sind.

VBA-Code:

```
Private Sub PL_Exit(Cancel As Integer)
  If (data.right = "u" And Me.PL <> data.kürzel) Then
        MsgBox "Sie sind nicht berechtig eine andere Person als
Projektleiter zu bestimmen!" & Chr(10) & Chr(13) & _
               "Bitte setzen Sie Ihr eigenes Kürzel ein.", , "Keine
Berechtigung"
        Me.PL.SetFocus
        Exit Sub
    Else
        testInput
    End If
End Sub
```

SQL, benutzte Abfragen

Ein kleines Beispiel einer von Access generierten Abfrage. Hier geht es darum alle Informationen eines Projektes aus der Datenbank herauszulesen. Da diese Informationen aber in verschiedenen Tabellen gespeichert sind, ist es mit Hilfe der Auswahlabfrage von Access einfacher als selbst das passende SQL-Statement zu verfassen.



Man gibt an welche Tabellen beteiligt sind. Hierbei darf man natürlich die Beziehungstabellen nicht vergessen, nur kann man sicherstellen, dass auch nur korrekte Tupel herausgelesen werden.
Hat man die Tabellen ausgewählt entscheidet man in der unteren Hälfte des Fensters welche Felder man schlussendlich benötigt.
In dem oben angeführten Bild kann man erkennen das es auch möglich ist Kriterien anzugeben, in diesem Fall geht es darum das nur ein spezielles Projekt aus der DB gelesen werden soll. Nämlich das Projekt das im Moment auch dargestellt wird.

MS VBA

Kurze Übersicht

Bei VBA handelt es sich um eine adaptierte Version der Programmiersprache Visual Basic. VBA (Visual Basic for Applications) kommt für jedes Office-Produkt in einer leicht variirten Version vor. So gibt es ein VBA für Excel ein VBA für Access und so weiter. Es ist nicht möglich ein eigenständiges Programm ausschliesslich in VBA zu schreiben. Hierfür muss man das ursprüngliche Visual Basic verwenden.
VBA nutzt den Vorteil jeder Applikation speziell, so ist es einfach, bereits vorhandene Funktionalität zu benutzen. In Excel, zum Beispiel, die mathematischen Funktionen und tabellarischen Berechnungen. In Access sämtliche datenbankspezifischen Funktionen, wie das Ein- und Auslesen der Daten, SQL-Abfragen usw.

Vorteile gegenüber anderen Programmiersprachen

Den Hauptvorteil habe ich bereits oben erwähnt. Man hat in VBA für Access die Möglichkeiten sämtliche Funktionalität von Access zu benutzen.
Interessant für uns war vor allen die Einfachheit Formulare zu gestalten die wir in der Anfangsphase Herrn Berset vorgestellt haben. Praktisch alle Formularfelder können

sehr einfach und schnell mit den entsprechenden Datenbanktabellen oder Abfragen verknüpft werden.
Mit anderen Programmiersprachen muss man sich noch weitere Gedanken zum Programmaufbau machen, was in erster Linie nichts zum Lösen der Aufgabe "Portfolio" beiträgt. In VBA kann man sich konkret auf das zu lösende Problem konzentrieren, die Datenbank und die Manipulation der Daten.

Nachteile gegenüber anderen Programmiersprachen

Der grosse Nachteil der Programmiersprache VBA ist, das es keine Möglichkeit gibt ein "Standanlone"-Programm zu entwickeln. Es ist immer nötig auch das entsprechende Office-Produkt installiert zu haben. Darüber hinaus ist VBA auch relativ limitiert. Hat man spezielle Wünsche das Ausehen eines Formulars zu ändern oder zeitkritische Abfragen zu optimieren, kommt man nicht darum herum eine Lösung mit VB, resp. VC++ zu finden und diese dann als DLL in VBA zu nutzen.

ASTRA–Portfolio Datenbank

Detailliertere Beschreibung der Tabelleneinträge

Es folgt die detaillierte Beschreibung unseres Datenbankschemas:
Projekte:

Feldname	Felddatentyp	Beschreibung
Projektnummer	Autowert	Der "primary key" dieser Tabelle. Autowert ist ein Datentyp aus Access, der eine einedeutige, forlaufende Nummer für jeden neuen Eintrag generiert.
Titel	Text	Der Titel dieses Projektes
PL	Text	Projektleiter. Der Verantwortliche dieses Projektes
PAG	Text	Projektauftraggeber. Der Sachverständige und Ansprechperson auf Seiten des Kunden
Projektklasse	Text	Enumeration (gross, mittel, klein)
Priorität	Text	Enumeration (1 bis 6)
Konto Nummer	Text	Konto das für die Finanzierung dieses Projektes benötigt wird.
Gelplante Kosten	Währung	Anfänglich geplantes Budget. Kann nach der Erfassung des Projektes nicht mehr geändert werden. Nur über Änderungsanträge ist es möglich weitere finanziellen Mittel locker zu machen.
Geplanter Beginn	Datum	Anfänglich geplater Beginn der Realisierung. Auch dieser Wert ist nach der Erfassung des Projektes nich mehr veränderbar.
Geplantes Ende	Datum	Geplantes Ende dieses Projektes
Effektives Ende	Datum	Effektives Ende dieses Projektes. Es muss beim Erfassen hinter dem als "spätest erfassten" effektiven Ende jeder Phase, dieses Projektes, liegen.

Ziele	Text	Enumeration ("i.O.";"mit Schwierigkeiten";"nicht mehr möglich")
EffBeginn geändert	Bool	All diese Bool-Werte dienen lediglich dazu, das das Mailtool bestimmen kann zu welchem Projekt die entsprechenden Warn-Emails geschickt werden müssen.
EffBeginn geändert ok	Bool	
EffKosten geändert	Bool	
EffKosten geändert ok	Bool	
GeplKosten geändert	Bool	
GeplKosten geändert ok	Bool	
Ressourcen geändert	Bool	
Ressourcen geändert ok	Bool	
Ziele geändert	Bool	
Ziele geändert ok	Bool	

Änderungsanträge:

Feldname	Felddatentyp	Beschreibung
Antragsnummer	Autowert	Primay Key dieser Tabelle.
Projektnummer	Zahl	Foreign Key aus der Tabell "Projekte". Der Autowert ist vom Typ Zahl.
Änderungsantrag	Memo	Textfeld in dem der Projektleiter seine Änderungswünsche äussern kann. Dient gleichzeitig als Dokumentation des Projektes. Der Typ Memo entspricht dem Typ Text, mit dem Unterschied das bis zu 64000 Zeichen anstelle der 255 benutzt werden können
bearbeitet	Bool	Dieses Flag gibt Auskunft darüber ob ein Administrator diesen Änderungsantrag bereits bearbeitet hat oder nicht. In der Auflistung aller Änderungsanträge können die Bearbeiteten ausgeblendet werden.
Kosten	Bool	Diese Bool-Werte sagen aus um welche Art Änderung es geht.
Termine	Bool	
Ziele	Bool	
Resourcen	Bool	

Bemerkungen:

Feldname	Felddatentyp	Beschreibung
Projektnummer	Zahl	Primary Key
Bemerkung	Memo	Es wäre besser dieses Feld einfach in die Tabelle Projekte aufzunehmen. Da wir dieses Feld aber in einem Unterformular im Detailbericht eines Projektes anzeigen wollen, mussten wir uns dieses Tricks

		behelfen.

Phasen:

Feldname	Felddatentyp	Beschreibung
Phasennummer	Autowert	Primary Key.
Projektnummer	Zahl	Foreign Key auf die Tabelle Projekte.
Titel	Text	Titel dieser Phase
Externe Berater	Text	Einfacher Text um mitbeteiligte Personen, Firmen zu erfassen, die nicht in der Tabelle "ASTRA Personal" erfasst sind.
Geplanter Beginn	Datum	Geplanter Beginn dieser Phase. Darf nicht vor dem geplanten Beginn des Projektes liegen.
Effektiver Beginn	Datum	Datum an dem diese Phase in Realität begonnen hat.
Geplantes Ende	Datum	Geplantes Ende dieser Phase. Darf nicht hinter dem geplanten Ende des Projektes liegen.
Effektives Ende	Datum	Datum an dem diese Phase in Realität endet.
Geplante Kosten	Währung	Geplante Kosten dieser Phase. Diese können nach erfassen der Phase nicht mehr geändert werden.
Ist - Kosten	Währung	Die aktuellen Kosten dieser Phase.
Ressourcen	Text	Enumeration ("i.O.";"mit Schwierigkeiten";"nicht mehr möglich"). Im Detailbericht des Projektes wird unter Ressourcen des Projektes das Minimum der Ressourcen aller Phasen angezeigt.

Beteiligte:

Feldname	Felddatentyp	Beschreibung
Phasennummer	Zahl	Foreign Key auf die Tabelle Phasen
Kürzel	Text	Foreign Key auf die Tabelle ASTRA Personal

ASTRA Personal:

Feldname	Felddatentyp	Beschreibung
Kz	Text	Kürzel. Gleichzeitig auch der Primary Key dieser Tabelle. mit
Name	Text	Persönliche Angaben dieser Person. Die Email-Adresse wird vom Mailtool benötigt. Falls es sich bei dieser Person um einen Projektleiter handelt, dessen Projekt in einem der 4 Hauptpunkte ungenügend ist, das entsprechden Mail an diese Mailadresse geschickt.
Vorname	Text	
TelNr	Text	
BüroNr	Text	
Abt	Text	
Organisationseinheiten	Text	
Email	Text	

Zugriff:

Feldname	Felddatentyp	Beschreibung
Kürzel	Text	Foreign Key auf die Tabelle ASTRA Personal.
Passwort	Text	Das im Klartext abgespeicherte Passwort dieses Mitarbeiters. Wird beim Einloggen (Starten) des Programms verlangt.
Berechtigung	Text	Enumeration (u, e). Es gibt 2 verschiedene

		Benutzertypen, den Normalbenutzer (u: user) und den Administrator (a: admin)

In der Archiv Datenbank befinden sich, bis auf die Zugriffstabelle, genau die gleichen Tabellen. So ist auch einfach möglich die "alten" Daten aus der Portfolio-DB in das Archiv überzuführen.

Das Mailtool

Beschreibung

Das Mailtool ist das Programm, das wir erst in der zweiten Hälfte der Projektarbeit in Angriff nahmen. Die Idee, das Projektleiter automatisch auf Missstände bezüglich Ihrer Projekte hingewiesen werden, kam Herrn Berset erst später. Damit ein Projekt als gefährdet eingestuft wird muss sich mindestens eines der vier Kriterien (Ziele, Kosten, Ressourcen oder Termine) im roten, kritischen Bereich befinden. Die Berechnungsformeln haben wir von Herrn Berset bekommen und können in der Dokumentation Danzeisens eingesehen werden.
Ich habe Visual Basic verwendet um das Mailtool zu implementieren. Da wir uns ohnehin schon in VBA eingearbeitet haben lag das Nahe. Das Mailtool hat 2 Funktionsweisen einmal als grafisches Tool

und einmal als Konsolenapplikation. Hier übergibt man den Pfad der Datenbank als Parameter an das Programm, so wird auch das oben abgebildete Fenster unterdrückt. Die Option das Programm auch als Konsolenapplikation starten zu können war nötig, da im Normalbetrieb dieses Programm periodisch alle Woche einmal ausgeführt werden soll. Es war Vorgesehen das Programm auf dem gleichen Server, wie die Portfolio-Datenbank zu installieren. Und anschliessend mit dem 'AT'-Befehl von MS Windows NT den Task wöchentlich zu starten.

Aufbau

Um diese Funktionalität auch gewähleisten zu können, musste unser ursprüngliches Datenmodell noch um 8 weitere Flags in der Tabelle Projekte erweitert werden.

EffBeginn geändert
EffBeginn geändert ok
EffKosten geändert
EffKosten geändert ok
...
"EffBeginn geändert" wird im Portfolio gesetzt sobald die Daten geändert wurden. Das heisst für das Mailtool es muss auf jeden Fall ein Mail verschicken (sofern sich dann die aktuellen Kosten im roten Bereich befinden).
"EffBeginn geändert ok" wird vom Mailtool gesetzt damit dem gleichen Projektleiter nicht jede Woche das Gleiche Mail erreicht. Projektänderungen brauchen ja immer etwas Zeit.

Codeerklärung

Nun noch ein Codebeispiel aus dem Mailtool. Diese Funktion wird für ein spezielles Projekt aufgerufen. Als Parameter wird ein Recordset von Phasen (alle Phasen des Projekes) übergeben und das dazugehörige Projekt. Aus diesen Phasen wird die schlechteste Ressourcen-Verfügbarkeit bestimmt. Handelt es sich um ein ungenügendes Ergebnis wird der entsprechdende Fehlerstring zurückgegeben.

```
Private Function checkRessourcen(pr As Recordset, phasen As Recordset) As
String
    Dim wert As Integer

    'check ob mail schon verschickt..
    If (pr.Fields("ressourcen geändert ok") = True) Then
        checkRessourcen = ""
        Exit Function
    End If

    wert = 3
    phasen.MoveFirst
    On Error GoTo ende
    While Not phasen.EOF
        If (rsc2int(phasen.Fields("Ressourcen")) < wert) Then wert =
rsc2int(phasen.Fields("Ressourcen"))
ende:
        phasen.MoveNext
    Wend

    If (wert = 1) Then
        checkRessourcen = "Die Ressourcen Ihres Projektes sind ungenügend."
& Chr(10)
        'flag setzen
        pr.Edit
        pr![ressourcen geändert ok] = True
        pr.Update
    Else
        checkRessourcen = ""
    End If
End Function

Private Function rsc2int(str As String) As Integer
    rsc2int = Switch(str = "i.O", 3, str = "knapp", 2, str = "ungenügend",
1)
End Function
```

Schlusswort

Dieses Software-Projekt war nach meiner PSE-Erfahrung als Schüler und Assistent nicht das Erste. Trotzdem war es neu für mich direkt mit dem Kunden zu reden und seine Ansichten und Wünsche umzusetzen. Den Auftrag verdanken wir Marc, da er Beziehungen zum ASTRA hat. Wir haben versucht bei den ersten Treffen mit Herrn Berset ein möglichst klares Bild von seinen Erwartungen zu bekommen. Damit wir ungefähr den zeitlichen Aufwand abschätzen konnten. Und damit wir ein Pflichtenheft aufsetzen konnten mit dem Herr Berset auf der einen Seite und wir, als Programmierer auf der anderen Seite, einverstanden sein konnten.

Dies war bereits eine Aufgabe, die wir in keiner Vorlesung hätten üben können. Das Aufsetzen des Pflichtenheftes, wie viel Lohn kann man für so eine Arbeit verlangen oder was ist eigentlich möglich in 300-400 Arbeitsstunden, mit 2 Arbeitskräften? Uns war schon von Anfang an klar, das wir die Aufgabe mit MS Access und Visual Basic lösen wollen. Also war unser nächster Schritt uns mit dieser Umgebung vertraut zu machen.

Wir haben ja bereits, während der ersten Gespräche mit Herrn Berset, Prototypen der Oberfläche des Programms bereitgestellt. Diese Einfache Aufgabe ermöglichte uns gleichzeitig einen ersten Einblick in MS Access. Ausserdem war dieser Prototyp Bestandteil unseres Pflichtenheftes.

Nach 5-6 Besuchen im ASTRA bei Herrn Berset hatten wir den Auftrag einigermassen klar umrissen. Wir haben nun mit der Realisierung begonnen, wie ich in den vorangehenden Kapiteln beschrieben habe. Nachdem wir mit der Implementierung begonnen haben, hatten wir keine weiteren Treffen mit Herrn Berset vereinbart. Wir konnten Ihm nun keine Fortschritte mehr zeigen, wie zu beginn mit den Prototypen. Für den Benutzer sah das Programm immer noch wie der Prototyp aus, nur das nach und nach immer mehr Formulare mit der DB verbunden wurden. Und für das iterative Testen wollten wir Herrn Berset nicht einbeziehen, da wir auf dem Standpunkt standen das die ASTRA für die funktionierende Software bezahlt und nicht dafür das Herr Berset Zeit investiert um die Software zu debuggen. Nun, im Nachhinein, denke ich, wahrscheinlich ist es gar nicht schlecht auch Zwischenversionen dem Kunden zukommen zu lassen. Damit Sie im Bilde sind was funktioniert bereits; funktioniert es so wie gedacht usw. hätten wir z. B. unser DB-Schema anpassen müssen, wären wir niemals in der Zeit fertig geworden.

Erst nachdem wir den grössten Teil implementiert hatten, trafen wir uns wieder. Da wir unser Pflichtenheft detailliert genug verfasst hatten, entsprach das Programm auch genau diesen Ansprüchen. Trotzdem hatten wir nach unserem Treffen neue Anforderungen, bezüglich der Archivierung und der Emailbenachrichtigung in der Tasche. Wir haben das Pflichtenheft entsprechend angepasst und die Aufgaben unter uns aufgeteilt.

Diesen 2. Teil der Arbeit haben wir doch schon unter einem grösseren Zeitdruck bewältigen müssen, damit wir noch in den angegebenen 300 Arbeitsstunden bleiben konnten. So sind einige Teile des Archivierungsprogramms und auch des Mailtools Kopien aus dem bestehenden Programm. Wir hatten uns ja auch noch um die Installationsroutine und den Betrieb des Programms in der ASTRA Umgebung zu kümmern. Hierfür haben wir ein Treffen mit dem Systemverantwortlichen vereinbart,

der uns auch empfangen hat und seine Vorstellungen der Installation vermittelt hat. So konnten wir termingerecht die komplette Software auf dem ASTRA-Server installieren.

Leider stellte sich heraus das es Herrn Berset nicht möglich war die Verwendung des Portfolios in der ASTRA durchzusetzen. So wird unsere Applikation sehr wahrscheinlich nie scharf eingesezt, nicht weil die Applikation fehlerhaft oder neben den Bedürfnissen funktioniert, sondern aus politischen Überlegungen die ich nicht nachvollziehen kann.

Wenn ich das Projekt nochmals in Angriff nehmen müsste würde auf jeden Fall versuchen die Team-Arbeit anders zu gestalten. Unsere Aufteilung war, glaube ich, für keinen von uns Beiden wirklich befriedigend. Vielleicht hätte ein klares Programm-Design im Vorfeld geholfen. Wobei so ein Design, mit neuer Technologie und einer für uns neuen Programmiersprache, sicherlich ein weiteres unsicheres, zeitaufwendiges Projekt gewesen wäre.

Schlussendlich kann ich sagen das dieses Projekt doch sehr lehrreich für mich war. Nicht nur das ich jetzt Visual Basic und MS Access kenne, so hab ich doch auch meine persönlichen Lehren gezogen. Unsere Arbeitsaufteilung gegen Ende das Marc tagsüber und ich nachts abwechslungsweise programmierten, hat mich doch zeitweise an meine Grenzen gebracht. Zumindest soweit, das ich in Zukunft so eine Arbeitsteilung nicht mehr akzeptieren werde.