

Semesterarbeit Mathias Mader (HIF09)

# **Projekt Cugar**

Projektdokumentation Cugar .....	3
1 Projektbeschrieb / Management Summary .....	3
1.1 Beschreibung und Begründung .....	3
1.2 Optionen.....	3
1.3 Nutzen.....	3
1.4 Risiken.....	4
1.5 Kosten .....	4
1.6 Erreichte Ergebnisse .....	4
1.7 Nicht erreichte Ergebnisse.....	4
2 Anforderungskatalog.....	5
2.1 Anwendungsfälle .....	6
2.2 UML Anwendungsfalldiagramm .....	10
2.3 Marktabklärung.....	10
3 Lösungsevaluation.....	11
3.1 IST - Situation.....	11
3.2 SOLL-Situation .....	11
3.3 Das Datenbanksystem MySQL.....	12
3.4 Evaluation des verwendeten Frameworks .....	12
3.5 Nutzwertanalyse der Frameworks.....	13
4 Projektplan .....	15
5 Technische Dokumentation der Lösung.....	16
5.1 Die Entwicklung .....	16
5.2 Identifizierung der relevanten Tabellen .....	22
5.3 Klassendiagramme .....	24
5.4 Bemerkungen zu den Klassendiagrammen .....	26
5.5 Installation von Cugar in der Produktiven Umgebung .....	27
5.6 Remotedesktop Demonstration von Cugar. ....	27
5.7 Cugar mit virtuellen Maschinen.....	33
5.8 Cugar Bedienungsanleitung.....	36
6 Anhänge .....	40
6.1 Generelles .....	40
6.2 Verwendete Quellen .....	40
6.3 Verwendete Software .....	40
6.4 VPN Verbindung einrichten in Windows XP .....	41
7 Hinweise und Unterschrift .....	46

# Projektdokumentation Cugar

## 1 Projektbeschrieb / Management Summary

### 1.1 Beschreibung und Begründung

Das Projekt "Cugar" zielt darauf, die Adressdaten der freien CRM Software "SugarCRM" und dem Warenbewirtschaftungsprogramm "CAO-Faktura" mit einem eigens entwickelten Programm zu verwalten. Das Ziel ist die redundanten Daten nur noch einmal zu erfassen, was letzten Endes die Effizienz steigert und eine grosse Fehlerquelle eliminiert. Das Ziel des ersten Releases ist primär, dass Adressdaten nur noch einmal erfasst werden müssen. Das Programm soll Unterschiede in den Datensätzen erkennen und die Datensätze auf Wunsch des Benutzers zusammenführen. Ein optionales Ziel ist es in Cugar eine Funktion zum Ausdrucken der Datensätze zu ermöglichen.

### 1.2 Optionen

Beide eingesetzten Systeme sind Open Source und somit frei erhältlich. Da beide Systeme völlig unabhängig voneinander entwickelt wurden gibt es bis anhin keine Oberfläche die beide Programme kombiniert. Bis jetzt müssen Adressdaten in beiden Systemen separat erfasst werden. Dies ist ineffizient und stellt eine unnötige Fehlerquelle dar. Um dieses Problem zu lösen gibt es folgende Möglichkeiten:

#### ***Umstieg auf eine ERP Lösung***

Bei der CalandaComp GmbH wird wo immer möglich freie Software eingesetzt. Einerseits aus ideellen Gründen, andererseits aus Kostengründen. Es gibt genügend freie ERP Systeme, einen Link zu einer Liste findet sich im Abschnitt "Marktabklärung". Der Wechsel von den bestehenden Systemen auf eine umfassende ERP Lösung wurde von der Geschäftsleitung aber im vorneherein verworfen, da eine Übernahme der Daten zu viel Zeit in Anspruch nehmen würde.

#### ***Die Option "Nichts tun"***

Da bis Anhin auf beiden Systemen gearbeitet wurde und alle Mitarbeiter damit vertraut sind kann durchaus auch ohne "Cugar" gearbeitet werden, weswegen die Option "Nichts tun" durchaus legitim ist. Desweiteren fehlen zum jetzigen Zeitpunkt für eine firmeninterne Eigenentwicklung die Ressourcen.

### 1.3 Nutzen

Der Nutzen des Programms ist sicherlich in einer Steigerung der Effizienz zu sehen. Gleichzeitig wird eine Fehlerquelle eliminiert. Die Datensätze wurden zwar grösstenteils von den Mitarbeitern korrekt erfasst, jedoch finden sich im täglichen Gebrauch doch ab und zu noch Unterschiede und Fehler.

## **1.4 Risiken**

Da in keinem der beiden Systeme Adressdaten gelöscht werden, besitzt Cugar keine Löschfunktion. Somit sind auch die Risiken der Benutzung von Cugar relativ klein. Der schlimmste Fall der eintreten könnte wäre, dass durch einen Fehler in der Benutzung ein Datensatz im CRM mit einem falschen Datensatz aus dem Warenbewirtschaftungssystem überschrieben wird. Aus diesem Grund wird der Benutzer mit einer Meldung gewarnt, bevor er Datensätze zusammenführt.

## **1.5 Kosten**

Da das Projekt im Rahmen einer Semesterarbeit erstellt wird, fallen für die Entwicklung keinerlei Kosten an. Sämtliche verwendeten Programme und Dienste (MySQL, CAO-Faktura, SugarCRM) sind frei erhältlich. Eine effektive Einführung in der CalandaComp GmbH würde allerdings noch Anpassungen am Programm erfordern, da die eingesetzte SugarCRM Version älter ist als die für das Projekt verwendete. Das heisst es müssten ggf. noch Anpassungen am Code vorgenommen werden um die Kommunikation mit der älteren SugarCRM Datenbank zu gewährleisten. Dieser Aufwand wird auf ca. 1 Personentag geschätzt.

## **1.6 Erreichte Ergebnisse**

- Cugar ist in der Lage Adressen in zwei verschiedenen Datenbanksystemen zu schreiben.
- Cugar ist in der Lage Adressen zu konsolidieren.
- Cugar ist in der Lage Adressen in zwei verschiedenen Datenbanksystemen gleichzeitig zu ändern.

## **1.7 Nicht erreichte Ergebnisse**

Es gelang nicht die Emailadressen in die SugarCRM Datenbank zu schreiben. Trotz mitloggen aller Befehle und einer Übersicht über alle Tabellen mit dem Datenbank Management Tool HeidiSQL konnte nicht ausfindig gemacht werden wie SugarCRM Email Adressen speichert und den jeweiligen Adressen zuordnet.

## 2 Anforderungskatalog

Titel	Anforderung	M/W/O	Prio
<b>Clientseitig</b>			
Windows Kompatibilität	Das Programm ist lauffähig auf Windows XP und Windows 7 Clients.	M	1
Datensätze einfügen	Neue Adressen für Privatpersonen sowie Firmen können erstellt werden.	M	1
Datensätze ändern	Bestehende Adressen von Privatpersonen oder Firmen sollen geändert werden können.	W	2
Datensätze löschen	Datensätze können vom Benutzer gelöscht werden.	O	3
Datensätze ausdrucken	Datensätze können ausgedruckt werden.	O	3
Datensätze exportieren	Datensätze können als .vcf Datei exportiert werden.	O	3
<b>Serverseitig</b>			
MySQL 4.1	Die Kommunikation mit MySQL 4.1 muss gewährleistet sein.	M	1
MySQL 5.1	Die Kommunikation mit MySQL 5.1 muss gewährleistet sein.	M	1
SugarCRM	Die Datensätze werden so gespeichert dass sie von SugarCRM weiterverarbeitet werden können.	M	1
CAO-Faktura	Die Datensätze werden so gespeichert dass sie von CAO-Faktura weiterverarbeitet werden können.	M	1
<b>Legende:</b>			
M = Musskriterium, W = Wunschkriterium, O = Optionale Weiterentwicklung 1 = Hohe Priorität, 2 = Mittlere Priorität, 3 = Niedrige Priorität			

Tabelle 1: Anforderungskatalog

## 2.1 Anwendungsfälle

### **Tabellenform:**

<b>Beschreibung Anwendungsfall „Adresse erstellen“</b>	
Name	Projekt "Cugar"
Kurzbeschreibung	Ein Benutzer möchte eine neue Adresse erfassen.
Akteure	Angestellte der Firma CalandaComp.
Auslöser	Ein Angestellter erfasst einen neuen Kunden oder eine neue Firma.
Ergebnisse	Die eingetragenen Daten werden Fehlerfrei auf beide Datenbanken übernommen.
Vorbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Nachbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Eingehende Daten	Neue Adresse, eingetragen in Hauptformular.
Essenzieller Ablauf	<ul style="list-style-type: none"><li>- Cugar wird gestartet.</li><li>- Adresse wird eingegeben.</li><li>- Der Speichern Knopf wird betätigt.</li><li>- Die eingegebenen Daten werden durch Cugar korrekt in die beiden Datenbanken geschrieben.</li></ul>
Offene Punkte	Email Adresse wird in der SugarCRM Datenbank nicht gespeichert.
Sonstiges, Anmerkungen	

Tabelle 2: Anwendungsfall "Adresse erstellen"

<b>Beschreibung Anwendungsfall „Adresse ändern“</b>	
Name	Projekt "Cugar"
Kurzbeschreibung	Ein Benutzer möchte eine bestehende Adresse ändern.
Akteure	Angestellte der Firma CalandaComp
Auslöser	Ein Angestellter möchte eine Adresse ändern.
Ergebnisse	Die geänderten Daten werden fehlerfrei auf beide Datenbanken übernommen.
Vorbedingung(en)	<ul style="list-style-type: none"> <li>- Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.</li> <li>- Der Benutzer sucht eine bestehende Adresse mit Hilfe des Suchformulars.</li> <li>- Der Benutzer wählt zwei korrespondierende Datensätze aus</li> <li>- Cugar lädt die entsprechenden Adressdaten in das Hauptformular.</li> </ul>
Nachbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Eingehende Daten	Die Adressdaten inklusive der geänderten Werte.
Essenzieller Ablauf	<ul style="list-style-type: none"> <li>- Cugar wird gestartet.</li> <li>- Der Benutzer öffnet das Suchformular und sucht nach einer Adresse.</li> <li>- Die korrespondierenden Datensätze werden durch den Benutzer im Suchformular markiert und mit einem Click auf „Laden / Merge“ in das Hauptformular geladen.</li> <li>- Cugar prüft ob Unterschiede vorhanden sind und gibt ggf. eine Meldung aus.</li> </ul>
Offene Punkte	
Sonstiges, Anmerkungen	

Tabelle 3: Anwendungsfall Adresse ändern

<b>Beschreibung Anwendungsfall „Adresse suchen“</b>	
Name	Projekt "Cugar"
Kurzbeschreibung	Ein Benutzer möchte eine Adresse suchen.
Akteure	Angestellte der Firma CalandaComp.
Auslöser	Ein Angestellter möchte prüfen ob eine Adresse schon vorhanden ist (um diese ggf. zu ändern)
Ergebnisse	Entsprechende Suchresultate aus den beiden Datenbanken gemäss dem angewählten Suchkriterium.
Vorbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Nachbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Eingehende Daten	Suchbegriff, Suchkriterium
Essenzieller Ablauf	<ul style="list-style-type: none"> <li>- Cugar wird gestartet.</li> <li>- Suchbegriff wird in die Personen oder Firmensuche eingegeben.</li> <li>- Das Suchformular wird geöffnet und die Suchresultate angezeigt.</li> </ul>
Offene Punkte	
Sonstiges, Anmerkungen	

Tabelle 4: Anwendungsfall "Adresse suchen"



<b>Beschreibung Anwendungsfall „Datensatz laden“</b>	
Name	Projekt "Cugar"
Kurzbeschreibung	Ein Benutzer möchte einen Datensatz in das Hauptformular laden um die entsprechenden Daten zu sehen/kopieren/ändern.
Akteure	Angestellte der Firma CalandaComp.
Auslöser	Der Benutzer möchte die Daten aus dem Suchformular im Hauptformular angezeigt haben.
Ergebnisse	Die Werte aus den gewählten Datensätzen werden in das Hauptformular geladen.
Vorbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.  Der Benutzer sucht nach einem Datensatz über das Suchformular, welchen er in das Hauptformular laden möchte.
Nachbedingung	Cugar ist gestartet und mit beiden Datenbanksystemen verbunden.
Eingehende Daten	Zwei angewählte Datensätze aus dem Suchformular.
Essenzieller Ablauf	<ul style="list-style-type: none"> <li>- Cugar wird gestartet.</li> <li>- Es wird nach einer Adresse über das Suchformular gesucht.</li> <li>- Die beiden korrespondierenden Datensätze werden durch den Benutzer markiert.</li> <li>- Der Benutzer klickt auf „Laden“. Der Datensatz wird in das Hauptformular geladen.</li> </ul>
Offene Punkte	
Sonstiges, Anmerkungen	

Tabelle 5: Anwendungsfall "Datensatz laden"

## 2.2 UML Anwendungsfalldiagramm

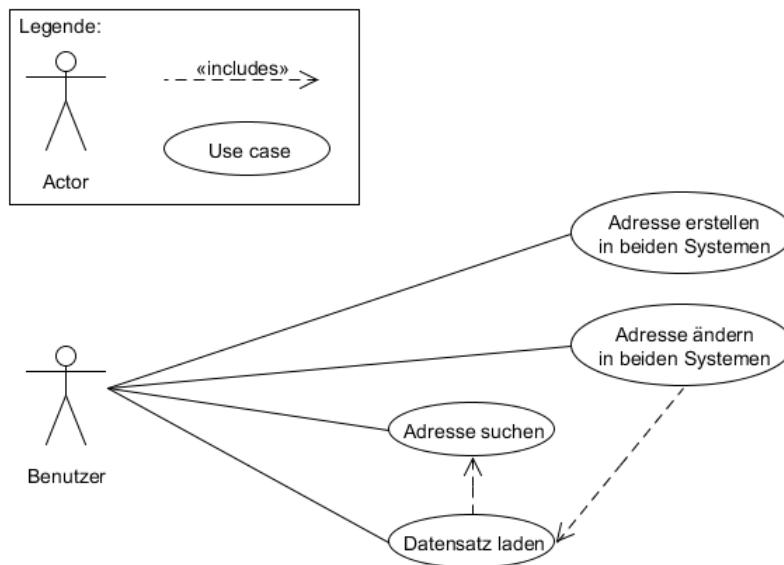


Abbildung 1: UML Anwendungsfalldiagramm

## 2.3 Marktabklärung

Die CalandaComp GmbH hat sich bereits bei der Gründung dazu entschieden, wo immer möglich, freie Software einzusetzen. Dies einerseits aus Kostengründen, andererseits aber auch, weil wir die Prinzipien der freien Software respektieren und schätzen.

Die Wahl fiel damals auf CAO-Faktura als freies Warenbewirtschaftungssystem und SugarCRM als CRM System. Es wurde bis anhin auf beiden Systemen parallel gearbeitet. Ein Ziel des Projektes Cugar ist, redundante Daten in einem Vorgang in beide Systeme einzutragen.

Da es auf dem Markt kein vergleichbares Produkt gibt wird an dieser Stelle auf eine Marktanalyse verzichtet.

Die einzige Option um die beiden Systeme zu kombinieren resp. durch ein einzelnes System zu ersetzen wäre der Umstieg auf eine ERP Lösung. Diese Option wurde von der Geschäftsleitung jedoch von vorne herein abgelehnt, da die Datenübernahme zu viele Ressourcen in Anspruch nehmen würde. Für die interessierte Leserschaft gibt es jedoch auf Wikipedia eine Liste mit freier ERP Software.

- [http://en.wikipedia.org/wiki/List\\_of\\_ERP\\_software\\_packages](http://en.wikipedia.org/wiki/List_of_ERP_software_packages)

### 3 Lösungsevaluation

#### 3.1 IST - Situation

Die beiden System CAO-Faktura und SugarCRM laufen auf zwei verschiedenen Servern mit jeweils unterschiedlichen MySQL Versionen als Datenbank.

Beide Systeme werden von den jeweiligen Client Programmen bedient. CAO-Faktura hat einen auf Windows XP/7 Systemen lauffähigen Client im Lieferumfang. SugarCRM wird über ein Web-Frontend, welches auf einem dedizierten Intranet Server läuft angesprochen.

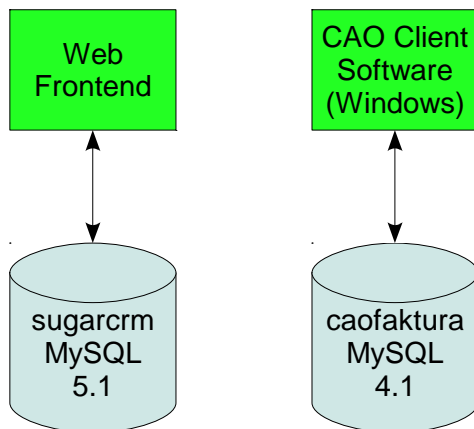


Abbildung 2: Schema IST-Situation

Das Problem bei der jetzigen Situation besteht darin, dass die Adressen (Privatkunden, Firmenkunden) in beiden Systemen erfasst werden müssen. Dieser Umstand soll durch die Entwicklung von Cugar geändert werden.

#### 3.2 SOLL-Situation

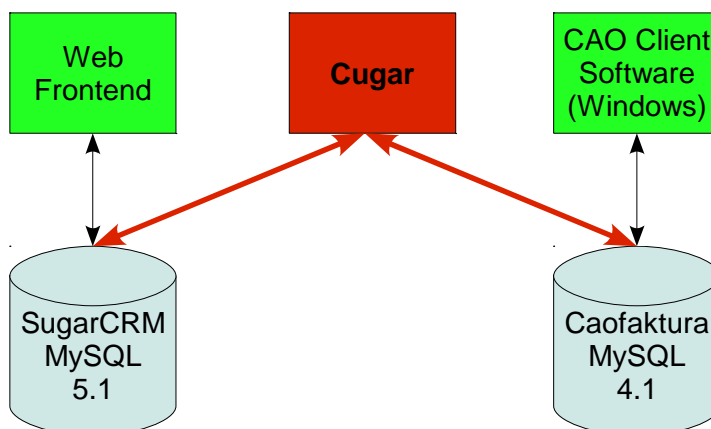


Abbildung 3: Schema SOLL Situation

Cugar soll eine Schnittstelle zwischen den beiden Systemen bilden und die redundanten Daten auf den beiden Datenbanken abgleichen.

### **3.3 Das Datenbanksystem MySQL**

MySQL ist ein weit verbreitetes, offenes Datenbanksystem dass von der Oracle Corporation gefördert wird. Es ist für alle gängigen Betriebssysteme (Mac OS, BSD, Linux und Windows) frei erhältlich. Deshalb ist davon auszugehen dass MySQL auch in Zukunft eine wichtige Rolle in der Datenbankwelt spielen wird.

### **3.4 Evaluation des verwendeten Frameworks**

MySQL Datenbanksysteme können durch die von Oracle angebotenen Connectoren von Java, HTML/PHP, C/C++ und dem .NET Framework angesprochen werden. Die nachfolgende Nutzwertanalyse soll veranschaulichen warum .NET als Zielframework gewählt wurde. Beachten Sie, dass es bei der Entscheidungsfindung nur um nicht funktionale Kriterien handelt, da die funktionalen Anforderungen (siehe Anforderungskatalog) von allen bewerteten Technologien erfüllt werden könnten.

### 3.5 Nutzwertanalyse der Frameworks

Nutzwertanalyse möglicher Frameworks									
Kriterium	Gewicht	.NET		C/C++		Java		HTML / PHP	
		Bewertung	Nutzen	Bewertung	Nutzen	Bewertung	Nutzen	Bewertung	Nutzen
<b>Nicht Funktionale Kriterien:</b>	<b>50</b>								
Bedienung mit Maus	7	10	70	10	70	10	70	10	70
Bedienung mit Tastaturkommandos	8	10	80	10	80	10	80	3	24
Benötigte Einarbeitungs- und Entwicklungszeit	10	10	100	3	30	3	80	3	30
Erweiterbarkeit	9	10	90	9	81	9	81	4	36
			170		131		156		80
<b>Technische Kriterien:</b>	<b>50</b>								
Kompatibel zu MySQL	10	10	100	10	100	10	100	10	100
Lauffähig auf Windows Clients (XP / 7)	10	10	100	10	100	10	100	10	100
			100		100		100		100
<b>Total Nutzen (Wirksamkeitskennziffer W)</b>	<b>100</b>		<b>270</b>		<b>231</b>		<b>256</b>		<b>180</b>

Tabelle 6: Nutzwertanalyse möglicher Frameworks

Die Nutzwertanalyse zeigt klar dass .NET das am besten geeignete Framework ist. Der entscheidende Faktor ist hierbei die erwartete Einarbeitungs- und Entwicklungszeit. Auch das Erweitern des Programms dürfte Aufgrund der bisher gemachten Erfahrungen mit dem .NET Framework am einfachsten von statten gehen.

### ***Vorteile von .NET***

.NET ist für Windows Umgebungen die wohl beste Wahl in Sachen Entwicklungsumgebung. Von Microsoft selbst entwickelt, unterstützt .NET sämtliche systeminternen Vorgänge und bietet viele Erweiterungsmöglichkeiten durch eigene Klassen und durch das Hinzufügen von Klassen von anderen Anbietern. Die Anfertigung der grafischen Oberfläche geschieht in .NET intuitiv mit der frei erhältlichen Entwicklungsumgebung Visual Studio Express.

### ***Nachteile von .NET***

.NET wird bis jetzt nur von Windows Umgebungen vollständig unterstützt. Für Linux Systeme ist eine Nachbildung mit dem Namen „Mono“ erhältlich, jedoch kann Mono das .NET Framework nur teilweise abbilden und ist noch in einem relativ frühen Entwicklungsstadium. Da Cugar aber nur auf Windows Clients lauffähig sein muss, muss dies nicht als Nachteil betrachtet werden.

### ***Risiken***

Es besteht eine grosse Abhängigkeit zu Microsoft. Dies ist in der Hinsicht ein Risiko, als dass die Entwicklung des .NET Frameworks von Microsoft gestoppt werden könnte. In diesem Fall müsste Cugar neu geschrieben werden. Zur Zeit gibt es aber keine Anzeichen seitens Microsoft dafür, dass das in naher Zukunft geschehen wird. Deswegen kann das Risiko als sehr klein eingestuft werden.

## 4 Projektplan

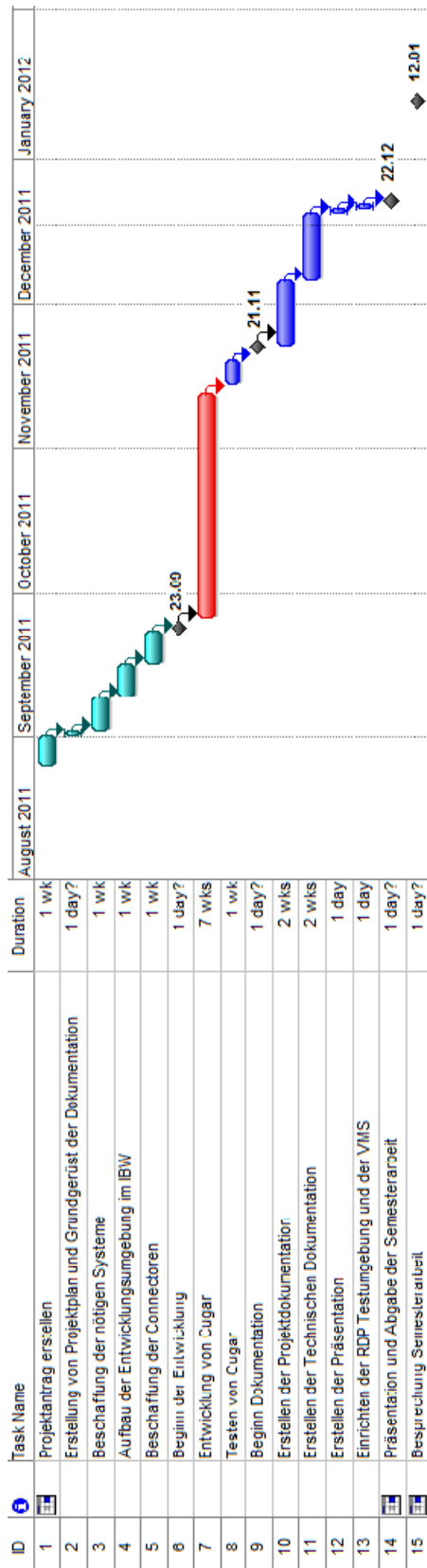


Abbildung 4: Der Projektplan

## 5 Technische Dokumentation der Lösung

### 5.1 Die Entwicklung

Für die Entwicklung wurden zwei virtuelle Server mit Virtualbox (<http://www.virtualbox.org>) aufgesetzt, welche später die Datenbanken von CAO-Faktura und SugarCRM enthalten sollen.

#### *Die Konfiguration der virtuellen Maschinen*

##### **Der Host „sugarcrm“:**

Hostname	sugarcrm
Netzwerkadapter 1:	Dynamisch (dhcp), Bridged Mode
Netzwerkadapter 2:	192.168.56.102, Host-only-Network
Betriebssystem:	Centos 6
MySQL Version:	mysql-server-5.1.52-1.el6_0.1.i686
Zusätzliche Software:	httpd (Apache), SugarCRM Community Edition Version 6.3.0beta6 (Build 6874)

Tabelle 7: Spezifikationen des Hosts "sugarcrm"

##### **Der Host „caofaktura“:**

Hostname	caofaktura
Netzwerkadapter 1:	192.168.56.101, Host-only-Network
Betriebssystem:	Windows XP Service Pack 3
MySQL Version:	mysql-server-4.1.22-1.win-nt
Zusätzliche Software:	CAO-Faktura

Tabelle 8: Spezifikationen des Hosts "caofaktura"

#### **Die Entwicklungsumgebung**

Das Programm wurde mit Microsoft Visual C# Express 2008 und Microsoft Visual Studio 2008 Professional entwickelt. Mehr Informationen zu dieser Entwicklungsumgebung finden Sie auf: <http://www.microsoft.com/germany/express/> resp. auf

Als Management Tool für die Datenbanken wurde HeidiSQL eingesetzt. Informationen und Download unter: <http://www.heidisql.com/>

Als weiteres Management Tool für die Datenbanken und für die generierung der Tabellenschemas in dieser Dokumentation wurde MySQL Workbench 5.2 eingesetzt. Informationen und download unter: <http://dev.mysql.com/downloads/workbench/>



## **Das Datenbanksystem MySQL**

Als Beschreibung soll ein Zitat von Wikipedia herhalten: „MySQL ist ein relationales Datenbankverwaltungssystem. Es ist als Open-Source-Software sowie als kommerzielle Enterpriseversion für verschiedene Betriebssysteme verfügbar und bildet die Grundlage für viele dynamische Webauftritte.“

MySQL ist kostenlos erhältlich und in den Repositories aller bekannten Linux Distributionen enthalten. Auch für alle anderen gängigen Betriebssysteme (FreeBSD, Solaris Linux, Mac OSX und Microsoft Windows) gibt es Installationspakete. Zu den Features von MySQL gehören u.a.:

- Kostenlos erhältlich
- Plattformunabhängig
- Viele Datentypen
- Replikationsfähig
- Clusteringfähig
- Kompatibel zur ANSI SQL Sprache
- Unterstützt mehrere Speicherengines (InnoDB, MyISAM, etc.)

Eine vollständige Liste der Features erhalten Sie mit der offiziellen MySQL Dokumentation: <http://dev.mysql.com/doc/>

## **Die verwendeten Connectoren**

### **Generelles**

Zur Kommunikation von Visual Studio mit den verwendeten Datenbanken werden sogenannte Connectoren benötigt. Das Finden der richtigen Connectoren für die Kommunikation von Visual Studio mit den Datenbanken erwies sich schwieriger als erwartet. Da MySQL ein Konkurrenzprodukt zu Microsofts SQL Server darstellt wird MySQL von Haus aus von Visual Studio nicht unterstützt. Es mussten zuerst geeignete Connectoren für die beiden Datenbanken gefunden werden.

### **MySQL Connector/.NET v. 6.3.7**

Die Verbindung zur MySQL 5.1 Datenbank wird mit dem offiziellen .NET Connector von MySQL bewerkstelligt. Da die aktuelle Version 6.4.4 beim Verbinden zu Problemen geführt hat wurde die ältere, jedoch stabilere Version 6.3.7 gewählt. Zu finden unter:

<http://dev.mysql.com/downloads/connector/>

### **OLEDB MySQL Provider**

Da CAO-Faktura auf der MySQL Version 4.1 basiert, welche veraltet ist und nicht mehr unterstützt wird, musste hier eine andere Lösung für die Verbindung gesucht werden. Der erste Lösungsansatz bot sich mit dem Einsatz eines kommerziellen Produkts. Dem MySQL OLE DB Provider von Cherry City Software, zu finden unter:

<http://cherrycitysoftware.com/ccs/providers/ProvMySQL.aspx>

Der gratis Testzeitraum dieser Software beträgt 30 Tage, zu wenig Zeit um Cugar damit fertig zu stellen. Da der Vertrieb des Produkts nicht auf Emails bezüglich Studentenrabatt reagiert hat musste nach einem anderen Weg gesucht werden.

## MySQL Connector/ODBC

Um doch noch auf die MySQL 4.1 Datenbank zugreifen zu können wurde eine ältere und langsamere Technologie für die Verbindung gewählt: ODBC.

Für diese Art der Verbindung gibt es von MySQL einen offiziellen Connector/ODBC inklusive den nötigen Treibern. Zu finden unter:

<http://dev.mysql.com/downloads/connector/>

### **Authentifizierung von CAO-Faktura in MySQL 4.1 aktivieren**

CAO-Faktura basiert eigentlich auf einer noch älteren MySQLVersion, nämlich der Version 3.5. Diese verwendet eine andere Authentifizierungsmethode als MySQL 4.1. Um diesen Umstand zu ändern muss MySQL 4.1 dahingehend umkonfiguriert werden die alte Authentifizierungsmethode zu akzeptieren. Dazu loggt man sich als root in die betreffende MySQL 4.1 Datenbank ein und gibt in der MySQL Console folgende zwei Befehle ein:

```
mysql> update user set password=old_password('12345') where user='root';  
Query OK, 2 rows affected (0.00 sec)  
Rows matched: 2  Changed: 2  Warnings: 0  
  
mysql> flush privileges;  
Query OK, 0 rows affected (0.00 sec)
```

Abbildung 5: MySQL 4.1 Auszug aus dem CAO-Faktura Installationshandbuch

Anschliessend kann CAO-Faktura wie vorgesehen auf die Datenbank zugreifen.

### Die Funktionsweise von Cugar

Cugar soll als Schnittstelle der beiden Systeme CAO-Faktura und SugarCRM dienen. Um Adressen korrekt in die beiden Datenbanken zu schreiben muss Cugar in der Lage sein die jeweils richtigen Tabellen und Spalten zu verwenden sowie die korrekten Datenformate für die einzelnen Zellen einzuhalten. Da die beiden verwendeten Datenbanken jeweils unterschiedliche Tabellen und Parameter für Privat- und Firmenkunden besitzen ist eine genaue Unterscheidung unverzichtbar. Für Änderungen an bestehenden Datensätzen muss im Programmcode darauf geachtet werden dass der aktuell gewählte Datensatz in der jeweiligen Datenbank identifiziert werden kann. Cugar identifiziert die aktuell gewählten Datensätze anhand des Primary Key der jeweiligen Tabelle.

Jenachdem in welche Tabelle geschrieben wird muss auch der entsprechend eingesetzte MySQL Befehl gewählt werden. Auf den nächsten paar Seiten werden die wichtigsten Abläufe des Programmes in Form von UML Aktivitätsdiagrammen genauer erläutert.

### Neue Datensätze einfügen

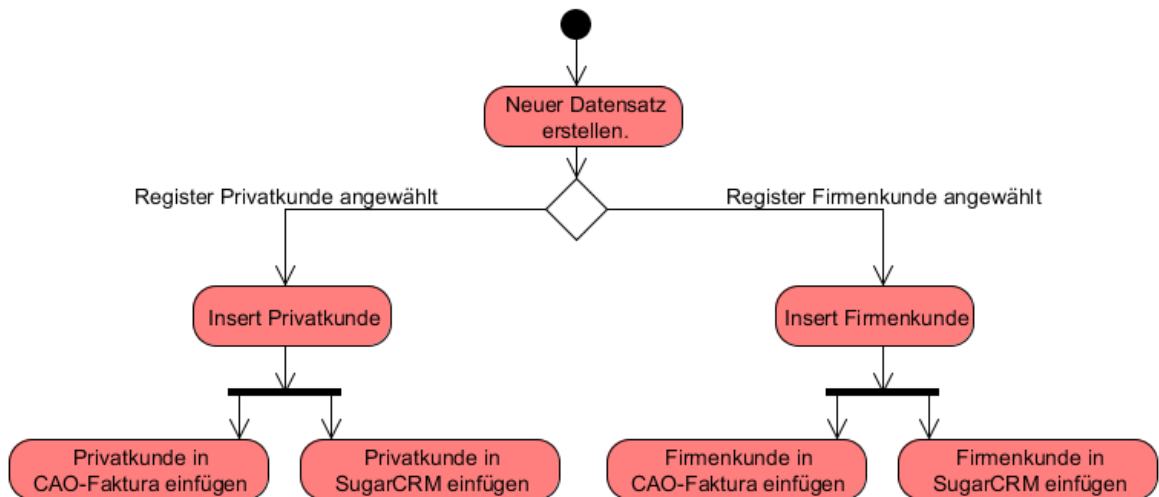


Abbildung 6: UML Aktivitätsdiagramm „Neuer Datensatz einfügen“

### Bemerkungen zum Diagramm „Neuer Datensatz einfügen“:

Es wird anhand des aktuell angewählten Registers entschieden ob es sich um einen Privat- oder um einen Firmenkunden handelt. Anschliessend wird der entsprechende Insert Befehl für die CAO-Faktura Datenbank und die SugarCRM Datenbank aufgerufen. Für beide Systeme existieren zwei Insert Methoden. Jeweils eine für Privat- und eine für Firmenkunden.

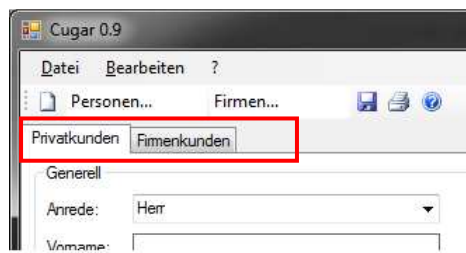


Abbildung 7: Die Register "Privatkunden" und "Firmenkunden"

### Bestehende Datensätze ändern

Wird eine bestehende Adresse geändert, sieht der Ablauf schon etwas komplizierter aus. Das vereinfachte UML Aktivitätsdiagramm zeigt den wesentlichen Ablauf dieses Vorgangs:

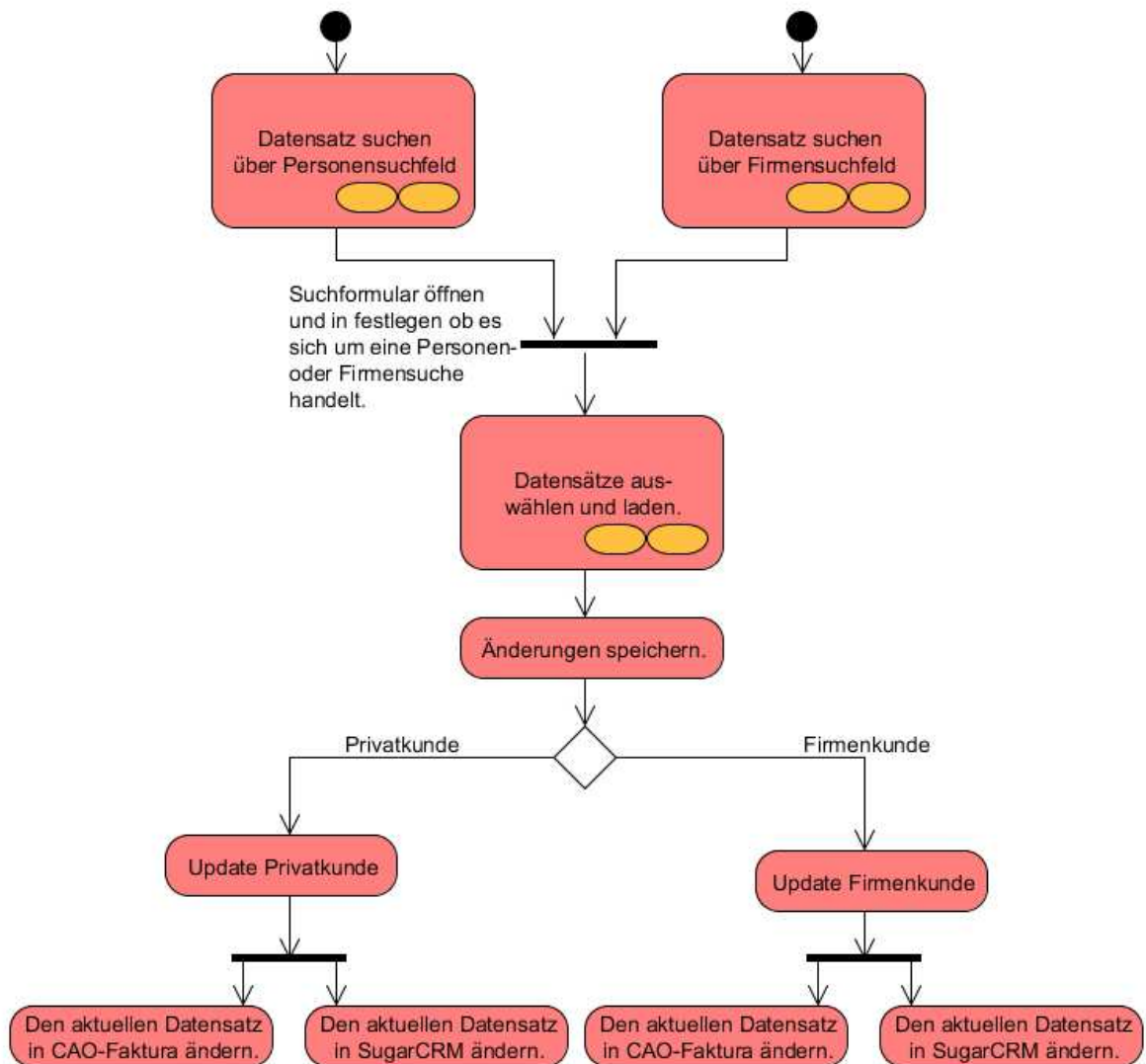


Abbildung 8: UML Aktivitätsdiagramm „Datensatz ändern“

#### Bemerkungen zum Diagramm „Datensatz ändern“:

**Datensatz suchen:** Bei der Suche nach einem Datensatz wird das Suchformular geöffnet und anhand des verwendeten Suchfelds entschieden ob es sich um eine Suche nach Personen oder Firmen handelt. Die entsprechenden Suchanfragen werden gestartet und in zwei DataGridViews dargestellt.

**Datensätze auswählen und laden:** Der Benutzer wählt die korrespondierenden Datensätze aus den Suchresultaten und klickt auf „Laden / Merge“. Das Programm überprüft, ob Unterschiede in den Datensätzen bestehen und lässt dem Benutzer die Wahl, ob er die Datensätze zusammenführen resp. ins Mainform laden will. Wenn Unterschiede bestehen

wird der Datensatz aus CAO-Faktura bevorzugt und der entsprechende SugarCRM Datensatz beim Speichern überschrieben.

**Änderungen speichern:** Die im Suchformular gewählten Datensätze wurden in jeweils eine Cao Tabelle und einer Sugar Tabelle im DataSet hinterlegt. Der Primary Key der Datensätze wird herausgelesen und in der jeweiligen Update Methode verwendet um zu bestimmen welcher Datensatz in der jeweiligen Datenbank überschrieben wird. Auch hier existieren für das Ändern von CAO-Faktura und SugarCRM Datensätzen vier Methoden. Für beide Systeme jeweils eine für Privatkunden und eine für Firmenkunden.

## 5.2 Identifizierung der relevanten Tabellen

Beide Datenbanksysteme enthalten eine Vielzahl an Tabellen. Um die relevanten Tabellen zu identifizieren wurde die Protokollierung der beiden Datenbanken dahingehend geändert dass jeder abgesetzte MySQL Befehl der Clients protokolliert wird. Durch Ausführen der gewünschten Aktionen in CAO-Faktura und dem SugarCRM Web Frontend konnten durch das Befehlsprotokoll die relevanten Tabellen ausfindig gemacht werden.

### Cao-Faktura

Die CAO-Faktura Datenbank enthält insgesamt 73 Tabellen. Für die Semesterarbeit existiert aber nur eine relevante Tabelle. Die Tabelle „adressen“. Sie enthält sämtliche Adressen der Privatkunden sowie der Firmenkunden. Zwischen Privat- und Firmenadressen wird in CAO-Faktura einzig durch das Attribut „Kundengruppe“ unterschieden.

1	REC_ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AU...
2	MATCHCODE	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"
3	KUNDENGRUPPE	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
4	SPRACH_ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
5	GESCHLECHT	CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'
6	KUNNUM1	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
7	KUNNUM2	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
8	NAME1	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
9	PLZ	VARCHAR	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
10	ORT	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
11	LAND	VARCHAR	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
12	NAME2	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
13	NAME3	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
14	ABTEILUNG	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
15	ANREDE	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
16	STRASSE	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
17	POSTFACH	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
18	PF_PLZ	VARCHAR	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
19	DEFAULT_LIEF...	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-1
20	GRUPPE	VARCHAR	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
21	TELE1	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
22	TELE2	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
23	FAX	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
24	FUNK	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
25	EMAIL	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
26	EMAIL2	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
27	INTERNET	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
28	DIVERSES	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Abbildung 9: Die Tabelle "adressen" von CAO-Faktura

## SugarCRM

Der Aufbau der SugarCRM Datenbank ist um einiges komplexer, sie enthält 107 Tabellen. Auch hier wurden durch das Ändern der Protokollierungseinstellungen die relevanten Tabellen identifiziert. Dabei handelt es sich um die Tabelle „contacts“ für die Privatkunden und die Tabelle „accounts“ für Firmenkunden. Beide Tabellen sind unterschiedlich aufgebaut.

contacts	accounts
<ul style="list-style-type: none"> <li>id CHAR(36)</li> <li>date_entered DATETIME</li> <li>date_modified DATETIME</li> <li>modified_user_id CHAR(36)</li> <li>created_by CHAR(36)</li> <li>description TEXT</li> <li>deleted TINYINT(1)</li> <li>assigned_user_id CHAR(36)</li> <li>salutation VARCHAR(255)</li> <li>first_name VARCHAR(100)</li> <li>last_name VARCHAR(100)</li> <li>title VARCHAR(100)</li> <li>department VARCHAR(255)</li> <li>do_not_call TINYINT(1)</li> <li>phone_home VARCHAR(100)</li> <li>phone_mobile VARCHAR(100)</li> <li>phone_work VARCHAR(100)</li> <li>phone_other VARCHAR(100)</li> <li>phone_fax VARCHAR(100)</li> <li>primary_address_street VARCHAR(150)</li> <li>primary_address_city VARCHAR(100)</li> <li>primary_address_state VARCHAR(100)</li> <li>primary_address_postalcode VARCHAR(20)</li> <li>primary_address_country VARCHAR(255)</li> <li>alt_address_street VARCHAR(150)</li> <li>alt_address_city VARCHAR(100)</li> <li>alt_address_state VARCHAR(100)</li> <li>alt_address_postalcode VARCHAR(20)</li> <li>alt_address_country VARCHAR(255)</li> <li>assistant VARCHAR(75)</li> </ul>	<ul style="list-style-type: none"> <li>id CHAR(36)</li> <li>name VARCHAR(150)</li> <li>date_entered DATETIME</li> <li>date_modified DATETIME</li> <li>modified_user_id CHAR(36)</li> <li>created_by CHAR(36)</li> <li>description TEXT</li> <li>deleted TINYINT(1)</li> <li>assigned_user_id CHAR(36)</li> <li>account_type VARCHAR(50)</li> <li>industry VARCHAR(50)</li> <li>annual_revenue VARCHAR(100)</li> <li>phone_fax VARCHAR(100)</li> <li>billing_address_street VARCHAR(150)</li> <li>billing_address_city VARCHAR(100)</li> <li>billing_address_state VARCHAR(100)</li> <li>billing_address_postalcode VARCHAR(20)</li> <li>billing_address_country VARCHAR(255)</li> <li>rating VARCHAR(100)</li> <li>phone_office VARCHAR(100)</li> <li>phone_alternate VARCHAR(100)</li> <li>website VARCHAR(255)</li> <li>ownership VARCHAR(100)</li> <li>employees VARCHAR(10)</li> <li>ticker_symbol VARCHAR(10)</li> <li>shipping_address_street VARCHAR(150)</li> <li>shipping_address_city VARCHAR(100)</li> <li>shipping_address_state VARCHAR(100)</li> <li>shipping_address_postalcode VARCHAR(20)</li> <li>shipping_address_country VARCHAR(255)</li> </ul>
5 more...	3 more...
Indexes	Indexes

Abbildung 10: Die relevanten Tabellen in SugarCRM

### 5.3 Klassendiagramme

Hier sind die Diagramme der relevanten Klassen aufgeführt die von mir angelegt oder modifiziert worden sind.

#### Die Formulklassen

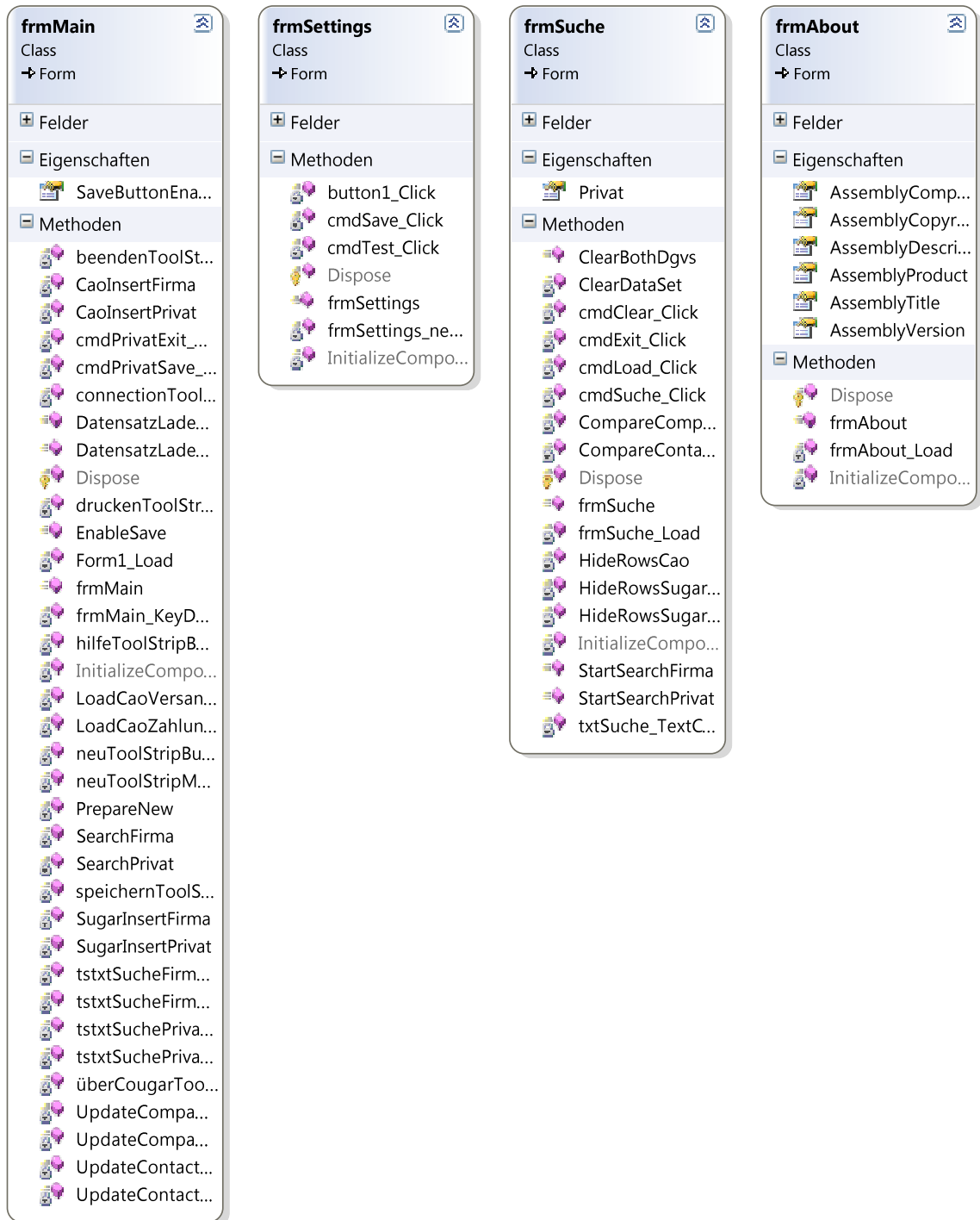


Abbildung 11: Die Formulklassen. Die Felder wurden zur besseren Übersicht ausgeblendet



## Die formularlosen Klassen

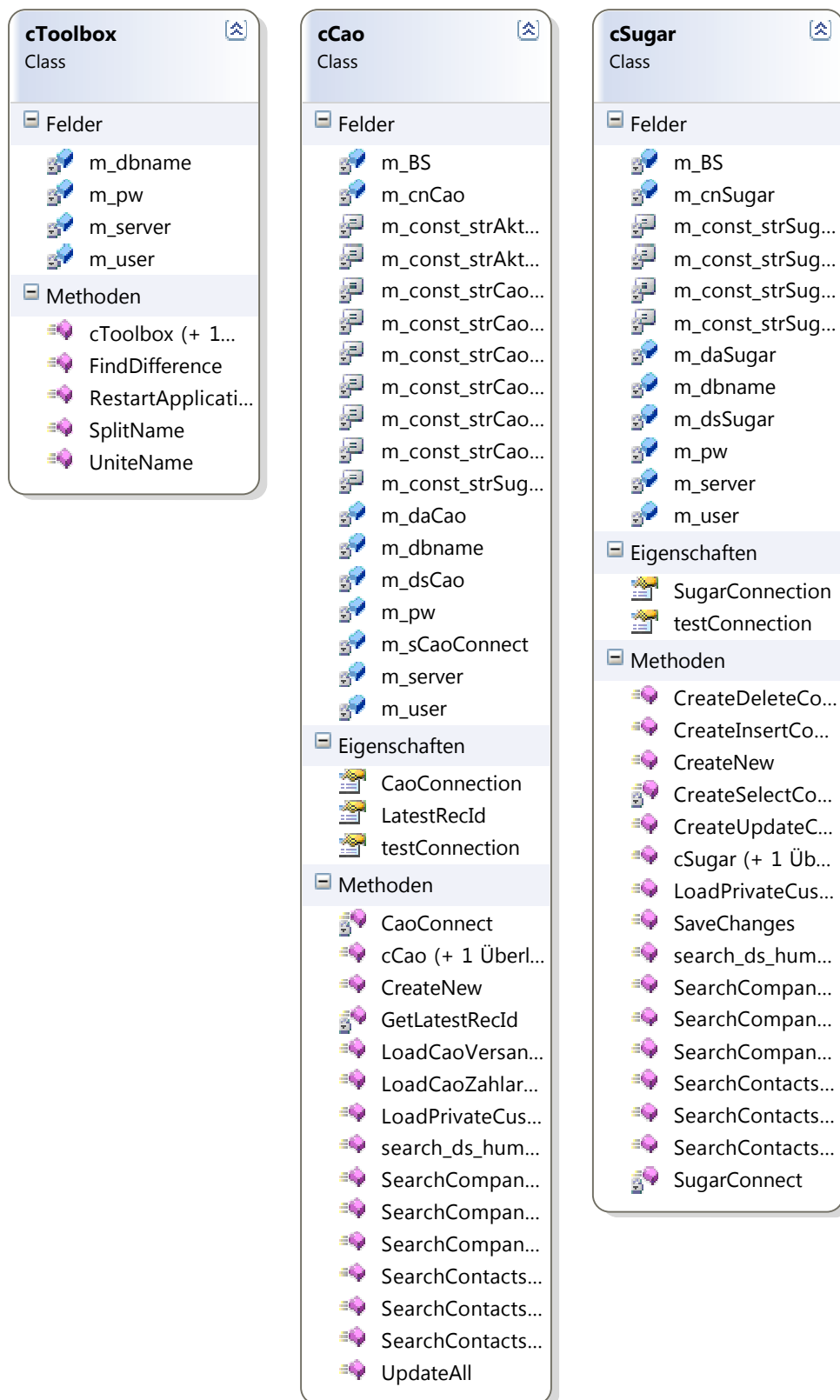


Abbildung 12: Die formularlosen Klassen

## 5.4 Bemerkungen zu den Klassendiagrammen

### **Die Formulare**

Cugar wird grundsätzlich über das Hauptformular (frmMain) und das Suchformular (frmSuche) bedient. Für den Betrieb sind jedoch insgesamt 4 Formulare relevant. In diesem Abschnitt sollen die wesentlichen Funktionen der Formulare aufgezeigt werden.

#### **frmMain.cs**

Das Hauptformular enthält ein DataSet und eine BindingSource, welche bei der Arbeit mit den Datensätzen benötigt werden. Diese werden jeweils an die cCao resp. cSugar Objekte mit Referenz auf das Original weitergegeben. So wird sichergestellt dass die Daten immer verfügbar sind. Ebenfalls enthält das Hauptformular die jeweils vier INSERT- und UPDATE-Methoden, welche die Daten in den Datenbanken eintragen oder ändern.

#### **frmSuche.cs**

Das Suchformular. Jenachdem aus welchem Suchfeld das Formular gestartet wird setzt es die entsprechenden Suchbefehle für Privat- oder Firmenkunden ab und stellt die Resultate in zwei DataGridViews dar. Ebenfalls kann das jeweilige Suchkriterium angepasst werden, worauf auch hier die entsprechenden Suchmethoden aufgerufen werden.

#### **frmSettings.cs**

Das Formular frmSettings.cs enthält die Einstellungen des Programms und kann diese ggf. testen und als Benutzerdaten auf der Festplatte speichern.

#### **frmAbout.cs**

Das Formular frmAbout.cs enthält lediglich Informationen über das Programm und den Autor.

### **Formularlose Klassen**

#### **Die Klassen cCao.cs und cSugar.cs**

Zur Kommunikation mit den jeweiligen Datenbanken sind die Klassen cCao.cs und cSugar.cs verantwortlich. Sie erhalten beide beim Erzeugen ein aus dem Hauptformular referenziertes DataSet und eine BindingSource. Dies hat den Vorteil, dass von jedem Punkt im Programm aus auf die relevanten Datensätze zugegriffen werden kann. Ebenso stellen die Klassen cCao.cs und cSugar.cs zu jedem Zeitpunkt eine Verbindung für die jeweiligen Datenbanken zur Verfügung.

Die UPDATE- und INSERT- Methoden sind jedoch in der Klasse frmMain.cs eingebettet, da so die Daten einfacher aus den Textfeldern ausgelesen und ausgewertet werden können.

#### **Die Klasse cToolbox.cs**

Die Klasse cToolbox enthält ein paar eigens geschriebene Methoden um beispielsweise Vor- und Nachnamen zu verbinden und wieder zu trennen.

## 5.5 Installation von Cugar in der Produktiven Umgebung

### **Voraussetzungen:**

Für den Betrieb von Cugar im Netzwerk werden folgende Voraussetzungen gestellt:

- Windows XP oder Windows 7 PC mit installiertem .NET Framework Version 3.5.
- Eine lauffähige SugarCRM Instanz im lokalen Netzwerk. *Siehe Abschnitt SugarCRM.*
- Eine lauffähige CAO-Faktura Instanz im lokalen Netzwerk. *Siehe Abschnitt CAO-Faktura.*
- MySQL Connector v. 6.3.7, Downloadlink:  
<http://www.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-6.3.7.msi/from/ftp://ftp.solnet.ch/mirror/mysql/> oder von [www.mysql.com](http://www.mysql.com)

Die Installation von Cugar erfolgt über den von Microsoft VisualStudio 2008 generierten Installer.

### **SugarCRM:**

- Download: <http://www.sugarcrm.com/crm/download>
- Offizielle Installations- und Bedienungsanleitung:  
[http://www.sugarcrm.com/crm/support/documentation/SugarCommunityEdition/6.3/-docs-Application\\_Guides-Sugar\\_Community\\_Edition\\_Application\\_Guide\\_6.3.0GA-toc.html](http://www.sugarcrm.com/crm/support/documentation/SugarCommunityEdition/6.3/-docs-Application_Guides-Sugar_Community_Edition_Application_Guide_6.3.0GA-toc.html)

### **CAO-Faktura:**

- Download: <http://www.cmr-faktura.de/index.php/caosupport/cao-downloads>
- Offizielle Installations- und Bedienungsanleitung: Wird mit CAO-Faktura mit installiert.

### **Die eigentliche Installation:**

Installieren Sie den MySQL Connetor/NET, zu finden auf dem USB Stick. Doppelklicken Sie dazu „mysql-connector-net-6.3.7.msi“ und wählen Sie die Installationsvariante „Full“.

Anschliessend öffnen Sie „setup.exe“ und bestätigen Sie die Installation. Nach erfolgreicher Installation finden Sie Cugar in Ihrem Startmenü im Ordner „Project Cugar“.

## 5.6 Remotedesktop Demonstration von Cugar.

Für eine einfachere Begutachtung des Programms wurde im Firmennetzwerk der CalandaComp GmbH eine lauffähige Verison von Cugar installiert, die per Remotedesktop ohne Einschränkungen genutzt werden kann. Um sich das Programm auf diese Weise anzusehen, müssen Sie zuerst eine VPN Verbindung ins CalandaComp GmbH Netzwerk aufbauen. Anschliessend können Sie über Ihren gewohnten Remotedesktopclient auf einen eigens dafür eingerichteten Windows XP Professional Remotedesktop Server zugreifen und das Programm nach Belieben testen. Eine detaillierte Anleitung zur Einrichtung der VPN Verbindung finden Sie auf den nachfolgenden Seiten.

### ***Erstellen einer VPN- Verbindung ins CalandaComp GmbH Netzwerk:***

Diese Anleitung resp. deren Bilder beziehen sich auf Microsoft Windows 7. Sollten Sie noch Windows XP verwenden nutzen Sie bitte die Anleitung welche sich im Anhang der Projektarbeit befindet.

Um in Windows 7 eine VPN Verbindung einzurichten, öffnen Sie als erstes Ihr „Netzwerk- und Freigabecenter“ und befolgen Sie die nächsten Schritte:

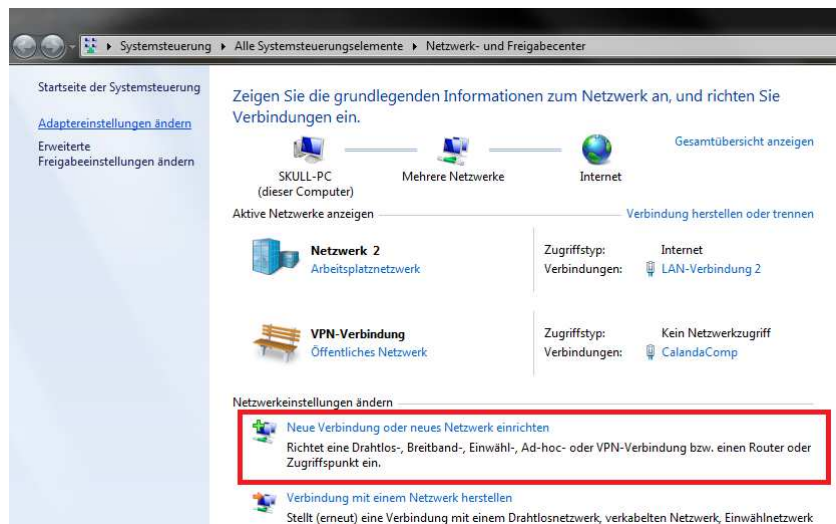


Abbildung 13: Klicken Sie auf "Neue Verbindung oder neues Netzwerk einrichten"

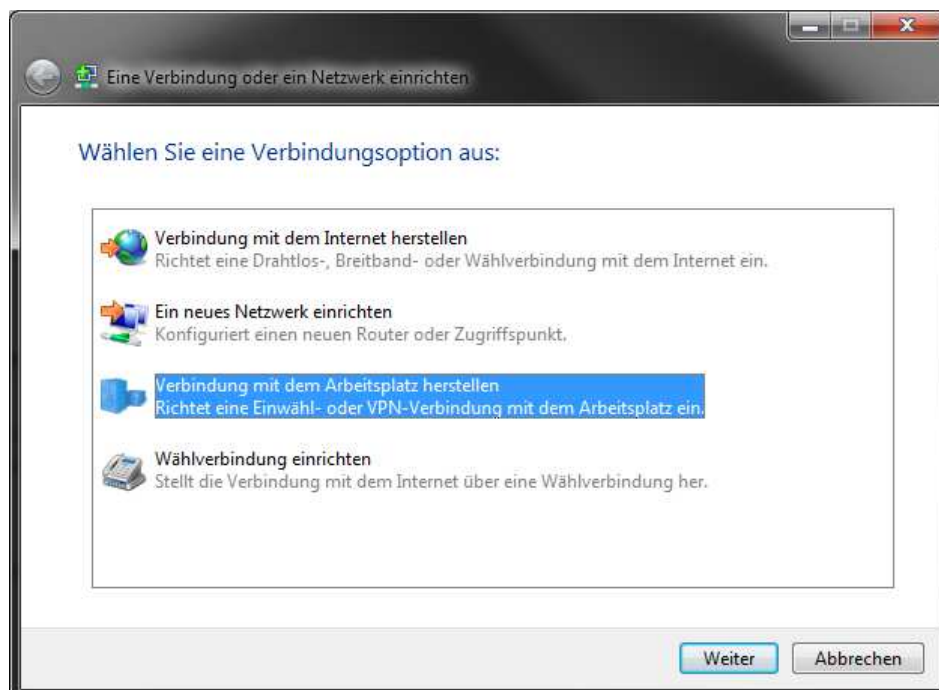


Abbildung 14: Klicken Sie auf "Verbindung mit Arbeitsplatz herstellen"



Abbildung 15: Klicken Sie auf "Die Internetverbindung (VPN) verwenden"

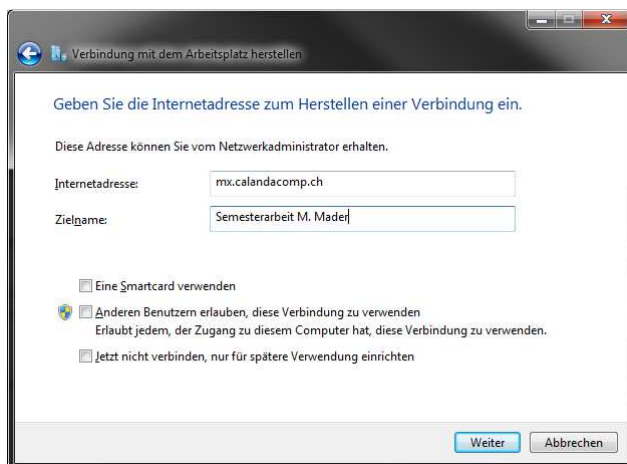


Abbildung 16: Als **Internetadresse:** tragen Sie "mx.calandacomp.ch" ein. Geben Sie der Verbindung einen treffenden Namen und klicken Sie auf "**Weiter**"

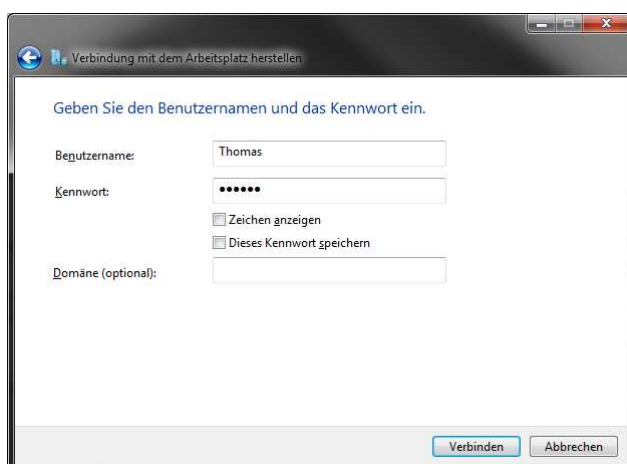


Abbildung 17: Geben Sie hier Ihre Zugangsdaten ein. (siehe nächste Seite)

Benutzername	Passwort
Thomas	Thomas
Silvio	Silvio
Peter	Peter

Tabelle 9: Benutzertabelle für die VPN Verbindung

### Vorsicht:

Im nachfolgenden Bildschirm versucht Windows bereits eine Verbindung herzustellen, brechen Sie diesen Vorgang ab indem Sie auf „**Überspringen**“ klicken! Wir müssen zuerst noch unterbinden, dass der gesamte Internetverkehr über die neu erstellte Verbindung geleitet wird. Gehen Sie dazu wie folgt vor:



Abbildung 18: Öffnen Sie die Eigenschaften Ihrer neu erstellten Verbindung.

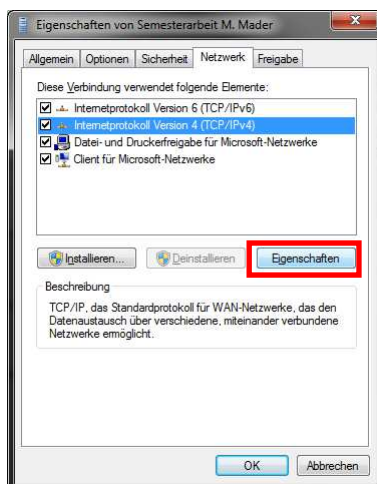


Abbildung 19: Öffnen Sie das Register „Netzwerk“, wählen Sie „Internetprotokoll Version 4 (TCP/IPv4)“ und klicken Sie auf „Eigenschaften“

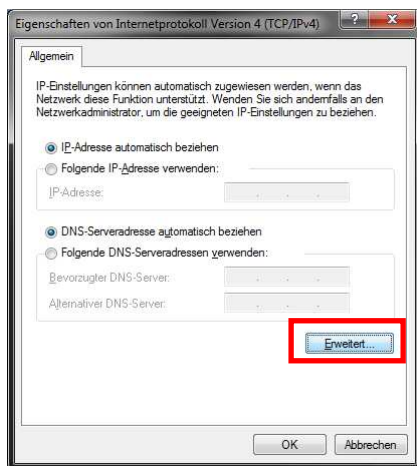


Abbildung 20: In diesem Dialogfeld klicken Sie auf „Erweitert“

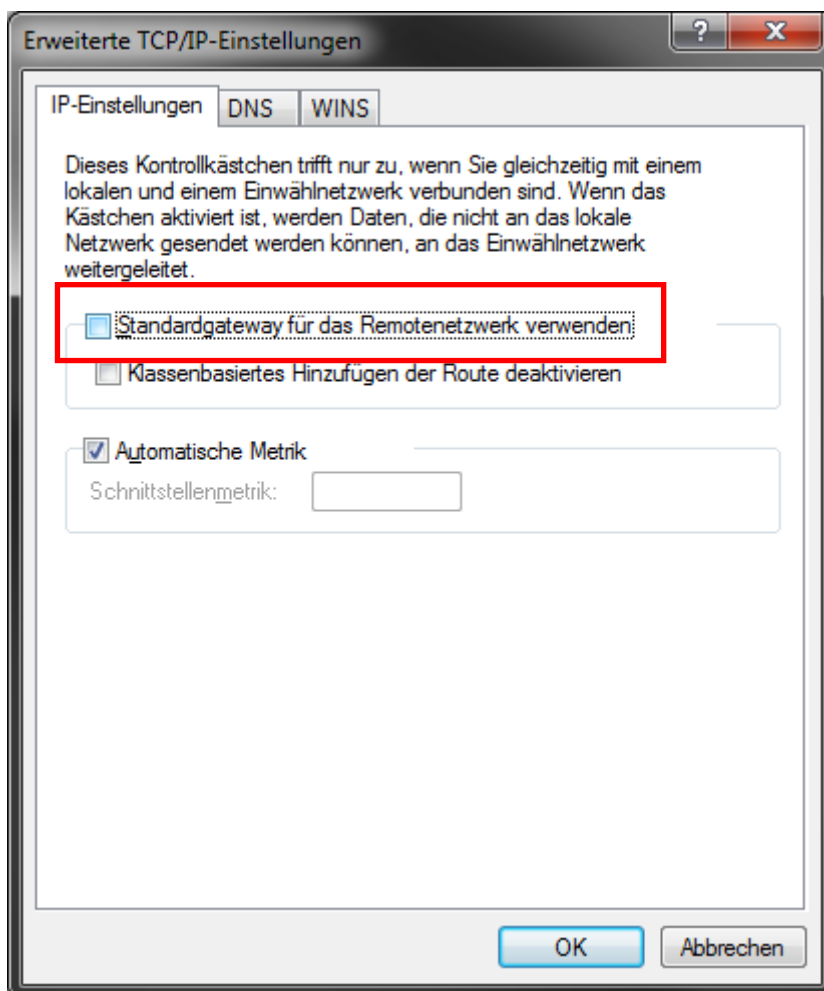


Abbildung 21: Entfernen Sie den Haken bei "Standardgateway für das Remotenetzwerk verwenden"

Bestätigen Sie anschliessend mit „OK“ um die Einstellungen zu speichern. Ihre neu erstellte Verbindung müsste nun im Netzwerk Widget von Windows 7 ersichtlich sein.

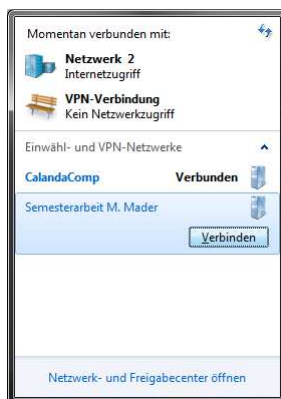


Abbildung 22: Starten Sie Ihre neu erstellte Verbindung mit einem Klick auf "Verbinden"

### **Starten der Remotedesktop Verbindung:**

Starten sie die Remotedesktop mit einem klick auf Start -> Ausführen und starten Sie „mstsc.exe“

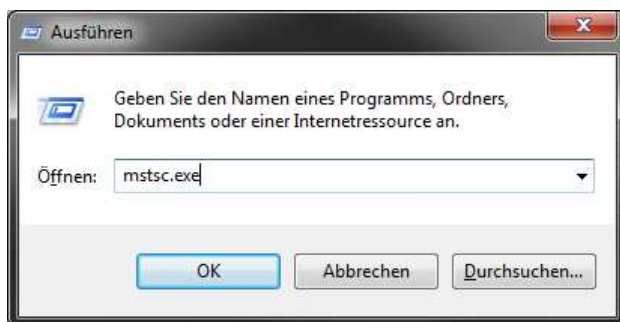


Abbildung 23: Geben Sie "mstsc.exe" ein und Drücken Sie die Eingabetaste

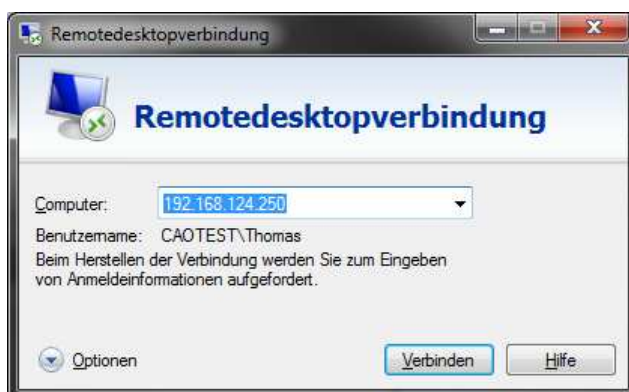


Abbildung 24: Geben Sie "192.168.124.250" als Serveradresse ein.



Die Logindaten des Remotedesktopservers sind mit denen der VPN Verbindung identisch:

Benutzername	Passwort
Thomas	Thomas
Silvio	Silvio
Peter	Peter

Tabelle 10: Logindaten RDP Server

Anschliessend öffnet sich Ihr persönlicher Desktop, wo Cugar sowie CAO-Faktura und SugarCRM bereits lauffähig installiert sind.

Um Cugar zu starten öffnen Sie einfach die entsprechende Verknüpfung auf dem Desktop.

Falls Sie einen Blick auf das SugarCRM werfen möchten öffnen Sie einfach den Internetbrowser Ihrer Wahl. Die Logindaten sind die selben wie die für den Remotedesktop Server selber.

Um sich CAO-Faktura genauer anzusehen benützen Sie die entsprechende Desktop Verknüpfung. Auch hier sind die Zugangsdaten identisch mit denen des Remotedesktop Servers.

## 5.7 Cugar mit virtuellen Maschinen

Im Lieferumfang der Dokumentation ist ein USB Stick enthalten aufdem alles nötige bereits auf zwei virtuellen Maschinen vorkonfiguriert ist.

### Maschine 1 (winxp.lan.darktemple.ch.vmx)

Name:	Caotest
Betriebssystem:	Windows XP Professional 32-bit SP3
Installierte Dienste:	MySQL 4.1.22 win, Remote desktop
Installierte Programme:	Cugar, CAO-Faktura

Tabelle 11: Konfiguration der Windows XP VM

### Maschine 2 (sugarcrm.lan.darktemple.ch)

Name:	sugarcrm
Betriebssystem:	CentOS 6 i686
Installierte Dienste:	httpd-2.2.15-5, MySQL v5.1.52
Installierte Programme:	SugarCRM Community Edition v. Version 6.3.0beta6 (Build 6874)

Tabelle 12: Konfiguration der CentOS 6 VM

### **VirtualBox: Installation**

Zum Betrieb der Virtuellen Maschinen wird VirtualBox benötigt. Das Installationsprogramm befindet sich ebenfalls auf dem mitgelieferten USB Stick oder zum downloaden unter folgender Adresse:

- <https://www.virtualbox.org/wiki/Downloads>

Laden Sie sich die Version die Ihrem Betriebssystem entspricht herunter und installieren Sie es mit den Standard Einstellungen. Windows Nutzer müssen die Treiberinstallation für die virtuellen Netzwerk Interfaces bestätigen da die Treiber nicht signiert sind.

### **VirtualBox: Maschinen importieren.**

Nachdem Sie VirtualBox installiert haben, starten Sie das Programm über die Desktopverknüpfung.

Wählen Sie im Menü „Datei“ den Eintrag „Appliance importieren...“

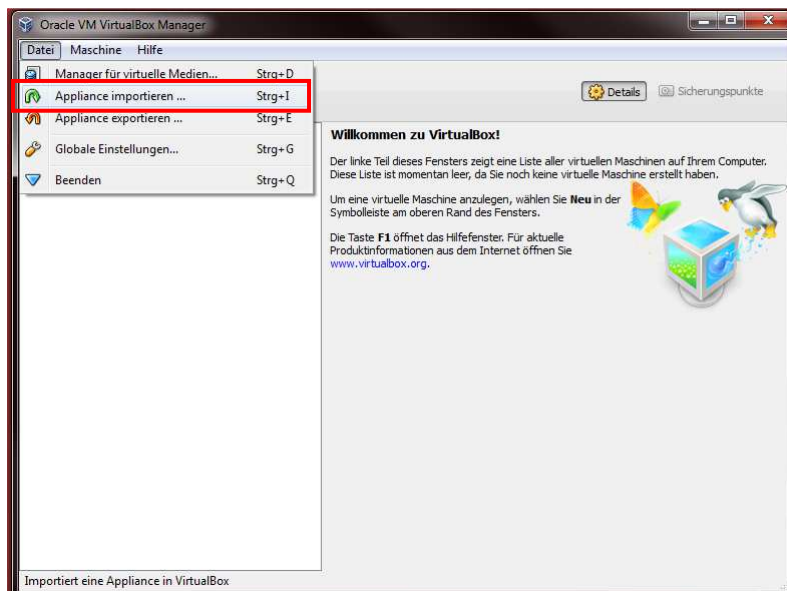


Tabelle 13: Appliance importieren...

Wählen Sie im nachfolgenden Dialog die entsprechende vm auf dem USB Stick aus. Und importieren Sie diese. Der Vorgang kann je nach Leistung des Computers ein paar Minuten dauern.

In der Windows XP Maschine sind CAO-Faktura, Cugar sowie ein Link zum SugarCRM Bereits hinterlegt. Die Zugangsdaten für Windows, Cao und SugarCRM lauten wie folgt:

<b>Benutzername:</b>	<b>Passwort:</b>
Thomas	Thomas
Silvio	Silvio
Peter	Peter

Tabelle 14: Benutzerdaten der virtuellen Windows Maschine

Da die SugarCRM VM nur als MySQL 5.1 Server dient sind für diese Maschine keinerlei Logindaten erforderlich. Die einzige Bedingung ist, dass die Maschine gestartet ist.

## 5.8 Cugar Bedienungsanleitung

### *Der erste Start*

Beim ersten Start von Cugar erscheint der Dialog mit den Einstellungen wo Sie jeweils den Hostnamen resp. die IP Adresse der beiden Server sowie die root Zugangsdaten und die jeweilige Datenbank angeben müssen.

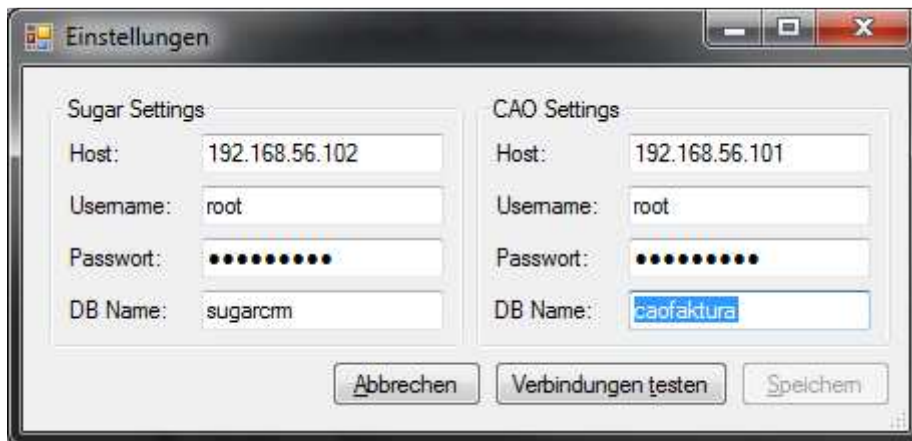


Abbildung 25: Der Einstellungsdialog

Nachdem Sie die nötigen Angaben eingetragen haben, klicken Sie auf „Verbindungen testen“ um sicherzustellen dass die Daten auch korrekt sind. Falls die Daten nicht stimmen wird Cugar nicht funktionieren! Wenn die Einstellungen korrekt getestet wurden, klicken Sie auf „Speichern“ um ins Hauptfenster zu gelangen. (Hinweis: In der VM- sowie der RDP Variante wurde das Programm bereits konfiguriert und dieser Schritt entfällt)

### *Das Hauptfenster*

Das Hauptfenster besteht aus dem Menü, der Toolbar und den entsprechenden Anzeige/Eingabefelder für Privatkunden und Firmenkunden.

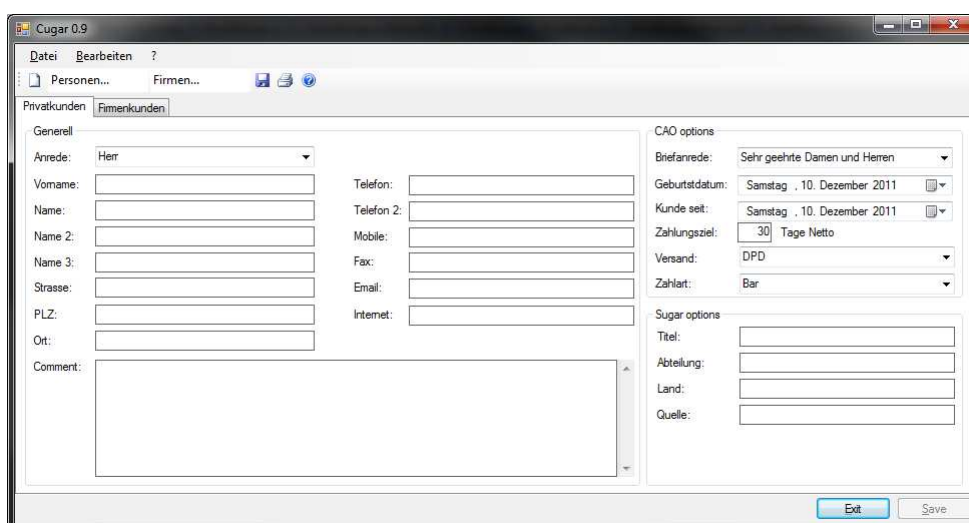


Abbildung 26: Das Cugar Hauptfenster

## Die Toolbar



Abbildung 27: Die Toolbar

Die Funktionen der Toolbar::



Der Button „Neuen Datensatz erstellen“: Mit diesem Knopf erstellen Sie einen neuen Datensatz. Cugar fragt ob die bestehenden Textfelder gelöscht wurden. Siehe hierzu „Datensätze erstellen“

**Suchfelder** Mit den beiden Suchfeldern „Personen...“ und „Firmen...“ können Sie entsprechende Datensätze in den beiden Datenbanken suchen. Geben Sie einen Suchbegriff ein und drücken Sie auf „Enter“ um das Suchformular zu öffnen. Siehe hierzu den Abschnitt „Das Suchfenster“



Der „Speichern“ Button: Mit diesem Button können Sie Änderungen in den Datensätzen speichern. Falls kein Datensatz aus der Suchmaske geladen wurde, fragt Cugar nach, ob ein neuer Datensatz erstellt werden soll.



Der „Drucken“ Button: In der Version 0.9 ist die Druckfunktion leider noch nicht implementiert. Die Funktion soll in einem späteren Release nachgereicht werden.

## Die Menüs

- **Das Menü „Datei“**

Neu...	Schaltet den „Save“ Button frei, mit dem Datensätze angelegt werden können. Das gleiche kann auch über den Button „Neuen Datensatz erstellen...“ in der Toolbar erreicht werden
--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Beenden	Beendet Cugar
---------	---------------

- **Das Menü „Bearbeiten“**

Einstellungen...	Hier können die Einstellungen für die jeweiligen Datenbankverbindungen angepasst werden.
------------------	------------------------------------------------------------------------------------------

- **Das Menü „?“**

Über...	Zeigt einen Info Dialog über Cugar an.
---------	----------------------------------------

## Einen Datensatz erstellen

Um einen Datensatz zu erstellen, klicken Sie in der Toolbar einfach auf „Neuen Datensatz erstellen...“ oder wählen Sie im Menü „Datei“ -> „Neu...“. Sie werden gefragt, ob die Textfelder gelöscht werden sollen oder nicht. Anschliessend schaltet sich der „Save“ Button rechts unten frei. Geben Sie ihre Daten entsprechend ein und klicken Sie anschliessend auf „Save“, um in den beiden Datenbanken einen neuen Datensatz anzulegen. Cugar erkennt automatisch, ob es sich um einen Kontakt oder eine Firma handelt.

## Einen Datensatz suchen

Geben Sie einfach einen Nachnamen resp. einen Firmennamen in den entsprechenden Feldern in der Toolbar ein und drücken Sie Enter. Anschliessend erscheint das Suchfenster welches Ihnen auch gleich die treffenden Suchresultate aufzeigt.

## Das Suchformular

Suchbegriff:  ☒ Name ☐ Strasse ☐ Telefon

Resultate:  
CAO-Faktura:

	NAME1	PLZ	ORT	ANREDE	STRASSE	POSTFACH	PF_PLZ	TELE1	TEL2
	Roland Mader	8889	Plons	Herr	Plonserfeldstr. 32				
	Mathias Mader	8889	Plons	Herr	Plonserfeldstr. 33			081 723 69 10	081
*									

SugarCRM:

	salutation	first_name	last_name	primary_address_st	primary_address_cil	primary_address_st	primary_address_pc	primary_address_cc
	Herr	Mathias	Mader	Plonserfeldstr. 33	Plons		8889	Schweiz
	Herr	Roland	Mader	Plonserfeldstr. 32	Plons		8889	Schweiz
*								

Abbildung 28: Das Suchfenster

Mit den Checkboxes „Name“, „Strasse“ und „Telefon“ können die Suchkriterien angegeben werden. Die oberen Suchresultate sind jeweils die Datensätze der CAO Datenbank, die Resultate im unteren Bereich sind jeweils diejenigen aus der Sugar Datenbank.

Tipp: Um alle Datensätze sehen zu können, lassen Sie das Suchfeld einfach leer und klicken Sie auf „Suchen“

## Die Buttons:

Suche: Startet eine neue Suche.

Clear: Löscht die Suchresultate

Laden / Merge: Hiermit können Sie die Suchdaten in das Hauptfenster laden. Dafür muss in beiden Suchfenstern jeweils der gewünschte Datensatz markiert sein. Die Ladefunktion überprüft auf eventuelle Unstimmigkeiten der beiden Datensätze und meldet dies entsprechend dem Benutzer. Der Benutzer kann selber entscheiden ob er die Unstimmigkeit überschreiben möchte oder nicht. Der Datensatz aus dem CAO Suchfenster hat automatisch Vorrang.

Abbrechen: Bricht die Suche ab und schliesst das Suchfenster.

***Einen Datensatz ändern:***

Um einen Datensatz zu ändern, suchen Sie die passenden Datensätze über die Suche und klicken Sie auf „Laden / Merge“. Der Datensatz wird in das Hauptformular übernommen und kann dort nach Belieben angepasst werden. Ein Klick auf das „Speichern“ Symbol in der Toolbar übernimmt die Änderungen in die beiden Datenbanken.

## 6 Anhänge

### 6.1 Generelles

Titel	Link
Projekthomepage (englisch, wurde während der Erstellung der Arbeit hauptsächlich als „Notizblock“ von mir verwendet.	<a href="http://www.darktemple.ch/wiki/doku.php?id=projects:cugar">http://www.darktemple.ch/wiki/doku.php?id=projects:cugar</a>
Sourcecode:	<a href="http://www.github.com/SchleimKeim/Cugar">http://www.github.com/SchleimKeim/Cugar</a>

Tabelle 15: Generelles

### 6.2 Verwendete Quellen

Titel	Link
Microsoft Developer Network (MSDN)	<a href="http://msdn.microsoft.com/">http://msdn.microsoft.com/</a>
Die offizielle MySQL Dokumentation:	<a href="http://dev.mysql.com/doc/">http://dev.mysql.com/doc/</a>
Die Connector/NET Dokumentation:	<a href="http://dev.mysql.com/doc/refman/5.0/en/connector-net.html">http://dev.mysql.com/doc/refman/5.0/en/connector-net.html</a>
Die CAO-Faktura Dokumentation	(wird mit CAO-Faktura mit installiert)
Unterrichtsmaterial von Thomas Gall	
Unterrichtsmaterial von Silvio Burger	
Vorlage für Semesterarbeit von Peter Reinmann	
Wikipedia	<a href="http://www.wikipedia.org">www.wikipedia.org</a>

Tabelle 16: Verwendete Quellen

### 6.3 Verwendete Software

Titel	Link
CAO-Faktura:	<a href="http://www.cmr-faktura.de/index.php/caosupport/cao-downloads">http://www.cmr-faktura.de/index.php/caosupport/cao-downloads</a>
SugarCRM:	<a href="http://www.sugarcrm.com/crm/download">http://www.sugarcrm.com/crm/download</a>
Microsoft Visual C# Express 2008:	<a href="http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express">http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express</a> (Der Link ist für die 2010er Version)
Microsoft Visual Studio 2008 Professional	<a href="http://www.microsoft.com/austria/visualstudio/products/team/visual-studio-professional.aspx">http://www.microsoft.com/austria/visualstudio/products/team/visual-studio-professional.aspx</a> (Link für die 2010er Version)
MySQL Server:	<a href="http://www.mysql.com/downloads/mysql/">http://www.mysql.com/downloads/mysql/</a>
MySQL Connector/NET:	<a href="http://www.mysql.com/downloads/connector/net/">http://www.mysql.com/downloads/connector/net/</a>
MySQL Connector/ODBC:	<a href="http://www.mysql.com/downloads/connector/odbc/">http://www.mysql.com/downloads/connector/odbc/</a>
MySQL Workbench:	<a href="http://www.mysql.com/downloads/workbench/">http://www.mysql.com/downloads/workbench/</a>
HeidiSQL:	<a href="http://www.heidisql.com/">http://www.heidisql.com/</a>
Windows XP SP3:	<a href="http://windows.microsoft.com/en-US/windows/products/windows-xp">http://windows.microsoft.com/en-US/windows/products/windows-xp</a>
Windows 7 Home:	<a href="http://www.microsoft.com/de-de/windows/default.aspx">http://www.microsoft.com/de-de/windows/default.aspx</a>
Centos Linux:	<a href="http://www.centos.org">http://www.centos.org</a>

Tabelle 17: Verwendete Software



## 6.4 VPN Verbindung einrichten in Windows XP

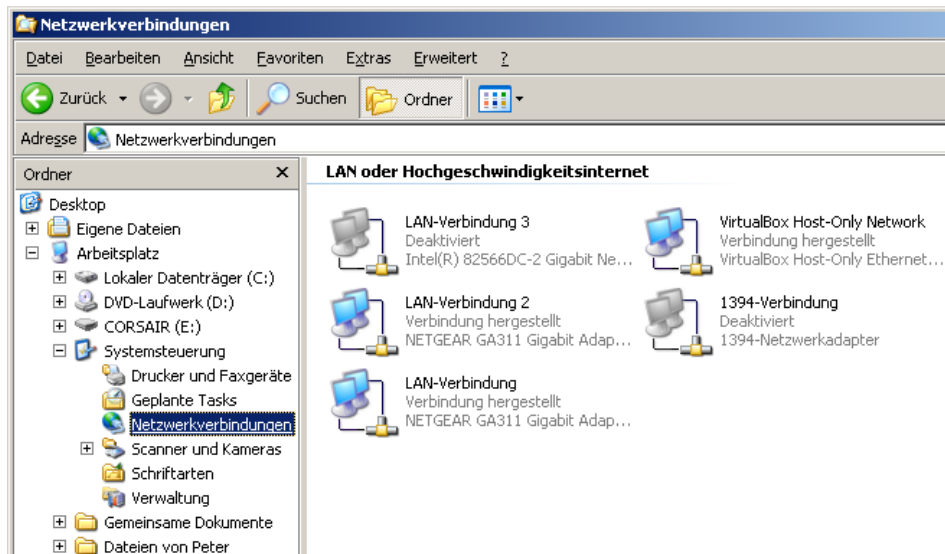


Abbildung 29: Öffnen Sie die "Systemsteuerung" und gehen Sie auf "Netzwerkverbindungen"

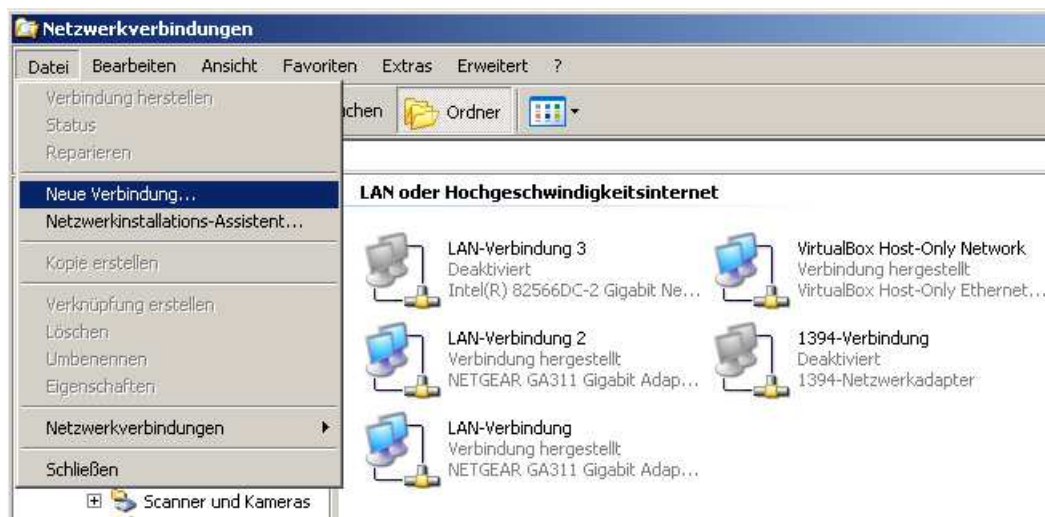


Abbildung 30: Wählen Sie im Menü "Datei" -> "Neue Verbindung..."

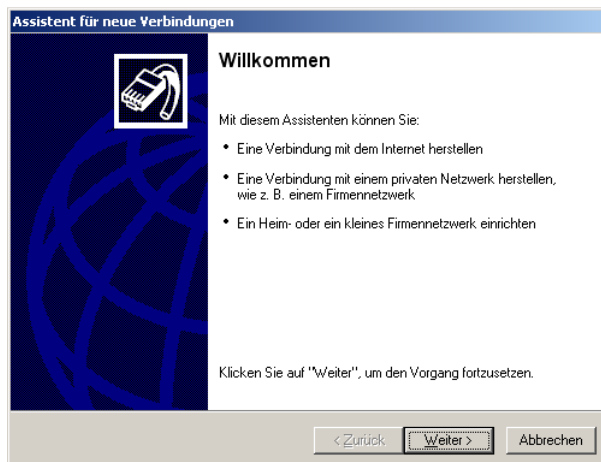


Abbildung 31: Klicken Sie auf "Weiter"

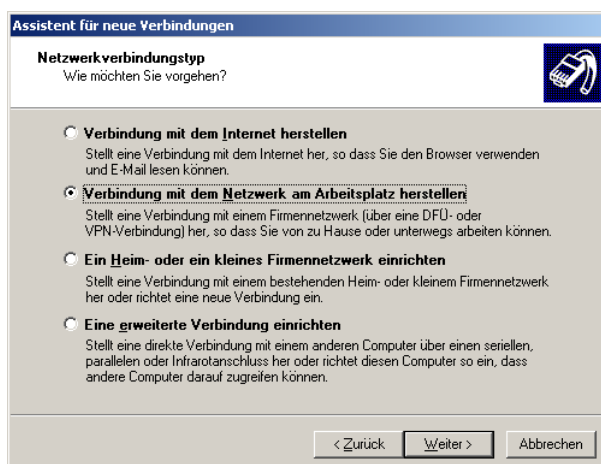


Abbildung 32: Wählen Sie "Verbindung mit dem Netzwerk am Arbeitsplatz herstellen"

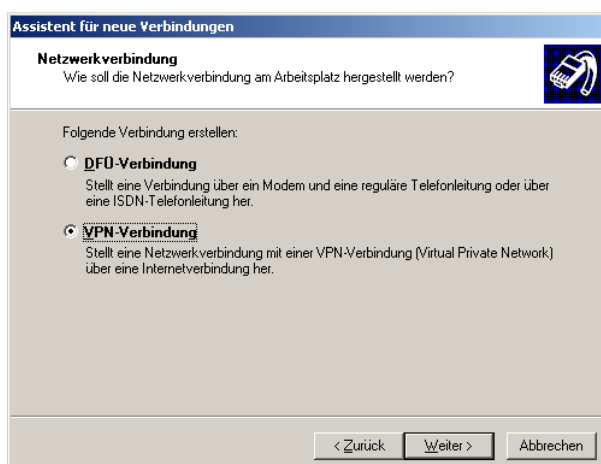


Abbildung 33: Wählen Sie "VPN-Verbindung"

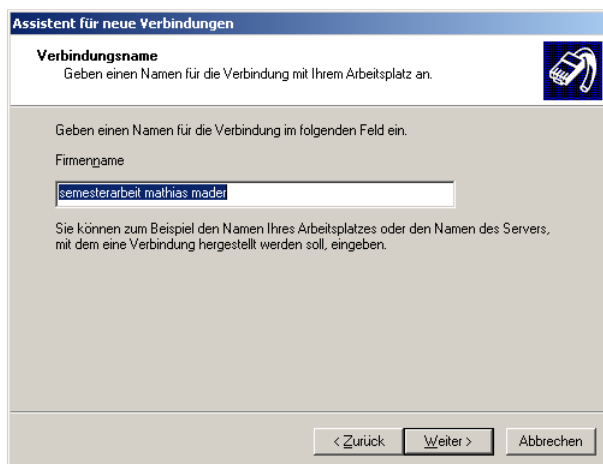


Abbildung 34: Geben Sie der Verbindung eine Bezeichnung Ihrer Wahl

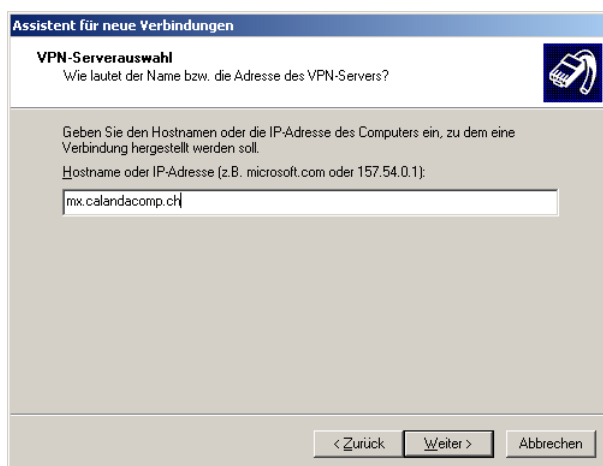


Abbildung 35: Als Hostname geben Sie ein "mx.calandacomp.ch"



Abbildung 36: Klicken Sie auf "Fertig stellen"



Abbildung 37: Geben Sie hier die Zugangsdaten gemäss Tabelle 09 ein und klicken Sie auf „Eigenschaften“

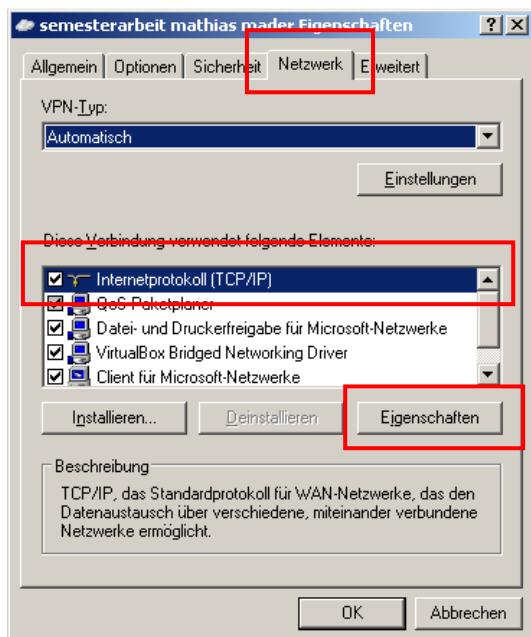


Abbildung 38: Klicken Sie im Register "Netzwerk" auf "Internetprotokoll (TCP/IP)" und anschliessend auf den Button "Eigenschaften"

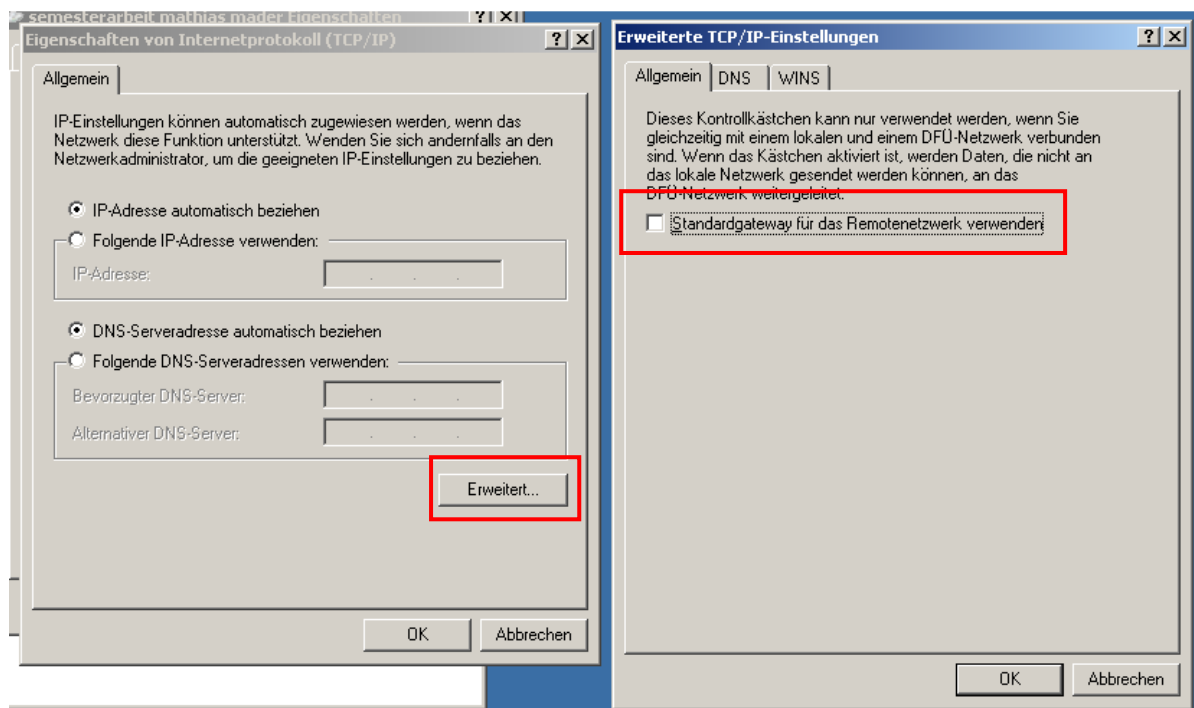


Abbildung 39: In den Eigenschaften klicken Sie auf "Erweitert" und entfernen Sie im nachfolgenden Dialog den Haken bei "Standardgateway für das Remotenetzwerk verwenden"

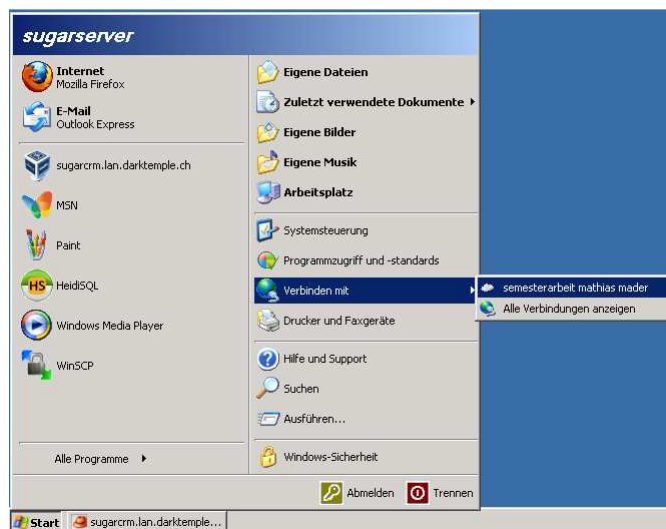


Abbildung 40: Anschliessend finden Sie Ihre neue VPN Verbindung in den Netzwerkverbindungen

## 7 Hinweise und Unterschrift

Hiermit erkläre ich, die vorliegende Semesterarbeit selbstständig und nur mit Hilfe der angegebenen Quellen erstellt zu haben.

Ort, Datum:

Unterschrift:

---

---

