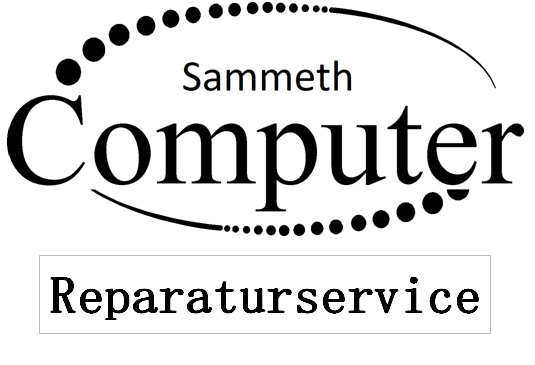
Projektdokumentation Computer Reparatur Service

Noname

IK11/G2

16.05.2013



Inhaltsverzeichnis

1. Erläuterung des Projektauftrages ......................................................................... 2

1.1 Projektumfeld ........................................................................................................ 2

1.2 Aufgabenstellung .................................................................................................. 2-3

2. Erläuterung des Projektauftrages

2.1 Problemanalyse .................................................................................................... 4

2.1.1 Ist-Analyse ............................................................................................................ 4

2.1.2 Soll-Konzept .......................................................................................................... 4

3. Pflichtenheft .......................................................................................................... 5

3.1 Zielbestimmungen ................................................................................................. 5

3.2 Produkteinsatz ..................................................................................................... 6

3.3 Produktumgebung ................................................................................................ 6-7

3.4 Produktfunktionen ................................................................................................ 7-8

3.5 Produktdaten ........................................................................................................ 8

3.6 Qualitätsbestimmungen ........................................................................................ 9

3.7 Testszenario ......................................................................................................... 10-12

3.8 Entwicklungsumgebung ........................................................................................ 13

4. Kosten-Nutzunganalyse ........................................................................................ 14

5. Zeitplan ................................................................................................................. 14

6. Quellcode .............................................................................................................. 15

***1. Erläuterung des Projektauftrages***

**1.1 Erläuterung des Projektauftrages**

**1.1.1 Projektumfeld**

Das Projekt wurde im Projektfeld 37 des Bereiches (IT Business Unit – ITBU) „Information Technologie & Business Services“ der Sammeth Computer Reparatur GmbH realisiert, dieses Projektfeld ist für von Computer verwaltete Geschäftslösungen verantwortlich.

Die Sammeth Computer Reparatur GmbH (SCR) ist ein mittelständisches, alleinstehendes Kleinunternehmen und bietet ihren Kunden Reparaturdienstleistungen für PC-technische Geräte aller Arten, wie zum Beispiel normale Midtower PC's, Notebooks oder Netbooks.

Die Reparaturen beinhalten Wartung bzw. Instandsetzung der Geräte, sowie Reinigung oder Austausch defekter Teile gegen eine zusätzliche Gebühr.

Um die Verwaltung und die tägliche Arbeit zu vereinfachen, wird bereits in der SCR intern eine alte Kundenverwaltung eingesetzt, die noch für Microsoft Windows 2000 Systeme programmiert worden ist.

Diese Kundenverwaltung, die von der New Media GmbH erstellt wurde, integriert sich vollständig in die IT-Infrastruktur. Des Weiteren bietet diese komfortable Möglichkeiten, um Stammdaten auf dieser zu bearbeiten und in eine im Unternehmen eingesetzte Access Datenbank von Microsoft zu speichern.

Die Software der New Media GmbH wurde über die letzten Jahre zwei mal erneuert. Sie erhielt unter anderem eine neue Funktion, die eine Erstellung von Kostenvoranschlagsrechnungen ermöglichte.

**1.1.2 Aufgabenstellung**

Um Kundenstammdaten besser bearbeiten zu können und die alte Software, die immer fehleranfälliger wird, auf die neuen Systeme anzupassen, soll ein Prototyp erstellt werden, der für Mitarbeiter möglichst gut zu bedienen ist, aber auch die Funktionen des alten Programms, der New Media GmbH2007 besitzt.

Damit die Anwendung auch einen Mehrwert für die SCR bietet, wurde das Projekt aufgrund folgender Situation erstellt:

Wie in vielen Software-Unternehmen, gibt es auch in der SCR eine große Anzahl an Kunden, die ihre Computer hier reparieren lassen und somit als Stammsatz hinterlegt wurden.

Diese werden in der SCR zentral gelagert in einer großen Datenbank gespeichert, und werden nicht bei den verwaltenden Mitarbeitern gespeichert.

In dieser befinden sich Informationen über alle Kunden der SCR, die eine Reparatur oder Wartung beantragt haben.

Damit eine gewisse Ordnung gewahrt bleibt, kann nur ein Mitarbeiter über das Programm auf die Datenbank zugreifen.

Leider ist die Handhabung und Pflege dieser Kundenstammsätze bisher sehr umständlich, daher soll nun eine einfachere Lösung geschaffen werden, um die zentralen Kundenstammsätze auf der Datenbank zu bearbeiten.

Da eine Migration vom alten Windows 2000 auf die neue Windows 7 Version geplant ist, soll die Anwendung auf Basis von Microsoft Visual Basic.net 2010 erstellt werden.

Das Programm soll zudem zusätzlich folgende Kriterien erfüllen:

* Das Programm läuft unter Visual Studio 2010.
* Übersichtliche Gestaltung für den Anwender.
* Die Grunddaten für das Programm befinden sich in ACCESS.
* Suchfunktion nach Datensätzen anhand eines Schlüssels (z. B. Artikelnummer). Überprüfung der Richtigkeit des eingegebenen Schlüssels. Anwenderfreundliche Gestaltung der Korrektur der Eingabe.
* Erfassung von neuen Datensätzen über eine Userform.
* Überprüfung der vorhandenen Daten auf Duplikate.
* Veränderung und Löschen der vorhandenen Datensätze.
* Verwenden Sie verschiedene Arten von Steuerelementen.
* Erstellen Sie mindestens 3 verschiedene Userforms zu ihrem Projekt.
* Das Projekt muss eine zeitliche Betrachtung von Doppelbelegungen/Ausleihen oder ähnliches beinhalten.

***2. Erläuterung des Projektauftrages***

**2.1 Problemanalyse**

**2.1.1 Ist-Analyse**

Die alte Software ist nur sehr umständlich zu bedienen, da die Navigation und Seitengestaltung sehr komplex und unübersichtlich gestaltet ist.

Das Hinzufügen erweist sich als ebenfalls sehr schwierig, da die Verbindung mit der Datenbank erst durch Eingabe zwei zusätzlicher Passwörter hergestellt werden kann.

Zudem benötigt der Verbindungsaufbau für heutige Verhältnisse viel zu lang. Dies ist vor allem auf den hohen Speicherressourcen-Verbrauch des Programmes zurückzuführen.

Ein zusätzliches Problem, dieser Software, stellt auch die Stabilität und somit ebenfalls die Sicherheit dar. Das Programm erleidet häufig einen Absturz auf den neuen PC-Systemen, wodurch eingegebene Daten, die nicht rechtzeitig gespeichert wurden, unwiederbringlich verloren sind.

**2.1.2 Sollkonzept**

Es soll eine einfach zu bedienende und schnelle Clientanwendung erstellt werden, welche einen guten Überblick über die Kundenstammdatensätze geben soll.

Die kompletten Daten werden auch weiterhin in eine Datenbank von Microsoft Access gespeichert.

Um den Mitarbeitern nun eine bedienfreundlichere grafische Benutzeroberfläche zu bieten, sollen die einzelnen Programmteile möglichst einfach strukturiert, aber gleichzeitig schön übersichtlich sein.

Durch eine einfachere Verwaltungsstruktur in der Software, sollen nun somit nicht nur Eingaben schneller getätigt werden können, sondern auch das Switchen zwischen den vielen Fenstern verringert werden.

Im Pflichtenheft (nachfolgend) werden unter anderem diese Angaben konkretisiert.

***3. Pflichtenheft***

**3.1 Zielbestimmungen**

Sammeth Computer Reparatur stellt ein Programm dar, der die Verwaltung von Kundenstammsätzen und Erstellung von Kostenvoranschlägen ermöglichen soll.

Ziel des Programmes Sammeth Computer Reparatur, ist es, das alte Kundenverwaltungsprogramm der Firma Sammeth Computer Reperatur zu ersetzen und dessen Funktionen auf die neuen Computer Systeme portieren soll.

Dieses Programm ist ein Prototyp und besitzt somit nur einen Teil der Funktionen.

**3.1.1 Musskriterien**

* Der Benutzer muss sich mit seiner Benutzerkennung und seinem Passwort anmelden, um Daten mit Hilfe des Programms verwalten zu können.
* Es soll für die Kundenstammsätze eine eigene Verwaltung, mit anlegen, löschen, bearbeiten und suchen existieren, in der alle relevanten Informationen eingetragen und anschließend gespeichert werden können.
* Der Benutzer soll sämtliche Datensätze einsehen und diese möglichst komfortabel verarbeiten können. Die Möglichkeit zur Suche von Datensätzen anhand von Wörtern soll ebenfalls bestehen.
* Die Grunddaten sollen in einer Microsoft Access Datenbank gespeichert werden.
* Das Programm muss einen sinnvollen, strukturierten und leicht übersichtlichen Quellcode besitzen, um bei späteren Änderungen wieder einen schnellen Überblick über die Funktionen zu bekommen.
* Die Benutzeroberfläche muss so benutzerfreundlich wie möglich gestaltet werden, sodass diese sich, selbst ohne große programmiertechnischen Kenntnisse, in diesem Programm schnell zurecht finden.

**3.1.2 Wunschkriterien**

* Eine zusätzliche Benutzerkontenverwaltung, die

- dem Administrator erlaubt, die Benutzerkonten einzusehen und zu verwalten zuzuweisen.

- dem Benutzer erlaubt sein Passwort oder Benutzernamen zu ändern.

* Erweiterung des Programmes um weitere Funktionen.

**3.1.3 Abgrenzungskriterien**

* zum jetzigen Versionsstand, müssen Benutzer ihr Kennwort vom Administrator erfragen
* die Kostenvoranschlagserstellung, ist nur auf das geringste ausgelegt
* das Programm soll nicht kompiliert werden

**3.2 Produkteinsatz**

**3.2.1 Anwendungsbereiche**

Sammeth Computer Reparatur verwendet dieses Programm um Kundenstammsätze zu verwalten, aber auch um Kostenvoranschläge zu erstellen.

Diese Anwendung ist im kompletten Unternehmen lauffähig. Es kann aber auch in anderen Sparten des IT-Betriebes ausgeführt werden.

Das Programm soll außerdem

* auf allen Windows Rechnern laufen können
* und ist für Personen gedacht, die erste Erfahrungen mit Software haben und mittleres Wissen in Betriebsverwaltungsabläufen besitzen

**3.2.2 Zielgruppen**

Dieses Programm wurde für die Mitarbeiter des Unternehmens Sammeth Computer Reparatur entwickelt. Es werden hierfür normale bis gute Sprachkenntnisse in Deutsch vorausgesetzt, da dieses Programm überwiegend in Deutsch gestaltet wurde.

**3.2.3 Betriebsbedingungen**

Dieses Programm soll sich bezüglich der Betriebsbedingungen nicht wesentlich von anderen Programmen unterscheiden.

* Betriebsdauer: täglich, 24 Stunden
* Fehlerresistent
* wartungsfrei
* selbsterklärend und benutzerfreundlich

**3.3 Produktumgebung**

**3.3.1 Software**

Dieses Programm ist weitgehend unabhängig vom Betriebssystem, sofern folgende Kernprogramme vorhanden sind.

- Client

* Visual Basic Bibliotheken (Net Framework 3.5 oder höher)

Es wird jedoch empfohlen dieses Programm mit Microsoft Windows zu verwenden, da die Optimierung auf dieses Betriebssystem erfolgte.

**3.3.2 Hardware**

Um die Funktionen des Programmes zu ermöglichen muss die folgende Hardware am

User – PC für eine fehlerfreie Ausführung vorhanden sein:

* + Mindestens 1000-MHZ-Prozessor oder schneller
  + Mindestens 512 MB RAM
  + Mindestens 20 MB Speicherplatz auf der Festplatte
  + Tastatur und Microsoft Mouse oder ein kompatibles Steuergerät
  + Videoadapter und Monitor mit mindestens 800 x 600 oder höherer Auflösung

**3.4 Produktfunktionen**

**3.4.1 Benutzerfunktionen**

**Anmelden:**/T0010/

Ein registrierter Benutzer kann sich über die Start- bzw. Login-Seite des Systems schnell und bequem anmelden (Login). Dazu ist seine Kennung erforderlich:

* Benutzername
* Passwort

**3.4.2 Eingabemöglichkeiten des Benutzers**

**Kunde anlegen:**/T0020/

Alle Informationen zu einem Kunden, die beim Anlegen eingegeben werden können:

* Kundennummer
* Vorname
* Nachname
* Straße
* Ort
* Postleitzahl (PLZ)
* Kontonummer

**Kunde bearbeiten:**/T0030/

Alle Informationen zu einem Kunden werden in einem zusätzlichem Fenster angezeigt, der zuvor ausgewählt wurde. Folgende Daten können verändert werden:

* Vorname
* Nachname
* Straße
* Ort
* Postleitzahl (PLZ)
* Kontonummer

**Kunde löschen:**/T0040/

Alle Informationen zu einem Kunden werden gelöscht. Dies betrifft folgende Daten:

* Kundennummer
* Vorname
* Nachname
* Straße
* Ort
* Postleitzahl (PLZ)
* Kontonummer
* Registrierdatum des Kunden

**Kunde suchen:**/T0050/

Informationen können mit Hilfe der Suchmaske sowohl über "Kunde suchen", als auch "Computer suchen" gesucht werden. In dieser Rohfassung müssen die Suchkriterien noch richtig geschrieben werden, auch mit Groß- und Kleinschreibung, um erfolgreich ein Ergebnis angezeigt zu bekommen. Dies wird in der späteren Version behoben.

**3.5 Produktdaten**

**3.5.1 Kundendaten D01**

Die über den Kunden eingetragene Informationen werden in der Access-Datenbank gespeichert. Folgende Informationen werden gespeichert:

* Kundennummer
* Vorname
* Nachname
* Straße
* Ort
* Postleitzahl (PLZ)
* Kontonummer
* Registrierungsdatum

**3.5.3 Benutzerdaten D03**

Die über den Benutzer eingetragene Informationen werden ebenfalls in der Access-Datenbank gespeichert. Folgende Informationen sind vom Benutzer gespeichert:

* Benutzername
* Passwort

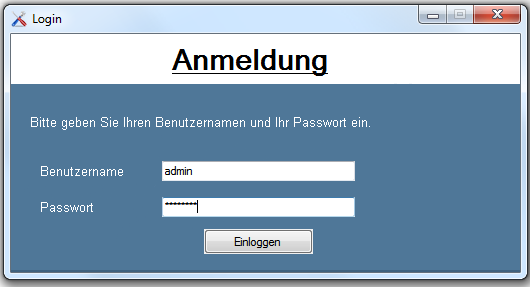
**3.6 Qualitätsbestimmungen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | sehr wichtig | wichtig | weniger wichtig | unwichtig |
| Funktionalität | X |  |  |  |
| Richtigkeit |  | X |  |  |
| Ordnungsmäßig |  | X |  |  |
| Zuverlässigkeit | X |  |  |  |
| Reife |  |  | X |  |
| Fehlertoleranz |  | X |  |  |
| Wiederherstellbarkeit |  |  | X |  |
| Benutzbarkeit | X |  |  |  |
| Verständlichkeit | X |  |  |  |
| Erlernbarkeit |  |  |  | X |
| Bedienbarkeit |  | X |  |  |
| Effizienz |  |  | X |  |
| Zeitverhalten |  |  |  | X |
| Verbrauchsverhalten |  | X |  |  |
| Änderbarkeit | X |  |  |  |
| Modifizierbarkeit |  | X |  |  |
| Stabilität |  |  | X |  |
| Prüfbarkeit |  |  | X |  |
| Übertragbarkeit |  | X |  |  |
| Anpassbarkeit |  |  | X |  |
| Installierbarkeit |  |  |  | X |

**3.7 Testszenario**

Für folgendes Testszenario können folgende Daten zum Einloggen in das Programm verwendet werden:

|  |  |  |
| --- | --- | --- |
| Benutzername: | admin | Max |
| Passwort: | password | Mustermann |



Max Mustermann möchte sich in das Programm Sammeth Computer Reparatur Service einloggen. Dazu tippt er seinen Benutzernamen und sein Passwort in die, in der obigen Abbildung zu sehenden, Textboxen ein. Klickt er nun auf den Button "Einloggen", so gelangt er in das Hauptmenü / Übersicht des Programmes.



Max Mustermann hat nun zwei Auswahlmöglichkeiten:

Drückt er den Button "Kundenstamm verwalten", so gelangt er in die Kundenstamm verwalten Übersicht. In dieser sieht er eine Liste aller in der Datenbank enthaltenen Kundenstammsätze.

Hier kann Max Mustermann nun einen Kunden suchen, indem er seinen gewünschten Suchbegriff in die Textbox mit dem Namen "Suchbegriff eingeben" eingibt. Mit einem Klick auf den Button "Suchen" werden ihm alle passenden Treffer ausgegeben.

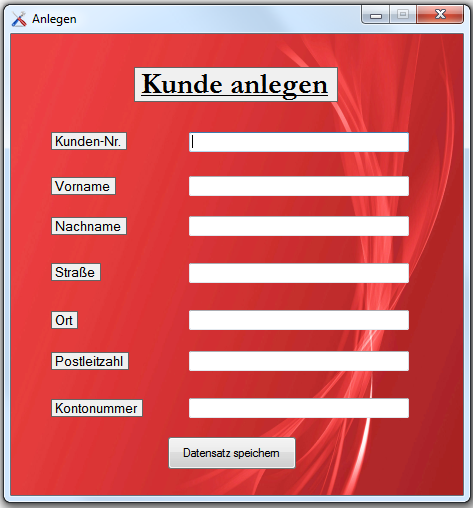
Mit einem Klick auf Aktualisieren, kann Max wieder alle Datensätze anzeigen lassen, aber auch die Liste aktualisieren.

Max Mustermann hat in diesem Fenster aber auch noch andere Möglichkeiten:

Wählt er in der Kundenstammsatzliste einen Datensatz aus, so aktivieren sich die zwei Buttons "Kunde löschen" und "Kunde bearbeiten". Mit einem Klick auf Kunde löschen wird der Datensatz, nach Bestätigung eines Sicherheitsabfrage, unwiederbringlich aus der Datenbank gelöscht.



Drückt Max Mustermann hingegen den "Kunde bearbeiten" Button, so gelangt er in das Fenster "Kunde bearbeiten". Hier kann er die angelegten Kundeninformationen des im "Kundenstamm verwalten" ausgewählten Datensatzes verändern. Ändert er den Datensatz jedoch so ab, dass dieser sich mit dem Vornamen, Nachnamen und Kontonummer eines anderen in der Datenbank angelegten Kunden überschneidet (beide Datensätze sind gleich), so wird eine Fehlermeldung ausgegeben.



Klickt er den Button "Kunde anlegen" an, so gelangt er in das Eingabefenster "Kunde anlegen". Hier können nun Kundeninformationen in eingegeben und mit einem Mausklick auf den Button "Datensatz speichern" in die Datenbank übertragen werden.

**3.8 Entwicklungsumgebung**

3.7.1 Software:

Als Entwicklungsumgebung und kam folgende Software zum Einsatz:

* Microsoft Visual Basic.NET 2010

Programme, die zur Erstellung der Software benutzt wurden:

* Microsoft Access 2010
* Microsoft Office Word 2010
* Microsoft Paint
* Waterfox (Internet Browser)

3.7.2 Hardware:

* + Rechner/Computer

3.7.3 Orgware:

* Hinweise des Lehrers
* Klassenkameraden

***4. Kosten-Nutzenanalyse***

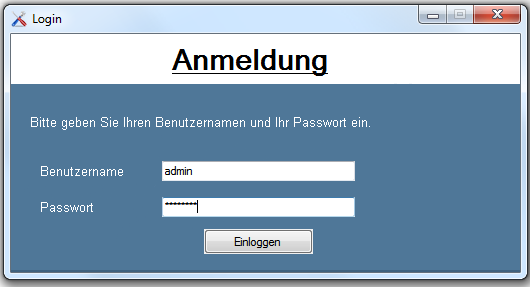
|  |  |  |  |
| --- | --- | --- | --- |
| Enstehende Kosten | | | |
| Stundenlohn |  |  | 25,00 € |
| Schulungen (pro Mitarbeiter) | | | 10,00 € |
| Integrationskosten | |  | 15,00 € |
| Kosten gesamt | |  | 50,00 € |
| Nutzen der Software | | | |
| Motivation |  |  | 230,00 € |
| Produktivitätssteigerung | |  | 125,00 € |
| Nutzen gesamt | |  | 355,00 € |
| Saldo aus Kosten und Nutzen | | |  |
| im Monat |  |  | 305,00 € |

***5. Zeitplan***

|  |  |  |
| --- | --- | --- |
| Nr. | Vorgangsname | Dauer |
| 1. | Planungsphase | 5 Stunden |
| 2. | Festlegung eines Projektes |  |
| 2.1. | Festlegung einer Programmstruktur | 8 Stunden |
| 3. | Realisierungsphase |  |
| 4. | Erstellen der Anwendung | 23 Stunden |
| 5. | Erstellung der Dokumentation | 5 Stunden |
| 6. | Erstellung des Pflichtenheftes | 10 Stunden |
| 7. | Gesamtbearbeitungszeit | ca. 51 Stunden |

***6. Quellcode***

**Login**



Imports System.Data.OleDb

Public Class login

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Dim ausgabe As Boolean

con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data Source=Datenbank Computer Verleih.accdb;"

cmd.Connection = con

cmd.CommandText = "select \* from login"

Try

'Datenbank öffnen

con.Open()

reader = cmd.ExecuteReader()

'Wenn der Benutzername mit Passwort nicht übereinstimmt "ausgabe" auf false setzen

Do While reader.Read()

If TextBox1.Text <> reader("Benutzername") And TextBox2.Text <> reader("Passwort") Then

ausgabe = False

'Wenn der Benutzername mit Passwort übereinstimmt "ausgabe" auf true setzen

ElseIf TextBox1.Text = reader("Benutzername") And TextBox2.Text = reader("Passwort") Then

MsgBox("Login erfolgreich")

ausgabe = True

hauptfenster.Show()

Me.Hide()

Exit Do

End If

Loop

'Hat die Boolean "ausgabe" den Wert false, so wird die folgende Meldung ausgegeben

If ausgabe = False Then

MsgBox("Benutzername oder Passwort falsch!")

End If

'Leeren der Textboxen, für erneute Eingabe

TextBox1.Text = ""

TextBox2.Text = ""

reader.Close()

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub Login\_FormClosed(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosedEventArgs) Handles MyBase.FormClosed

Application.Exit()

End Sub

End Class

**Hauptmenü / Übersicht**



Public Class hauptfenster

'Entfernen des Exit-Buttons um Nutzer vom unvorsichtigem Beenden des Programmes abzuhalten

'Quelle: www.vbforums.com/showthread.php?670755-Disabling-Exit-%28-X-%29-Button- in-a-form

Protected Overrides ReadOnly Property CreateParams() As CreateParams

Get

Dim cp As CreateParams = MyBase.CreateParams

Const CS\_NOCLOSE As Integer = &H200

cp.ClassStyle = cp.ClassStyle Or CS\_NOCLOSE

Return cp

End Get

End Property

Private Sub btn\_Kunde\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_Kunde.Click

kundenstamm\_verw.Show()

Me.Close()

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Close()

login.Show()

End Sub

Private Sub kostenvor\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles kostenvor.Click

End Sub

Private Sub btn\_logout\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_logout.Click

Me.Close()

login.Show()

End Sub

End Class

**Kundenstamm verwalten**



Imports System.Data.OleDb

Public Class kundenstamm\_verw

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Private Sub Kundenstamm\_FormClosing(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

hauptfenster.Show()

Me.Hide()

End Sub

Private Sub btn\_anlegen\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_anlegen.Click

Me.Hide()

kunde\_anlegen.Show()

End Sub

Private Sub txtbox\_sucheing\_MouseClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles txtbox\_sucheing.MouseClick

txtbox\_sucheing.Text = ""

End Sub

Private Sub btn\_akt\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_akt.Click

Anzeigen()

'Deaktivierung der Löschen und Bearbeiten Buttons, um Benutzer dazu zu bewegen einen Datensatz auszuwählen

btn\_bearbeiten.Enabled = False

btn\_löschen.Enabled = False

End Sub

Private Sub kundenstamm\_verw\_Activated(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Activated

Anzeigen()

btn\_bearbeiten.Enabled = False

btn\_löschen.Enabled = False

End Sub

Private Sub Anzeigen()

Dim counter As Integer

listv\_kunden.Items.Clear()

counter = 0

con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data Source=Datenbank Computer Verleih.accdb;"

cmd.Connection = con

cmd.CommandText = "select \* from Kunde;"

Try

con.Open()

reader = cmd.ExecuteReader()

'Auslesen der Datensätze aus der DB und Übertragung dieser in die Listview

Do While reader.Read()

listv\_kunden.Items.Add(reader("Kunden\_Nr"))

listv\_kunden.Items(counter).SubItems.Add(reader("Vorname"))

listv\_kunden.Items(counter).SubItems.Add(reader("Nachname"))

listv\_kunden.Items(counter).SubItems.Add(reader("Straße"))

listv\_kunden.Items(counter).SubItems.Add(reader("Ort"))

listv\_kunden.Items(counter).SubItems.Add(reader("PLZ"))

listv\_kunden.Items(counter).SubItems.Add(reader("Kontonummer"))

counter += 1

Loop

reader.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

End Sub

Private Sub btn\_suchen\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_suchen.Click

Dim counter As Integer

counter = 0

Try

'

con.Open()

'Suchabfragebefehl für die Datenbank

cmd.CommandText = "select \* from kunde where" & " kunden\_nr like '%" & txtbox\_sucheing.Text & "%'" & " or vorname like '%" & txtbox\_sucheing.Text & "%'" & " or nachname like '%" & txtbox\_sucheing.Text & "%'"

listv\_kunden.Items.Clear()

reader = cmd.ExecuteReader()

'Die Datenbank überprüft die vom Command String erzeugte Abfrage und gibt alle gefundenen Ergebnisse in die Listview aus

Do While reader.Read()

listv\_kunden.Items.Add(reader("Kunden\_Nr"))

listv\_kunden.Items(counter).SubItems.Add(reader("Vorname"))

listv\_kunden.Items(counter).SubItems.Add(reader("Nachname"))

listv\_kunden.Items(counter).SubItems.Add(reader("Straße"))

listv\_kunden.Items(counter).SubItems.Add(reader("Ort"))

listv\_kunden.Items(counter).SubItems.Add(reader("PLZ"))

listv\_kunden.Items(counter).SubItems.Add(reader("Kontonummer"))

counter += 1

Loop

reader.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

btn\_bearbeiten.Enabled = False

btn\_löschen.Enabled = False

End Sub

Private Sub btn\_löschen\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_löschen.Click

Dim antwort As Boolean

Dim counter As Integer

Dim text As Integer

Dim ant\_msgbox As Boolean

antwort = True

'Der Text des in der Listview angeklickten Objekts wird in eine variable übertragen

text = listv\_kunden.FocusedItem.Text

'Klickt der Nutzer kein Objekt in der Listview an, so wird folgende Meldung ausgegeben

If antwort = False Then

MsgBox("Bitte einen zu löschenden Datensatz auswählen")

Else

'Sicherheitsabfrage, ob Nutzer den Datensatz wirklich löschen will

ant\_msgbox = MsgBox("Wollen Sie diesen Datensatz wirklich löschen?", MsgBoxStyle.YesNo)

If ant\_msgbox = True Then

Try

con.Open()

cmd.CommandText = "DELETE \* FROM Kunde " & "WHERE Kunden\_Nr = " & text

counter = cmd.ExecuteNonQuery

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

Anzeigen()

End If

End If

End Sub

Public Sub btn\_bearbeiten\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_bearbeiten.Click

'Der Text des in der Listview angeklickten Objekts wird in eine Public variable übertragen

lstv\_text = listv\_kunden.FocusedItem.Text

Me.Hide()

kunde\_bearbeiten.Show()

End Sub

Private Sub listv\_kunden\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles listv\_kunden.SelectedIndexChanged

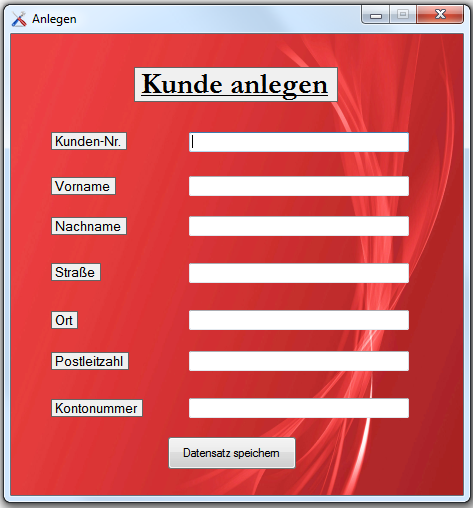
btn\_bearbeiten.Enabled = True

btn\_löschen.Enabled = True

End Sub

End Class

**Kunde anlegen**



Imports System.Data.OleDb

Public Class kunde\_anlegen

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Private Sub btn\_speichern\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_speichern.Click

Dim anzahl As Integer

Dim check1 As Boolean

Dim check2 As Boolean

Dim check3 As Boolean

Dim regdate As Date

'Überprüfungsvariablen werden angelegt

check1 = IsNumeric(txtb\_kundnr.Text)

check2 = IsNumeric(txtb\_plz.Text)

check3 = IsNumeric(txtb\_konto.Text)

regdate = Date.Today

con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data Source=Datenbank Computer Verleih.accdb;"

cmd.Connection = con

'http:// msdn.microsoft.com/en-us/library/6cd3f6w1%28v=vs.90%29.aspx

'isnumeric

If txtb\_kundnr.Text = "" Or txtb\_vorn.Text = "" Or txtb\_nachn.Text = "" Or txtb\_stras.Text = "" Or txtb\_ort.Text = "" Or txtb\_plz.Text = "" Or txtb\_konto.Text = "" Then

MsgBox("Bitte alle Felder ausfüllen!")

Else

Try

'Überprüfen der Textboxen von kunden\_nr, plz und kontonummer auf vorhanden sein eines Integer-Wertes

If check1 = True And check2 = True And check3 = True Then

doppelbelegung()

If doppel = False Then

Try

'Alle Werte der Textboxen, der Form kunde\_anlegen in die Datenbank übertragen

'Datenbank öffnen

con.Open()

'cmd.CommandText = "insert into kunde (Registrierungsdatum) values (" & regdate & ")"

cmd.CommandText = "INSERT INTO Kunde(Kunden\_Nr, Vorname, Nachname, Straße, Ort, PLZ, Kontonummer, Registrierungsdatum) VALUES ('" & txtb\_kundnr.Text & "', '" & txtb\_vorn.Text & "', '" & txtb\_nachn.Text & "', '" & txtb\_stras.Text & "', '" & txtb\_ort.Text & "', '" & txtb\_plz.Text & "', '" & txtb\_konto.Text & "', '" & regdate & "')"

MsgBox("Daten erfolgreich gespeichert")

anzahl = cmd.ExecuteNonQuery()

'Löschen der Eingegebenen Daten, falls ein weiterer Kunde angelegt werden soll

txtb\_kundnr.Text = ""

txtb\_vorn.Text = ""

txtb\_nachn.Text = ""

txtb\_stras.Text = ""

txtb\_ort.Text = ""

txtb\_plz.Text = ""

txtb\_konto.Text = ""

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

Else

MsgBox("Kunden-Nr., PLZ oder Kontonummer enthält andere Zeichen als zugelassen. Bitte für diese eine gerade Zahlenfolge eingeben.")

End If

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub doppelbelegung()

Dim checkvorname As String

Dim checknachname As String

Dim checkkonto As Integer

doppel = False

cmd.CommandText = "select \* from kunde"

Try

'Datenbank öffnen

con.Open()

reader = cmd.ExecuteReader()

'Eingegebene Werte werden auf Doppelbelegung in der Datenbank überprüft, falls Doppelbelegung vorhanden,

'so wird eine Fehlermeldung ausgegeben

Do While reader.read()

checkvorname = reader("Vorname")

checknachname = reader("Nachname")

checkkonto = reader("Kontonummer")

If checkvorname = txtb\_vorn.Text And checknachname = txtb\_nachn.Text And checkkonto = txtb\_konto.Text Then

MsgBox("Fehler! Kundendaten in dieser Konstellation bereits vorhanden!")

doppel = True

Exit Do

End If

Loop

reader.Close()

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub kunde\_anlegen\_FormClosed(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosedEventArgs) Handles MyBase.FormClosed

Me.Dispose()

kundenstamm\_verw.Show()

End Sub

End Class

**Kunde bearbeiten**



Imports System.Data.OleDb

Public Class kunde\_bearbeiten

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Private Sub kunde\_bearbeiten\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data Source=Datenbank Computer Verleih.accdb;"

cmd.Connection = con

'Suchen und Einfügen des auf der Form "Kundenstamm verwalten" gewählten Datensatzes in die Textboxen

con.Open()

cmd.CommandText = "SELECT \* FROM Kunde " & "WHERE Kunden\_Nr = " & lstv\_text.ToString

reader = cmd.ExecuteReader()

Do While reader.Read()

kunden\_nr = reader("Kunden\_Nr")

vorname = reader("Vorname")

nachname = reader("Nachname")

strasse = reader("Straße")

ort = reader("Ort")

plz = reader("PLZ")

kontonummer = reader("Kontonummer")

Loop

reader.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

txtb\_kundnr.Text = kunden\_nr

txtb\_vorn.Text = vorname

txtb\_nachn.Text = nachname

txtb\_stras.Text = strasse

txtb\_ort.Text = ort

txtb\_plz.Text = plz

txtb\_konto.Text = kontonummer

End Sub

Private Sub btn\_speichern\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_speichern.Click

Dim anzahl As Integer

doppelbelegung()

If doppel = False Then

Try

'Verbindung zur Datenbank öffnen

con.Open()

'Alle geänderten Werte der Textboxen, der Form kunde\_bearbeiten in die Datenbank übertragen

cmd.CommandText = "UPDATE Kunde SET " & "kunden\_nr = '" & txtb\_kundnr.Text & "', " & "vorname = '" & txtb\_vorn.Text & "', " &

"nachname = '" & txtb\_nachn.Text & "', " & "straße = '" & txtb\_stras.Text & "', " &

"ort = '" & txtb\_ort.Text & "', " & "plz = '" & txtb\_plz.Text & "', " &

"kontonummer = '" & txtb\_konto.Text & "' " & "where kunden\_nr = " & txtb\_kundnr.Text

MsgBox("Datensatz erfolgreich geändert")

anzahl = cmd.ExecuteNonQuery()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

kundenstamm\_verw.Show()

Me.Dispose()

End If

End Sub

Private Sub doppelbelegung()

Dim checkvorname As String

Dim checknachname As String

Dim checkkonto As Integer

doppel = False

cmd.CommandText = "select \* from kunde"

Try

'Datenbank öffnen

con.Open()

reader = cmd.ExecuteReader()

Do While reader.read()

checkvorname = reader("Vorname")

checknachname = reader("Nachname")

checkkonto = reader("Kontonummer")

If checkvorname = txtb\_vorn.Text And checknachname = txtb\_nachn.Text And checkkonto = txtb\_konto.Text Then

MsgBox("Fehler! Kundendaten in dieser Konstellation bereits vorhanden!")

doppel = True

Exit Do

End If

Loop

reader.Close()

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub kunde\_bearbeiten\_FormClosing(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

kundenstamm\_verw.Show()

Me.Dispose()

End Sub

End Class

**Kostenvoranschlag**



Imports System.Data.OleDb

Public Class kostenvoranschlag

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Private Sub btn\_weiter\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_weiter.Click

Dim ergebnis As Double

Dim check1 As Boolean

Dim check2 As Boolean

Dim check3 As Boolean

Dim check4 As Boolean

Dim reg\_zeit As Integer

Dim sysdate As Date

Dim rabatt As Double

rabatt = 0.95

check1 = IsNumeric(txtb\_repdauer.Text)

check2 = IsNumeric(txtb\_kosters.Text)

check3 = IsNumeric(txtb\_stdlohn.Text)

check4 = IsNumeric(txtb\_kdnr.Text)

sysdate = Date.Today

con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data

Source=Datenbank Computer Verleih.accdb;"

cmd.Connection = con

Try

'Datenbank öffnen

con.Open()

cmd.CommandText = "select \* from kunde"

reader = cmd.ExecuteReader()

'Überprüfung der Kundennummertextbox auf einen Eintrag, ist einer vorhanden, so

berechnet er die Tage

Do While reader.Read()

If txtb\_kdnr.Text = "" Then

MsgBox("Fehler! Keine Kundennummer eingegeben!", , "Reparatur")

Exit Do

Else

If txtb\_kdnr.Text = reader("Kunden\_Nr") Then

reg\_zeit = DateDiff(DateInterval.Day,

reader("Registrierungsdatum"), sysdate)

Exit Do

Else

MsgBox("Kunde nicht vorhanden! Bitte zuerst anlegen.")

Exit Do

End If

End If

Loop

'Ist der Kunde bereits Registriert, wird überprüft, ob dieser länger als 30 Tage angelegt ist,

'falls ja, bekommt er 5% Rabatt

If check1 = True And check2 = True And check3 = True Then

If reg\_zeit > 30 Then

ergebnis = ((txtb\_repdauer.Text \* txtb\_stdlohn.Text) +

txtb\_kosters.Text + basiskosten) \* rabatt

MsgBox("Der Kostenvoranschlag beträgt (abzüglich 5% Kundenrabatt):

" & ergebnis & " Euro")

Else

ergebnis = (txtb\_repdauer.Text \* txtb\_stdlohn.Text) +

txtb\_kosters.Text + basiskosten

MsgBox("Der Kostenvoranschlag beträgt: " & ergebnis & " Euro")

End If

Else

MsgBox("Bitte nur Zahlenbeträge eingeben.")

End If

reader.Close()

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub rad\_mid\_CheckedChanged(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles rad\_mid.CheckedChanged, rad\_lap.CheckedChanged, rad\_thin.CheckedChanged, rad\_net.CheckedChanged

If rad\_mid.Checked Then

basiskosten = 200.0

ElseIf rad\_thin.Checked Then

basiskosten = 150.0

ElseIf rad\_lap.Checked Then

basiskosten = 100.0

ElseIf rad\_net.Checked Then

basiskosten = 50.0

End If

End Sub

Private Sub kostenvoranschlag\_FormClosed(ByVal sender As System.Object, ByVal e As

System.Windows.Forms.FormClosedEventArgs) Handles MyBase.FormClosed

hauptfenster.Show()

End Sub

Private Sub RadioButton1\_CheckedChanged(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles RadioButton1.CheckedChanged, RadioButton2.CheckedChanged

If RadioButton1.Checked Then

txtb\_kosters.Enabled = False

ElseIf RadioButton2.Checked Then

txtb\_kosters.Enabled = True

End If

End Sub

End Class