**Dokumentation C# Projektaufgabe AI2010**

**Andreas Stadelmeier, Simon Herrmann**

**Software-Architektur**

In dem Projekt wurde das Softwaremuster „MVVMC“ verwendet. MVVMC ist eine Weiterentwicklung des Entwurfsmusters MVVM und erweitert dieses um eine Controller-Einheit.

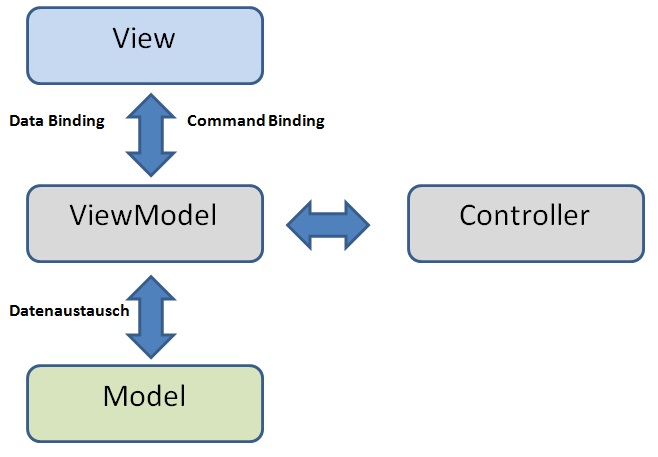


Abbildung 1: MVVMC-Modell

Abbildung 1 zeigt das MVVMC-Modell. Die View-Klassen repräsentieren hierbei die Präsentationsschicht und kommunizieren über Bindings mit dem ViewModel, welche das Interface INotifyPropertyChanged implementiert. Das ViewModel darf das Model, welches die „Daten“ der Applikation enthält. Das Model darf die View jedoch nicht kennen. Der Controller gewährleistet die Unabhängigkeit zwischen View und und ViewModel.

Folgende Klassen befinden sich in den jeweiligen Schichten:

View: MainWindowView, ActionBarView, ArtikelverwaltungView, AuftragsverwaltungView, KundenverwaltungView, MitarbeiterverwaltungView, ReportingView

ViewModel: MainWindowViewModel, ActionBarViewModel, ArtikelverwaltungViewModel, AuftragsverwaltungViewModel, KundenverwaltungViewModel, MitarbeiterverwaltungViewModel, ReportingViewModel

Model: Adress, Article, City, CompleteOrder, Country, Customer, Employee, Order, OrderLine

Controller: MainWindowController, ActionBarController, ArtikelverwaltungController, AuftragsverwaltungController, KundenverwaltungController, MitarbeiterverwaltungController, ArticleCount, Counter, ReportingController

Weitere verwendete Klassen:

Mappings:

AdressMap, ArticleMap, CityMap, CountryMap, CustomerMap, EmployeeMap, OrderLineMap, OrderMap

In den Mapping-Klassen wird eine Verbindung zwischen den Feldern in der Datenbank und den Attributen in den Model-Klassen hergestellt.

**Oberflächen-Design**

Die einzelnen Fenster (Mitarbeiterverwaltung, Artikelverwaltung, etc.) werden als Benutzersteuerelemente (WPF) implementiert. In diesen Fenstern wird jeweils nur das DataGrid und die Detailansicht, welche sich je nach verwendeten Tabellen unterscheiden, dargestellt. In dem DataGrid werden die ganzen Inhalte der Tabellen angezeigt. Neben dieser Ansicht befindet sich eine Detailansicht, welche aus Textfeldern besteht. Bei Auswahl einer Zeile werden die Textfelder mit den in der ausgewählten Zeile befindlichen Daten gefüllt. Der Anwender hat nun die Möglichkeit, diese Daten zu ändern. Die Umsetzung der Detailansicht durch Textfelder wurde entschlossen, da dies einfach, übersichtlich und leicht zu bedienen ist. Um Aktionen auf der Datenbank auszuführen, wird ebenfalls ein Benutzersteuerelement erzeugt, welches nur aus Buttons (Neu, Löschen, Speichern, etc.) besteht. Diese Buttons müssen in einem eigenen Benutzersteuerelement dargestellt werden und können nicht ins Hauptfenster übernommen werden, da die einzelnen Buttons je nach gewählten Menüpunkt aktiv oder inaktiv sind. Das Hauptfenster verfügt über ein Auswahlmenü (Buttons), in welchem ausgewählt werden kann, welches Menü (Artikelverwaltung, Kundenverwaltung, etc.) angezeigt werden soll. Die einzelnen Benutzersteuerelemente können dann in diesem Hauptfenster per ContentControl angezeigt werden. Dies hat den Zweck, dass die einzelnen Fenster, welche unterschiedlich aufgebaut sind in ein einzelnes Hauptfenster geladen werden können.

Abbildung 2 zeigt die gesamte Oberfläche am Bespiel Mitarbeiterverwaltung

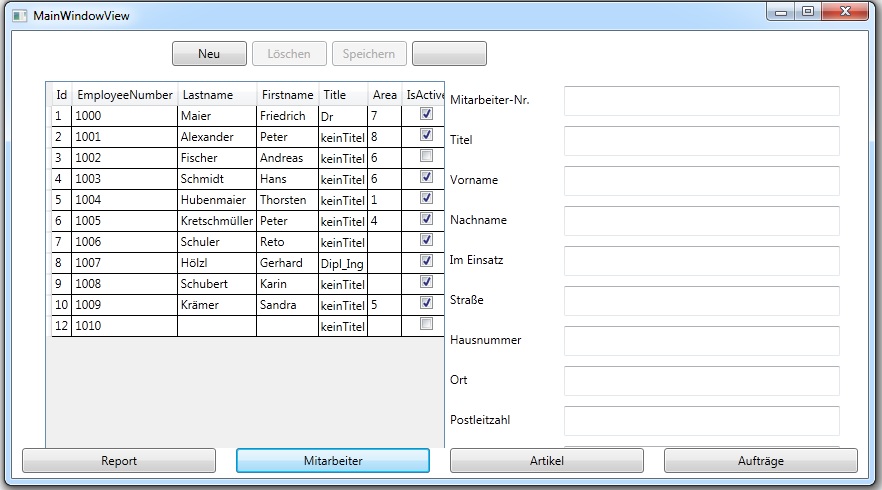


Abbildung : Oberfläche der Anwendung