

Projektdokumentation und – reflexion zum Projekt: Cocktail Almanac

Im Rahmen des Schulfach: Projekt Praktikum

Von den Schülern Aherne Rory und Eppensteiner Rene 3cInft

Inhaltsverzeichnis

| | |
|---------------------------------------------|----|
| Einleitung..... | 3 |
| Beschreibung des Projekts | 3 |
| Die Tools..... | 4 |
| Microsoft Visual Studio: | 4 |
| ASP.NET: | 7 |
| ASP.NET MVC Framework: | 11 |
| Microsoft SQL Server:..... | 12 |
| Persönlicher Ausblick auf das Projekt | 15 |

Einleitung

Schon im zweiten Lehrgang der TFBS EKE wurde uns gesagt, dass wir im dritten Lehrgang ein Projekt realisieren müssen. Aus diesem Grund haben Herr Aherne und Ich schon im Vorfeld Gedanken über das Thema gemacht. Nach viel Brainstorming haben wir und dann entschieden, ein Programmiertechnisches Projekt, mit ASP.NET und Visual Studio für das Fach Projekt Praktikum zu nehmen.

Wir haben drei Stunden PP in der Woche und neun Wochen Schule, was insgesamt 27 Stunden pro Person ergibt, was für ein so umfangreiches Projekt zwar recht knapp bemessen ist, aber durch viel Engagement dennoch erreichbar ist.

Beschreibung des Projekts

Das Cocktail Almanac Projekt wird mit den Tools ASP.NET und Visual Studio 2015 Community und Microsoft SQL als Datenbank ausgearbeitet. Erklärungen zu den Tools werden im Anschluss noch erläutert. Aber vorweg kann man schon mal sagen das es eine Web Applikation mit den Sprachen HTML und C# ist.

Da wir in dem Fach Informatik und Programmierlabor mit C# und Microsoft SQL Server arbeiten, ist es natürlich sehr mit diesem Fach verzahnt.

Um die wenigen Stunden die wir haben, so gut als möglich zu nutzen, versuchen wir natürlich außerhalb der Schule so viel Knowhow wie möglich zu erlangen um dieses so gut als möglich innerhalb der Stunden einzusetzen.

Die Tools

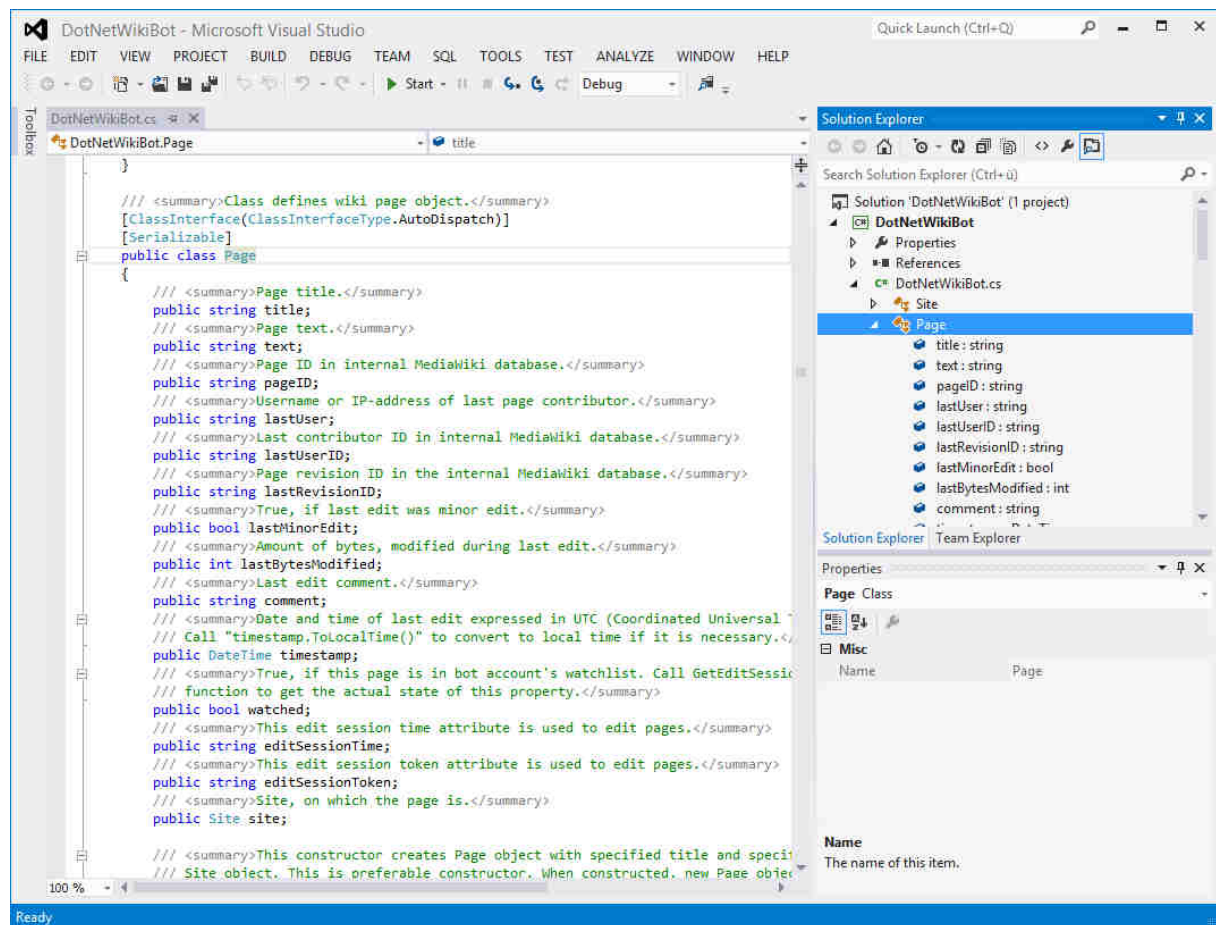
Microsoft Visual Studioⁱ:



Visual Studio ist eine von dem Unternehmen Microsoft angebotene, integrierte Entwicklungsumgebung für verschiedene Hochsprachen. Die aktuelle Version 14.0 („Visual Studio 2015“) unterstützt Visual Basic .NET, C, C++, C++/CLI, C++/CX, C#, F#, SQL Server, TypeScript und Python, sowie HTML, JavaScript und CSS für die Entwicklung von Webanwendungen und Windows Store-Apps.

Visual Studio ermöglicht es dem Programmierer, sowohl native Win32/Win64-Programme als auch Anwendungen für das .NET-Framework zu entwickeln. Darüber hinaus lassen sich mit Visual Studio Windows-Apps, dynamische Webseiten bzw. Webservices für das Internet/Intranet oder Azure-Services entwickeln. Ein weiterer Schwerpunkt liegt in der Entwicklung von mobilen Apps für Windows Phone, Android und iOS mit durch Xamarin bereitgestellten Tools.

Die Oberfläche:



Für die Version 2015 gab es eine Reihe von CTP (Community Technology Previews) Releases und eine Visual Studio 2015 Preview vom November 2014. Ausgeliefert wurde Visual Studio 2015 am 20. Juli 2015.

Neuerungen sind für die .NET-Entwicklung die Umstellung auf die neue Roslyn-Compilerplattform, die Unterstützung für Themen wie ASP.NET und Web Development vNext oder .NET Native für Apps. Eine neue Ansicht "Diagnosetools" zeigt CPU und Speicherauslastung der Anwendung an.

Die IDE bietet erstmals die Möglichkeit, Ansichten (also die Fensteraufteilung) explizit zu speichern, anzuwenden oder zu verwalten. Dabei gehört wie bisher zu unterschiedlichen Betriebszuständen (wie Debugging und Code Bearbeitung) eine dem Modus angepasste Ansicht. Im Bereich C++ gab es Erweiterungen. Darüber hinaus wird die C++-Entwicklung für Android durch einen "Visual Studio Emulator für Android" sowie für iOS unterstützt.

In Visual Studio 2015 wurde die Unterstützung für das .NET Framework 4.6 und das Kinect für Windows 2.0 SDK hinzugefügt.

Funktion:

Visual Studio gilt als komfortable Entwicklungsumgebung mit vielen Funktionen. Der Editor unterstützt z. B. eine von der Cursorposition abhängige Online-Hilfe, das Ein- und Ausblenden von Codeblöcken, farbliche Hervorhebung von Schlüsselwörtern, automatische Syntaxprüfung und IntelliSense, die automatische Methoden- und Funktionsergänzung während der Quelltext-Eingabe. Des Weiteren bietet die Entwicklungsumgebung grafische Schnittstellen zur Einbindung von Web Services, ActiveX- und .NET-Bibliotheken, einen „Server-Explorer“ zum Zugriff auf Datenquellen wie z. B. Microsoft SQL Server, den Windows-Nachrichtendienst oder WMI, sowie WYSIWYG-Editoren zur Entwicklung der Benutzeroberfläche von Windows- und Web-Anwendungen, zur Bearbeitung von Icons und von XML-Dokumenten.

Seit Visual Studio .NET (2002) erweiterte sich der Umfang der unterstützten Sprachen stark. Hierbei wirkt sich vor allem ein Zertifizierungsprogramm von Microsoft aus, das es Drittanbietern ermöglicht, eigene Sprachen als Ergänzungen zum Visual Studio anzubieten. Beispiele sind Delphi, Eiffel, F# oder Prolog. In Visual Studio 2008 wurde die Unterstützung für Microsofts Java-Variante Visual J# aufgegeben.

Visual Studio enthält einen integrierten Debugger. Dieser enthält die "Bearbeiten und Fortfahren"-Funktion und erlaubt das nachträgliche Anhängen an bereits laufende Prozesse, sowohl am lokalen Rechner als auch über das Netzwerk.

Für die Entwicklung von ASP.NET-Anwendungen enthält Visual Studio seit Version 2005 einen integrierten Webserver, sodass keine eigene Internet Information Services-Installation mehr notwendig ist. Ebenfalls neu in dieser Version sind Funktionen für das Refactoring, die bisher durch Zusatzprodukte ("Add-Ins") wie ReSharper oder Refactor! Pro nachgerüstet werden mussten.

ASP.NETⁱⁱ:



ASP.NET (Active Server Pages .NET) ist ein serverseitiges Verfahren von Microsoft zum Erstellen dynamischer Webseiten, Webanwendungen und Webservices auf Basis des Microsoft-.NET-Frameworks. Seit 2002 ist ASP.NET der Nachfolger von ASP (Active Server Pages).

ASP.NET kommt auf ca. 17,2 % aller Websites als serverseitige Programmiersprache zum Einsatz und liegt damit nach PHP (82 %) und vor dem drittplatzierten Java (2,8 %) auf dem zweiten Platz der am häufigsten serverseitig verwendeten Sprachen zum Erstellen von Webseiten.

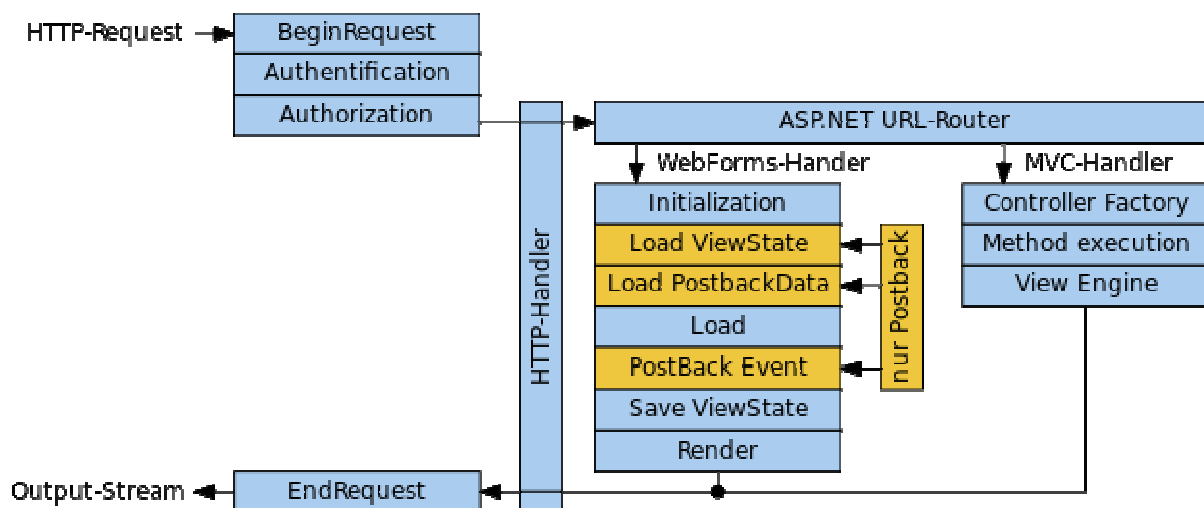
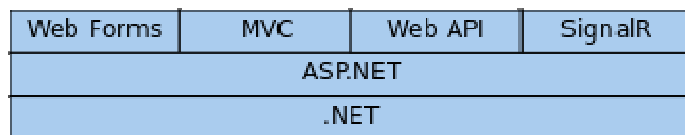
Aufbau von ASP.NET:

ASP.NET baut auf dem .NET-Framework auf und stellt eine Methode dar, um .NET-Anwendungen in IIS oder anderen mit ASP.NET kompatiblen Webservern zu verwalten („hosting“). ASP.NET stellt hierbei Klassen zur Verfügung, die allen Webanwendungen gemeinsam sind, wie etwa die Authentifizierung und Autorisierung von Benutzern. Das wichtigste Element ist hierbei die HTTP-Laufzeitumgebung, die HTTP-Anfragen bearbeitet.

Mit ASP.NET können Webanwendungen in von .NET unterstützten Sprachen (theoretisch alle CLR-kompatiblen Sprachen) erstellt werden, gebräuchlich sind nahezu ausschließlich objektorientierte Sprachen wie C# und VB.NET. Eingeschränkt wird die Sprachwahl meistens nur von der jeweiligen Entwicklungsumgebung. ASP.NET ist somit keine

Programmiersprache, sondern eine Bündelung von Techniken. ASP.NET-Sprachen werden nicht interpretiert, sondern in CIL übersetzt und kompiliert.

Auf die ASP.NET-Plattform setzen die jeweiligen Verfahren, wie ASP.NET WebForms, ASP.NET AJAX, ASP.NET MVC, ASP.NET Dynamic Data, ASP.NET WebAPI und ASP.NET SignalR auf, welche jeweils als HTTP-Handler implementiert sind.



ASP.NET Web Forms:

Die Web Forms bilden den Grundbaustein der Entwicklung von Webinhalten mit ASP.NET in seiner klassischen Form. Web Forms sind in Dateien mit der Dateiendung .aspx enthalten. Die ASPX-Dateien enthalten typischerweise statisches (X)HTML Markup sowie die „Web Controls“ und „User Controls“, die gemäß der serverseitig hinterlegten Logik auf dem Webserver verarbeitet und als dynamischer (X)HTML-Code ausgegeben werden. Beispiele für Web Controls, die Microsoft standardmäßig mitliefert, sind ein Kalendercontrol oder die Gridview (Tabellenansicht von Daten).

Zusätzlich können dynamische Code-Inhalte, die vom Server ausgeführt werden sollen, innerhalb einer ASPX-Datei in der Form `<% -- dynamic code -- %>` hinterlegt werden. Diese Art der Einbettung ähnelt stark den Konstrukten anderer Web-Programmiersprachen wie PHP, JSP und ASP. Allerdings sollte der Entwickler möglichst auf im HTML-Code verschachtelten Programmcode verzichten, da der Compiler diesen Code erst bei

Ausführung auf dem Webserver überprüft. Der Code separat verfasster „Code-Behind“-Dateien (siehe unten) wird dagegen vom Compiler vollständig überprüft, was eine höhere Laufzeitsicherheit erzielt.

Mit dem ASP.NET Framework 2.0 veröffentlichte Microsoft den neuartigen Ansatz des „Code-Behind“-Modells. Dabei verbleiben statische Inhalte in der ASPX-Datei, während der dynamische Code in eine Datei des Typs .aspx.vb oder .aspx.cs respektive .aspx.fs ausgelagert wird. Der Dateityp ist dabei von der verwendeten .NET-Sprache abhängig. Entwickler, die nicht mit dem „Code-Behind“-Ansatz arbeiten möchten, können die Funktion `Render` überschreiben und sich hier (vollkommen unabhängig von der Design-Ansicht) HTML selbst erzeugen.

Code-Behind-Modell:

Microsoft empfiehlt die Nutzung des „Code-Behind“-Ansatzes für die dynamischen Code-Anteile. Die „Code-Behind“-Dateien tragen üblicherweise die Dateiendung .aspx.cs oder .aspx.vb. Der Dateiname der ASPX-Datei entspricht dabei stets genau dem Namen der „Code-Behind“-Datei, unterschiedlich sind nur die Dateinamensendungen. Microsoft Visual Studio und andere IDEs geben diese Art der Entwicklung standardmäßig vor. Der Code einer solchen „Code-Behind“-Datei wird bereits vor Aufruf der entsprechenden Seite kompiliert. Dies senkt die Fehlerrate gegenüber Interpretersprachen, bei denen die Überprüfung auf Fehler erst zur Laufzeit stattfindet. In der „Code-Behind“-Datei hat der Entwickler die Möglichkeit, auf unterschiedlichste Ereignisse („events“) im Lebenszyklus einer ASP-Web-Forms-Web-Seite zu reagieren, darunter nicht nur z. B. das Laden der Webseite, sondern auch Ereignisse, die sich aus Benutzerinteraktionen mit Web-Formular-Inhalten ergeben, so etwa der Klick auf eine Schaltfläche.

Das Programmmodell (mit Events, die von Controls ausgelöst werden) ist dem von .NET-Applikationen sehr ähnlich, somit ist für Entwickler, die schon Applikationen für das .NET Framework geschrieben haben, die Nutzung von ASP.NET (und C#/VB.NET) leicht möglich. Meist können für Windows-Applikationen programmierte Funktionen oder Klassen sogar ohne Änderung in ASP.NET weiterverwendet werden.

Mit diesem Ansatz unterscheidet sich ASP.NET deutlich vom klassischen ASP. Eine erste Trennung von Code zur Darstellung und den Inhalten wird dadurch erreicht. Ähnlich der Trennung zwischen „View“ und „Controller“ im MVC-Muster erlaubt diese Trennung eine bessere arbeitsteilige Entwicklung zum Beispiel durch eine Aufteilung zwischen Webdesigner, der die ASPX-Datei bearbeitet, und Programmierer, der die „Code-Behind“-Datei erstellt. Die neuere Komponente ASP.NET MVC betont konzeptionell in noch viel stärkerer Form diese Art des Separation of Concerns.

Code-Beispiel:

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="ASP.NET_Beispiel._Default"
%>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server"
ContentPlaceHolderID="MainContent">
    <source lang="XML">
        <h2>
            Welcome to ASP.NET!
        </h2>
        <p>
            To learn more about ASP.NET visit <a href="http://www.asp.net"
title="ASP.NET Website">www.asp.net</a>.
        </p>
        <p>
            You can also find <a
href="http://go.microsoft.com/fwlink/?LinkID=152368&clcid=0x409"
            title="MSDN ASP.NET Docs">documentation on ASP.NET at
MSDN</a>.
        </p>
    </source>
</asp:Content>
```

Code-Behind-Beispiel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace ASP.NET_Beispiel
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

ASP.NET MVC Frameworkⁱⁱⁱ:

Das ASP.NET MVC Framework ist eine Komponente zur Software-Entwicklung von Webanwendungen (Web Application Framework), das Paradigmen des Entwurfsmusters Model-View-Controller (MVC-Muster) befolgt.

Das vom Hersteller Microsoft entwickelte MVC-Framework basiert auf ASP.NET und ermöglicht Webanwendungen als ein Zusammenspiel der drei Rollen Model, View und Controller zu erstellen.

Ein Model repräsentiert einen Teil der Daten der Anwendung. Ein Controller handhabt Interaktionen des Benutzers oder anderer Komponenten und aktualisiert das Model, um eine Änderung im Status der Anwendung zu reflektieren und übergibt dann Informationen der View. Ein View nimmt notwendige Informationen vom Controller und erstellt eine Benutzeroberfläche, um diese Informationen anzuzeigen. Das ASP.NET-MVC-Framework fügt die Models, Views und Controller mit Schnittstellen zusammen, wodurch jede Komponente unabhängig von den anderen getestet werden kann.

Das Framework bietet Vorlagen für Webseiten und verschiedene Verfahren zur Authentifizierung von Benutzern an.

Eine View Engine ist der Programmteil innerhalb des Webservers, der aus ASP.NET-Quellcode letztlich HTML-Seiten zur Verwendung im Browser erzeugt. Die View Engines, die im ASP.NET-MVC-3- und MVC-4-Framework benutzt werden, sind die Razor View Engine und die Web Forms View Engine, die fester Bestandteil des MVC-3-Frameworks sind. Standardmäßig nutzt Razor .cshtml- oder .vbhtml-Dateien, während Web Forms .aspx-Dateien verwendet, um das Layout der Benutzeroberfläche zu erstellen. Prinzipiell können auch weitere View Engines verwendet werden.

Microsoft SQL Server^{iv}:



Der Microsoft SQL Server (auch kurz MSSQLServer) ist ein relationales Datenbankmanagementsystem von Microsoft.

Der SQL Server ist ein relationales Datenbankmanagementsystem, das sich am Standard der aktuellen SQL-Version orientiert. Der Microsoft SQL Server liegt in verschiedenen Editionen vor, die ein vielfältiges Angebot abdecken. Die Editionen unterscheiden sich vor allem im Preis, ihren Funktionen und Hardwareeinschränkungen. Der MSSQL-Server kann auch als Data-Warehouse genutzt werden, in dem es den Mitarbeitern in einem Unternehmen eine Sicht auf das Geschäft und dessen Daten ermöglicht. Durch seine Business Intelligence-Plattform bietet er eine skalierbare Infrastruktur, die es der IT ermöglicht, die Nutzung von Business Intelligence im gesamten Unternehmen zu fördern und Business Intelligence dort bereitzustellen, wo Anwender es wünschen.[5] Der SQL Server besteht aus vielen Services und Tools, wie Analysis Services, Reporting Services, Integration Services und Sync Services.

MSSQLServer verwendet für Datenbankabfragen die SQL-Variante T-SQL (Transact-SQL). T-SQL fügt hauptsächlich zusätzliche Syntax zum Gebrauch in Stored Procedures und Transaktionen hinzu. Weiterhin unterstützt MSSQL OLE DB und ODBC (Open Database Connectivity).

Seit SQL Server 2005 (Codename „Yukon“) werden unter anderem Programmiersprachen, welche auf der .NET CLR laufen, für das Erstellen von Stored Procedures unterstützt. Mit Visual Studio wird seit 2005 auch eine passende IDE mitgeliefert

TSQL Sprache:

Transact-SQL (T-SQL) ist eine proprietäre Erweiterung des SQL-Standards von Sybase und Microsoft. T-SQL erweitert den SQL-Standard um Funktionen wie Prozedurale Programmierung, lokale Variablen, Fehlerbehandlung, Funktionen zur Zeichenketten (STRING) Verarbeitung, Datumsverarbeitung und mathematische Operationen. Zusätzlich wurden Änderungen an der Funktionalität von DELETE und UPDATE Statements im Vergleich zu SQL durchgeführt.

Transact-SQL ist wesentlicher Bestandteil des Microsoft SQL Server. Bei Anwendungen erfolgt die Kommunikation mit der Instanz des SQL-Servers über Transact-SQL Anweisungen, dabei spielt die Benutzeroberfläche der Anwendung keine Rolle.

Durch den erweiterten Funktionsumfang von T-SQL können komplexere Abfragen erstellt werden, außerdem werden administrative Tätigkeiten innerhalb des SQL-Servers mit T-SQL Anweisungen durchgeführt. Wiederkehrende Aufgaben oder häufig verwendete Anweisungen, welche sonst vom Client ausgeführt werden, können als Stored Procedures auf dem Datenbankserver speichern werden, jeder Benutzer der Datenbank kann diese Programme aufrufen und nutzen. Über das Rechtemanagement der Datenbank kann der Zugriff auf Transact-SQL-Prozeduren gesteuert werden.

Um lokale Variablen zu verwenden bietet Transact-SQL die

Befehle: `DECLARE`, `SET` und `SELECT`

```
DECLARE @varName NVARCHAR(30)
SET @varname = 'Max Mustermann'
SELECT @varName = Name
FROM Kunde
WHERE KundeID = 1000
```

Mit dem SQL Server 2005 wurde die `TRY CATCH` Logik eingeführt um die Behandlung einer Exception (Ausnahme) zu unterstützen. Hierdurch können Entwickler ihren SQL-Code vereinfachen, da `@@ERROR` Prüfungen, die Prüfung ob eine Anweisung ohne Fehler durchgeführt wurde, nun nicht mehr nach jeder Anweisung durchgeführt werden müssen.

```
-- beginn einer Transaktion
BEGIN TRAN

BEGIN TRY
    -- Ausführung der Anweisungen
    INSERT INTO KUNDE(NAME) VALUES ( 'ASDF' )
    INSERT INTO KUNDE(NAME) VALUES (1234)

    -- Commit der Transaktion
    COMMIT TRAN
END TRY
BEGIN CATCH
    -- Dieser Codeblock wird ausgeführt sollte eine der Anweisungen
    fehlerhaft sein.
    -- Rollback der Transaktion
    ROLLBACK TRAN
END CATCH
```

Persönlicher Ausblick auf das Projekt

ⁱ https://de.wikipedia.org/wiki/Microsoft_Visual_Studio

ⁱⁱ <https://de.wikipedia.org/wiki/ASP.NET>

ⁱⁱⁱ https://de.wikipedia.org/wiki/ASP.NET_MVC_Framework

^{iv} https://de.wikipedia.org/wiki/Microsoft_SQL_Server