DevShop - Dokumentation

**ERSTELLT VON:**

Benjamin M. Knoll

**BETREUT VON:**

Josef Strasser-Leitner, ba

**TERMIN DER ABGABE:**

27.01.2023

# Eidesstattliche Erklärung

**Ehrenwörtliche Erklärung**

Hiermit bestätige ich, die vorliegende Projektarbeit namens „DevShop“ zur Gänze eigenständig und ausschließlich mit Hilfe der angegebenen Quellen verfasst zu haben. Die Arbeit hat einer anderen Prüfungsbehörde in gleicher oder ähnlicher Form noch nie vorgelegen.

Datum Unterschrift des Verfassers

# Abstract

Der Vertrieb von Produkten im Onlinehandel ist heutzutage nicht mehr wegzudenken. Aufgrund meines Ursprungs als Web-Entwickler habe ich bereits zahlreiche Onlineshops mittels hauseigenem CMS der Firma aufgebaut. Jedoch war mir der programmier-Prozess im Hintergrund bislang unbekannt, da sich meine Tätigkeiten hauptsächlich auf das Frontend beschränkten. Das Projekt „DevShop“ diente somit als perfekte Gelegenheit, mein Wissen zu erweitern und einen Einblick in die Backend-Entwicklung eines Shops zu erhalten.

**Inhaltsverzeichnis**

[1. Eidesstattliche Erklärung 1](#_Toc124446296)

[2. Abstract 2](#_Toc124446297)

[3. Einführung 4](#_Toc124446298)

[4. Konzept 5](#_Toc124446299)

[4.1. Datenbank 5](#_Toc124446304)

[4.2. Website 7](#_Toc124446305)

[5. Programm-Logik 8](#_Toc124446306)

[5.1. Einteilung des Programmcodes 8](#_Toc124446308)

[5.2. Datenbankanbindung 9](#_Toc124446309)

[6. Literaturverzeichnis 10](#_Toc124446310)

# Einführung

Ziel des Projektes ist es, Kenntnisse über den vollständigen Entwicklungsprozess eines Online-Shops zu erlangen. Primär liegt die Backend-Entwicklung im Fokus, welche sich auf den Authentifizierungsprozess sowie die Datenbankanbindung und Logik der einzelnen Funktionalitäten konzentriert.

Das Resultat aus dieser Entwicklung ist ein Demo-Shop, der die Möglichkeiten eines Onlinesystems mittels rechtegeschützter Eingabemasken im Backend und Ausgabe der angelegten Artikel im Frontend demonstriert. Sinn und Zweck der Ausgabe ist es, den Besucher über die gewünschten Artikel zu informieren, statt einen Bestellvorgang einzuleiten. Zudem stehen dem Anwender eine Suchfunktion und eine Auswahl zweier verschiedener Darstellungen der Artikel-Ausgabe zur Verfügung.

**-- PIC - SHOP --**

# Konzept

Für jede Software wird vor der Entwicklung zuerst ein Konzept erarbeitet, in dem alle Fragen hinsichtlich des Ausmaßes und Zieles des Projektes vorab geklärt und die verwendeten Technologien bestimmt werden. Die sorgfältige Planung jedes Prozesses dient nicht nur dazu, spät im Projekt eintretende fehlende Ressourcen oder falsch umgesetzte Funktionen vorzubeugen, sondern gibt auch einen Ablauf der notwendigen Arbeitsschritte vor.



## Datenbank

Eine der wichtigsten Säulen für Online-Shops ist die Datenbank. Der Entwurf der einzelnen Tabellen, ihre Abhängigkeiten zueinander sowie die benötigten Attribute und deren Datentypen erfolgt unter Berücksichtigung der Normalformen. Die Normalisierung von Datenbanken dient „bei relationalen Datenbanken zur Vermeidung von Redundanzen“ (Hosting-Technik, 2023).

**Technologie**

Für dieses Projekt wurde das relationale Datenbank-Managementsystem (DBMS) „SQL Server Management Studio 2018“ von Microsoft verwendet. Begründung der Wahl ist sowohl die bereits im Vorfeld vorhanden gewesene persönliche Erfahrung mit dem Programm als auch die Flexibilität des DBMS und die Möglichkeit, große Datenmengen schnell zu verarbeiten.

**Verwendete Tabellen**

Die ersten drei Tabellen sind den Standorten von Produkt-Herstellern gewidmet, und bestimmen Land, Bundesland und Stadt des Firmensitzes. Jede Stadt ist vom Bundesland abhängig, dieses wiederum vom jeweiligen Land.

Hersteller bzw. Firmen im Allgemeinen sind nur vom Land abhängig. Bundesland und Stadt dienen lediglich zur Bestimmung der Adresse, welche in einer eigenen Tabelle ausgelagert ist.

Da die Verwaltungsoberfläche hinter einem Login und Admin-Recht geschützt liegt, ist je eine Tabelle für Benutzer und verfügbare Rechte von Nöten. Jeder Benutzer kann nur genau ein Recht zugewiesen bekommen, weshalb diese Tabelle in einer 1:n Beziehung zur Rechte-Tabelle steht, und wird einer Firma bzw. einem Hersteller zugeordnet.

Grundidee des Shops ist es, aus verschiedenen Kategorien zu wählen und somit die dazugehörigen Artikel anzuzeigen. Hierfür benötigen die Kategorien eine eigene Tabelle. Diese können auch verschachtelt werden – sprich es können beliebig viele Unterkategorien mittels Attributes als Verweis auf den übergeordneten Datensatz erstellt werden.

Als Gruppierung zusammengehöriger Produkte dient die Tabelle der Produkt-Gruppen. Auch hier können den Gruppen wieder mithilfe eines Attributes in der Tabelle übergeordnete Einträge zugeordnet werden. Außerdem verweisen sie in einer 1:n Beziehung auf die Kategorien und sind abhängig vom Hersteller.

Folgend auf die Produkt-Gruppen gibt es die Tabelle der Produkte. Sie können als Überbegriff oder Zusammenfassung gleichartiger Artikel verstanden werden und verweisen mittels Fremdschlüssel auf die jeweilige Gruppe des jeweiligen Herstellers.

Letztlich ist eine Artikel-Tabelle von Nöten, welche auf ein Produkt verweist. Doch verschiedene Artikel haben möglicherweise verschiedene Einheiten, wie etwa Stück oder Liter. Um dies zu ermöglichen, werden die Einheiten in einer weiteren Tabelle ausgelagert. Weiters können Artikel zusätzliche Informationen erhalten, die zu Artikeln aus einem anderen Produkt variieren. Hierzu wird die Tabelle der Artikel-Header benötigt. Sie verweist über einen Fremdschlüssel auf ein Produkt und gibt an, welche Eigenschaften die Artikel im jeweiligen Produkt besitzen können.

## Website

Grundsätzlich wird die Seite in zwei Bereiche unterteilt. Eine Verwaltungsoberfläche, die nur für Admins zugänglich ist und alle Eingabemasken beinhaltet, und der für jeden zugängliche eigentliche Shop mit Menü, Artikelausgabe und Detailansicht eines Artikels. Um sofort eine eindeutige Unterscheidung beider Bereiche zu ermöglichen, gibt es für das Frontend und das Backend je ein unterschiedliches Layout. Lediglich die Grundeinstellungen wie verwendete Farben, Schriftarten/-größen und Abstandsdefinitionen bleiben gleich.

**Technologie**

Als Editor für HTML, CSS und JS-Code kam aufgrund des extrem schnellen und flüssigen Workflows das Programm „Sublime Text 3“ zum Einsatz.

Alle auf der Website dargestellten Icons und Bilder wurden eigenständig mit dem Designer-Programm „Affinity Designer“ kreiert bzw. bearbeitet. Die vielseitigen Einsatzmöglichkeiten reichen von simpler Bildbearbeitung bis hin zur professionellen Illustration und machen das Programm somit zur geeigneten Wahl für das Designen von Seiten und Erstellen von Icons.

Programmiert wurde mit „Visual Studio 2022“ von Microsoft unter Verwendung der Programmiersprache C#. Bei der Auswahl des Frameworks fiel die Entscheidung auf Blazor Server, da die am Server vorgerenderten Komponenten für interaktiv bedienbare Elemente am Client sorgen.

**Frontend**

Vorgesehen ist ein Menü mit Option zum Login/Logout, ein aufklappbares Menü zur Auswahl der Kategorien und ein Footer. Auf der Einstiegsseite sollen eine gewisse Anzahl an zufälligen Artikeln platziert sein. Die Shop-Seite soll eine Suchfunktion und maximal 20 Artikel pro Seite mit der Möglichkeit zum Durchklicken auf vorige und nachführende Seiten beinhalten, um den Benutzer nicht mit Datenmengen zu überfluten und die Seitenlänge kurz und übersichtlich zu halten. Je nach Vorzügen des Anwenders ist eine Listendarstellung und eine Boxen-Ansicht der Artikelauflistung geplant. Bei Auswahl eines Artikels soll dieser auf einer Detail-Seite dargestellt werden.

**Backend**

Die Verwaltungsoberfläche ist mit einer Navigation zu den jeweiligen Seiten der Eingabemasken zu versehen. Jede Maske beinhaltet neben den einzelnen Feldern zur Eingabe der Werte eine Auswahl aller bereits angelegten Datensätze und Buttons mit den Funktionen zum Löschen oder Erstellen eines Eintrages.

# Programm-Logik

Herzstück der Applikation ist die Logik der einzelnen Funktionalitäten sowie Verarbeitung und Austausch der Daten. Von notwendigen Einstellungen, die eine Verbindung zur Datenbank ermöglichen, über Hilfs-Klassen, um einen oft verwendeten Vorgang auszulagern, bis hin zu den eigentlichen Unterseiten der Website.



## Einteilung des Programmcodes

Damit eine effiziente und erleichterte Arbeitsweise gewährleistet werden kann, werden alle Abschnitte des Programmcodes, beispielsweise Klassen und Views, in einer übersichtlichen Struktur unterteilt.

**wwwroot**

Dieses Verzeichnis dient als Wurzelverzeichnis für alle Dateien, die vom Browser verwendet werden bzw. auf der Website eingebunden sind. Nicht nur sind hier CSS und JS-Dateien enthalten, sondern auch jegliche grafischen Inhalte, welche auf der Seite Verwendung finden. Raufgeladene Bilder werden ebenfalls hier abgelegt.

**Authentication**

In diesem Ordner spiegeln sich alle für den Authentifizierungsprozess notwendigen Klassen und Methoden wider. Letztendlich liegen die dazugehörigen Views allerdings in einem anderen, extra dafür vorgesehenen Verzeichnis.

**Data**

Alles rund ums Thema Datenverarbeitung ist hier abgelegt – sei es die Verbindung zur Datenbank an sich, Modell-Klassen, oder Methoden zum Handhaben gewisser Daten.

**Pages**

Die eigentlichen Unterseiten der Website sind in diesem Verzeichnis vorzufinden. Die dazugehörigen Klassen dienen lediglich der Dynamisierung der Oberfläche durch interaktive Funktionalitäten und der Anforderung benötigter Daten. Es findet jedoch keine Verarbeitung der Daten mehr statt.

**Shared**

Hier hinterlegte Dateien sind für alle Views zugänglich. Layouts für das Frontend und das Backend sind hier gespeichert.

## Datenbankanbindung

In der, wie in **4.2 Website** erläutert, gewählten Technologie Blazor Server wird eine Verbindung zur Datenbank mittels Entity Framework (EF) erstellt. EF ist ein von Microsoft entwickelter „ORM (Objektrelationaler Mapper), der den Zugriff auf die Datenbank über ein Objektmodell gewährleistet“ (Visual C# 2017 - Grundlagen, Profiwissen und Rezepte, 2018, S. 728). Für jede Tabelle in der Datenbank wird daher je eine Modell-Klasse benötigt. Die darin enthaltenen Properties repräsentieren die einzelnen Attribute der Tabelle. Über eine Kontext-Klasse werden anschließend alle Datenbankzugriffe geregelt. Da der Kontext bereits allerlei Funktionen für das Zusammenspiel mit der Datenbank besitzt, ist es nicht mehr notwendig, SQL-Code selbst zu schreiben.

Prinzipiell ist zwischen zwei Herangehensweisen zu unterscheiden, die für die Erzeugung der Objektmodelle sorgen. Auf der einen Seite steht die sogenannte „CodeFirst“-Methode zur Verfügung, in der die Modelle und der Kontext händisch angelegt werden und die Datenbank anhand dieser Modelle über einen Befehl erstellt wird. „DatabaseFirst“ auf der anderen Seite ist das genaue Gegenstück. Nach händischer Erstellung der Datenbank werden mit einem Befehl die Modelle und die Kontext-Klasse automatisch generiert.

Für dieses Projekt kam letztere Methode zum Einsatz, da die Festlegung der Relationen der einzelnen Tabellen mittels SQL schneller und einfacher erfolgt.

**Repositories (Repos)**

Aufgrund der einzelnen Modell-Klassen wird pro Klasse ein sogenanntes Repository angelegt. Dieses beinhaltet unter anderem Methoden zum Selektieren, Speichern, Aktualisieren und Löschen von Datensätzen. Alle weiteren Funktionalitäten decken jegliche Datenverarbeitung des Modells ab, beispielsweise die Konvertierung eines Modell-Objektes in ein View-Modell, welches nur Daten beinhaltet, die in der View auch verwendet werden.

**Unit-Of-Work**

Meistens werden in der View nicht nur Daten aus einem einzigen Modell, sondern aus mehreren verschiedenen Modellen benötigt. Hier kommt die Unit-Of-Work, oder kurz UOW, ins Spiel. Sie dient als Mittelmann zwischen den Repositories und den Views, indem sie jedes verfügbare Repo als Property einbindet. Somit muss in der View nur noch die UOW eingebunden werden, um Zugriff auf alle Repos zu erlangen.

## Authentifizierung

Microsoft stellt einen fix fertigen Authentifizierungsprozess mit Login, Logout, Registrierung und noch mehr für die gewählte Technologie zur Verfügung. Diese Seiten müssen lediglich einmal generiert werden und sind dann sofort einsatzbereit.

In diesem Projekt wurde jedoch auf die automatisch generierbare Variante von Microsoft verzichtet. Der komplette Prozess vom Hashen des Passwortes bis hin zum Setzen eines Cookies beim Login wurde vollständig händisch selbst gemacht. Grund für diesen nicht ganz trivialen Umweg sind zum einen die, auch wenn bis zu einem gewissen Grad manipulierbaren, mit der Generierung erstellten Tabellen in der Datenbank, von denen weniger als die Hälfte genutzt werden würde, mitsamt vieler überflüssiger Attribute. Zum anderen war der eigene Lerneffekt ebenfalls ein ausschlaggebender Faktor.

# Literaturverzeichnis

Doberenz, W., Gewinnus, T., Kotz, J., & Saumweber, W. (2018). *Visual C# 2017 - Grundlagen, Profiwissen und Rezepte.* München: Carl Hanser Verlag.

Hosting-Technik. (12. Januar 2023). *Normalisierung von Datenbanken*. Von https://www.ionos.de: https://www.ionos.de/digitalguide/hosting/hosting-technik/normalisierung-von-datenbanken/ abgerufen