# Design Document – Product management web site

This documentation tries to explain the contents of the application from a technical perspective and aims to be used as a design guide of the current functionality throughout the development phase and future improvements to the application as well as to act as a guide to any person that want to know the technical specifications of the application.

Any future changes should be translated into a new entry in the following table to keep track of the versions over the documentation and a summary of the changes done.

|  |  |  |
| --- | --- | --- |
| **Version** | **Updated by / date updated** | **Comments / changes** |
| 1.1 | Marcos Olivera - 10/07/2017 | Document created, versions table added and contents started to be added. |
|  |  |  |

## 1 – List of technologies used

### Technologies used



* IDE: Microsoft Visual Studio Community 2015 with updates
* .NET Framework: Version 4.6.1
* Language: C# 6.0
* ASP.NET MVC 5
* NuGet package manager to install required packages into Visual Studio
* Microsoft SQL Server 2016
* No dependency injection was used because of lack of time but I would use for example C# Ninject (I have used before and works well)
* Unit test project: *Created with the solution but only one test available, more tests should be added if the application started to grow so we have good code coverage.*
* Bootstrap v3.0
* Jquery v1.10.2

## 2 – Explanation of the architecture/ structure / design implementation and reasons of various choices

The required application and functionalities make use of the following technologies and tools:

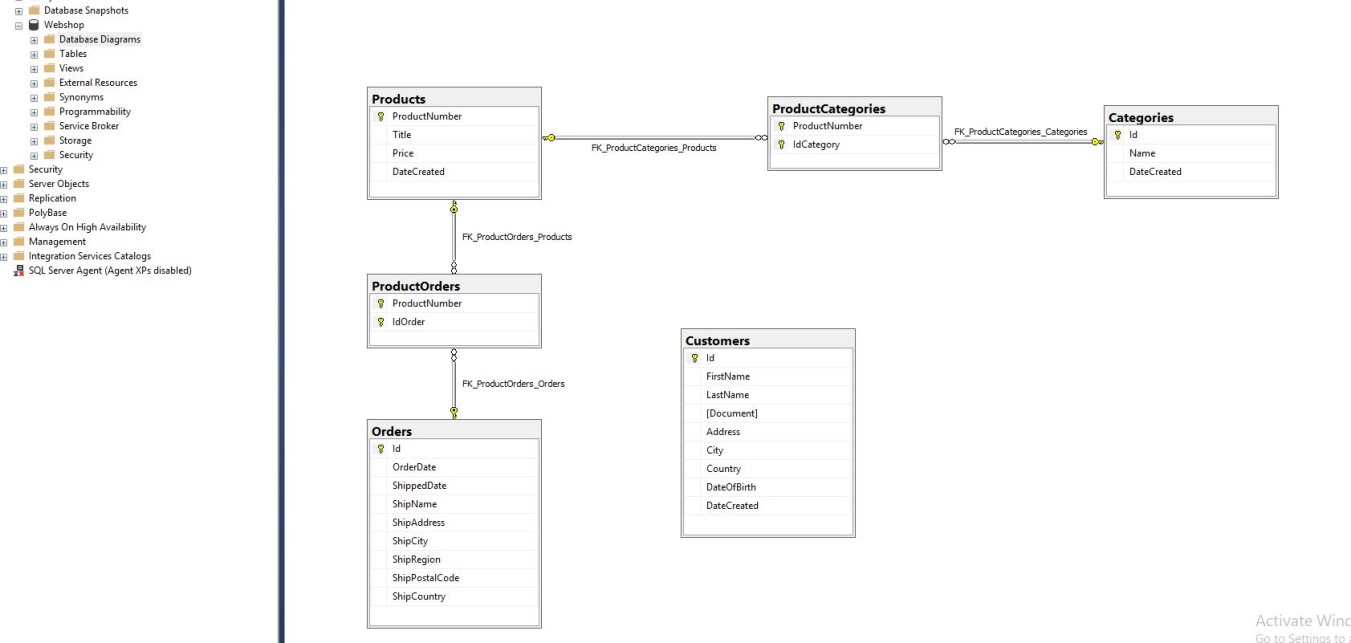
Visual studio community 2015 as the main IDE and C# as the main programming language, it uses the .NET Framework 4.6.1.

As per the required specifications the MVC architectural pattern is used, it provides a way of separation of concerns and gives the opportunity to write unit tests and automate manual testing; it is divided into 3 sections:

* Model - Represents the real world object and provides data to the view.
* View - Responsible for the look and feel.
* Controller - Responsible for taking the end user request and loading the appropriate Model and View.

Unit Test Project is integrated in the solution but as mentioned only as a matter of future usage.

Regarding the database we use SQL Server 2016 to create the “webshop” database for which there is a script named **WebshopCreateDB.sql** that can be used in order to create the database, also the following diagram shows the structure of the database:



## 3 – Considerations and assumptions.

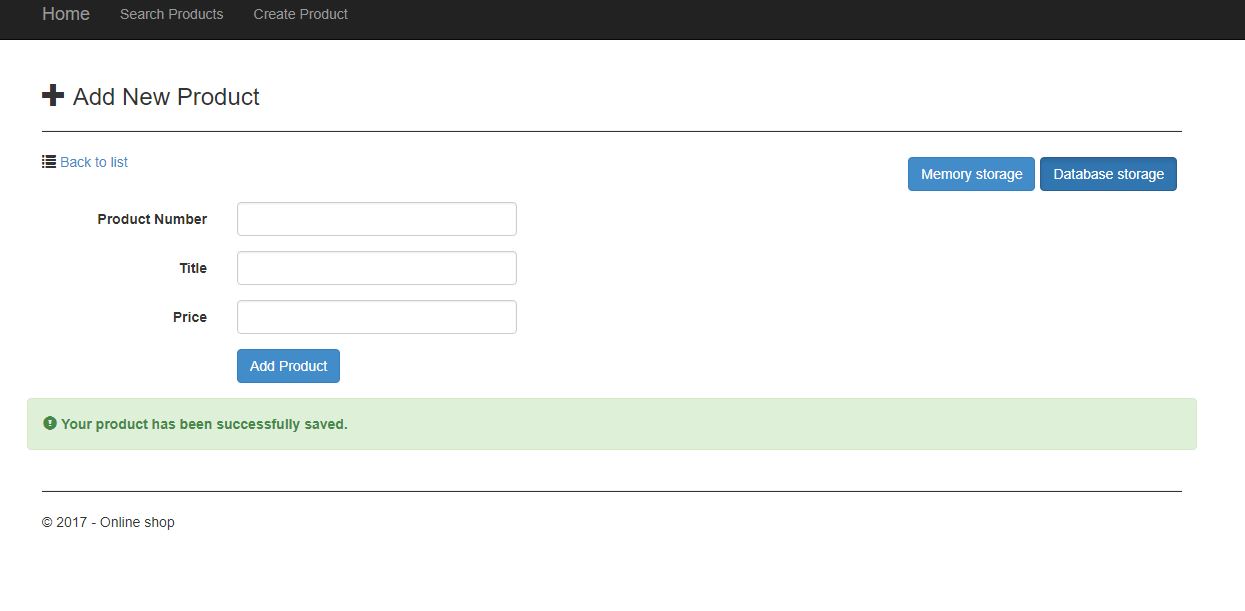
* As there is no login feature that can identify which user is accessing the application then the [Authorize] attribute is not used when accessing the controllers.
* There are two ways of data storage as requested but there is no synchronization between both (Application State and SQL Server database through Entity Framework).
* Some methods are provided as a reference for other operations in the class DataController that provides the access to EF mappings and operations with the DB.
* The class ProductModel inside ProductManagementWebSample.Models namespace has validation attributes that will take place in the View but it could be improved for example by adding validation for decimal numbers used with Price field.
* Selection of persistence method (memory or DB) is done by checking a hidden field containing the selected method; it can be improved to use a more secure way.

## 4 – Other technical details

* Comments are in place where necessary and to explain some functionality or notes about something that needed explanation.
* Custom error handler is defined for when there are errors to capture wrong behavior on database operations or calls to controller, also it can be improved by logging data in case of error (to files or into an audit log in the DB).
* Folders are in place to separate functionalities
* UI design is simple and user friendly providing the data and information in a clear manner to the user.

## 5 – Sample screenshots

Add Product:



Products List (with search depending on storage type selected):

