



Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding
Abteilung für Informatik

AWO - Administration and Website for Opticians

Eingereicht von: **Eva Pürmayr, 5AHIF**
Danijal Orascanin, 5AHIF
Datum: **April 4, 2018**
Betreuer: **Michael Bucek**
Projektpartner: **Augenoptik Aigner**

Declaration of Academic Honesty

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 4, 2018

Eva Pürmayr, Danijal Orascanin

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 4. April 2018

Eva Pürmayr, Danijal Orascanin

Inhaltsverzeichnis

1	Introduction	3
1.1	Ist-Situation	3
1.2	Zielsetzung	3
1.3	Aufgabenstellung	3
2	Verwendete Technologien	4
2.1	Visual Studio	4
2.2	C-Sharp	4
2.3	MVC	4
2.4	HTML	6
2.5	CSS	6
2.5.1	Media Query	6
2.6	Bootstrap	7
2.6.1	Grid-Systeme	7
2.7	SMTP	7
2.8	Base64	7
3	Beschreibung des Endproduktes	8
3.1	Brillenkategorien	9
3.1.1	Brillenmodelle	10
3.2	Wunschzettel	11
3.2.1	Kaufinteresse	11
3.3	Administration	11
3.3.1	Administratoren	11
3.3.2	Brillen	11
3.4	Allgemeine Informationen	13
3.5	Kontakt	14
4	Selbstevaluation	16
5	Zusammenfassung	17
A	Additional Information	21

Kapitel 1

Introduction

1.1 Ist-Situation

Im Unternehmen “Augenoptik Aigner” sind zur Zeit nur wenige Rechner im Einsatz, welche alle mit dem Betriebssystem “Windows 95” funktionieren. Auf diesen existiert ein Dos-Programm welches Kunden, Aufträge, Lieferanten und lagernde Produkte verwaltet. Weil mehrere Rechner im Einsatz sind, muss die aktuelle Version des Programms immer auf den Rechner kopiert werden, auf dem man dann gerne arbeiten würde. Dies ist natürlich sehr umständlich und zeitaufwendig, weshalb es Zeit wird, alle Rechner auf ein aktuelles Betriebssystem zu aktualisieren und eine zentrale Datenbank einzurichten, damit alle Rechner immer synchronisiert sind.

Die Website des Unternehmens Augenoptik Aigner ist aktuell weder responsive noch ist sie visuell ansprechend außerdem ist sie in der Funktionalität beschränkt. Die einzigen Funktionen die man hat sind allgemeine Daten anzusehen, wie die Geschäftszeiten, den Standort des Unternehmens, ein Impressum und man kann den Verkäufer noch mittels eines Formulars kontaktieren.

1.2 Zielsetzung

1.3 Aufgabenstellung

Kapitel 2

Verwendete Technologien

2.1 Visual Studio

Visual Studio ist eine Entwicklungsumgebung von Microsoft. Sie wird verwendet um Computerprogramme sowie Webseiten, Webapps, Webservices und mobile Apps herzustellen. Die aktuellste Version des Programmes unterstützt eine Vielzahl von Programmiersprachen: Visual Basic, .NET, C, C++, C++/CLI, C++/CX, C#, F#, SQL Server, TypeScript, Python, HTML, JavaScript und CSS. Weiters verfügt Visual Studio über einen Code editor mit IntelliSense (unterstützt den Programmierer mittels Vervollständigung des Codes).

2.2 C-Sharp

C# ist eine objektorientierte Programmiersprache. Sie wurde von Microsoft entwickelt ist jedoch mittlerweile Plattform unabhängig. Sie unterstützt zahlreiche Funktionen wie Lambda Expressions, Delegates, Enumerationen und auch einem direkten Speicherzugriff. Außerdem werden generische Methoden und Typen von C# unterstützt.

2.3 MVC

Model-View-Controller ist ein Design-Muster für die Trennung einer Software in 3 miteinander verbundene Teile. Das MVC-Muster wurde entwickelt um eine Wiederverwendung von Objektcode zu ermöglichen. Das Muster soll die Entwicklungszeit von Anwendungen mit Benutzeroberflächen stark reduzieren. Die 3 Hauptkomponenten die bei hier verwendet werden sollen stecken schon im Namen und zwar:

- **Model:** Ein Model ist eine simple Klasse und stellt das einfachste Element in MVC dar. Das Modell enthält die darzustellenden Daten. Es ist von View und Controller unabhängig. Es bedarf keiner Ableitung oder anderer Implementierung und ist im Prinzip ein Datencontainer. Die Models befinden sich in dem Order /Models. Wo bei die Klassen ohne weiteres in eine Klassenbibliothek ausgelagert werden können.

In einem Model befinden sich nichts anderes als Properties – Properties, die das Element bzw. die Daten darstellen, die für die Präsentation bestimmt sind. Ebenfalls kann beim Senden von Daten (POST) von einer View zum Controller eine Model-Klasse verwendet werden. Dabei wird der Controller-Action das Model als Parameter übergeben. Das Mappen zwischen POST-Parametern und Model findet über das Model-Binding statt.

- View: Sie kennt sowohl ihren Controller als auch das Model, dessen Daten sie präsentiert, ist aber nicht für die Weiterverarbeitung der vom Benutzer übergebenen Daten zuständig. Im Regelfall wird die View über Änderungen von Daten im Model mithilfe des Observers unterrichtet und kann daraufhin die aktualisierten Daten abrufen. Die View kann sich selbst aktualisieren, kennt das Model und ist dort registriert. Es gibt außerdem noch spezielle Views, die von allen Controllern bzw. allen Views verwendet werden können, die sogenannten Partial Views.
- Controller: Der Controller ist Verantwortlich für die Interaktion des Benutzers mit der Anwendung und für die Steuerung der Verarbeitung zwischen Model und View (Ablauf, Datenverarbeitung, entscheidet welche Views aufgerufen werden) Er kann den Kontext für verschiedene Models und Views darstellen und bestimmt die Möglichkeiten, mit denen die Benutzungsschnittstelle auf Benutzereingaben reagieren kann. Er ist nichts anderes als eine Klasse, die sich von `System.Mvc.Controller` ableitet. Eine Controller besteht meist aus mehreren Aktionen, die HTTP-Methoden, wie GET und POST, darstellen. Im einfachsten Falle übernimmt der Controller die Aufgabe des Business Layers und verarbeitet Daten. Dies kann ein Serviceaufruf, eine Datenbankabfrage, Dateihandling oder einfach auch nur ein Verarbeiten von Strings oder Zahlen sein.

Es verwandelt eine Webanwendung in eine wartbare, modulare und effizient entwickelte Anwendung. Anwendungsaufgaben in Models, Views und Controllern zu teilen macht Deine Anwendung sehr schlank. Neue Features sind einfach hinzugefügt, alte Features schnell in einer neuen Oberfläche verpackt. Die modular und unterteilte Logik erlaubt Entwicklern und Designern gleichzeitig an der Anwendung zu arbeiten. Dies beinhaltet ebenso die schnelle Entwicklung eines ersten Prototyps. Dadurch ist es ebenso möglich einen Teil der Anwendung zu verändern, ohne einen anderen Teil zu beeinflussen.

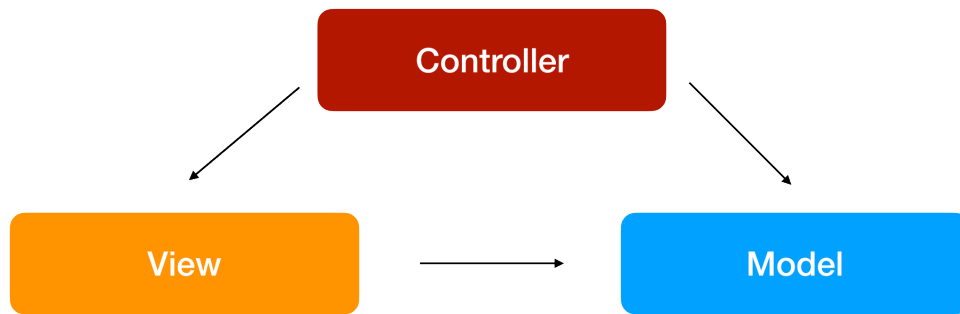


Abbildung 2.1: Model-View-Controller Konzept

2.4 HTML

Hypertext Markup Language oder auch HTML ist die Standardsprache für das erstellen von Webseiten und Web Applikationen. Gemeinsam mit CSS und Javascript gehört HTML zu den Eckpfeiler Technologien vom World Wide Web. Die meisten HTML-Elemente starten und enden jeweils mit einem Tag. Der Starttag beginnt immer mit dem Zeichen `<`. Nach dem `<` folgt der Elementname und dann wird der Starttag mit dem `>` Zeichen geschlossen. Nach dem Elementnamen können auch noch Attribute eingefügt werden. Der Endtag startet immer mit den Zeichen `</`, dann dem Elementnamen und endet mit einem `>` Zeichen.

2.5 CSS

Cascading Style Sheets ist eine style sheet Sprache welche gemeinsam mit HTML und DOM zu den Kernsprachen des World Wide Web gehört. Sie wird meist verwendet um einen visuellen Stilauf Webseiten festzulegen.

2.5.1 Media Query

Media Query ist eine Technik die es seit CSS3 gibt. Sie verwendet die `@media` Regel um einen bestimmt Block von Css Eigenschaften zu verwenden, wenn eine bestimmte Bedingung erfüllt wurde.

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: black;  
  }  
}
```

2.6 Bootstrap

Bootstrap ist ein freies web framework für das designen von Webseiten und Webapps. Es enthält HTML- und CSS basierte Design Vorlagen für Typografie, Buttons, Tabellen, Grid-Systeme, Formulare, Navigations- und andere Elemente. Außerdem kann man optional auch JavaScript Erweiterungen verwenden.

2.6.1 Grid-Systeme

Die Bootstrap Grid-Systeme sind mobile responsive was bedeutet das sie ihr Layout auf die größe des Gerätes anpassen. Würde man das unten angeführte Beispiel auf dem Computer aufrufen würden 3 gleich große nebeneinander gestellte Spalten angezeigt werden, am Handy würde sich dies platztechnisch nicht ausgeben deshalb werden diese 3 Spalten automatisch untereinander gestellt.

```
<div class="row">
  <div class="col-sm-4">
    Spalte 1
  </div>
  <div class="col-sm-4">
    Spalte 2
  </div>
  <div class="col-sm-4">
    Spalte 3
  </div>
</div>
```

2.7 SMTP

Simple Mail Transfer Protocol ist ein standard Internetprotokoll zur Übertragung von emails. SMTP wird vorrangig für das versenden von Mails benutzt, da zum Empfang von Mails andere Protokolle wie zum Beispiel POP3 oder IMAP verwendet werden.

2.8 Base64

Base64 ist eine Gruppe ähnlicher Binär zu Text Codierungsschemas, die Binärdaten in einem ASCII-Zeichenfolgenformat darstellen, indem sie in eine Radix-64-Darstellung übersetzt werden.

Kapitel 3

Beschreibung des Endproduktes

3.1 Brillenkategorien

Über den Menüpunkt "Home" gelangt man zu einer Übersicht aller Brillenkategorien. Dies ist gleichzeitig die Hauptseite der Webseite. Wird eine der 3 Kategorie (Herrenbrillen, Damenbrillen und Kinderbrillen) ausgewählt so werden die dazugehörigen Brillen auf einer neuen Seite angezeigt.

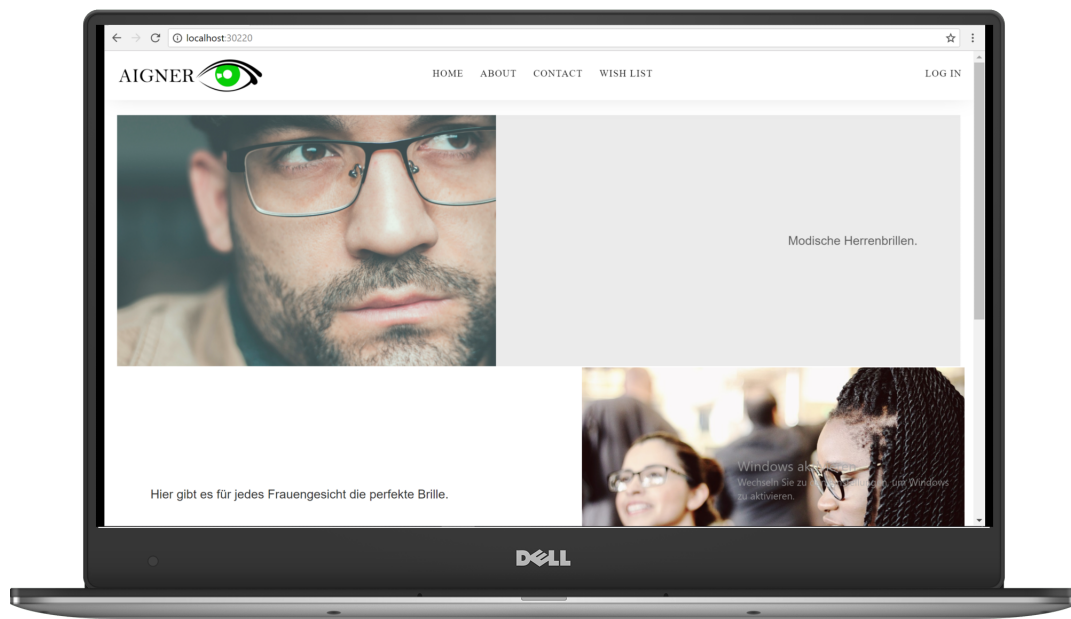


Abbildung 3.1: Brillenkategorien

3.1.1 Brillenmodelle

Die Brillenmodelle sind in drei Kategorien untergliedert "Herrenbrillen", Frauenbrillen und "Kinderbrillen". Auf jeder der Unterseiten werden die dazugehörigen Brillen mit einem Foto, dem Titel und dem Preis dargestellt. Jede Brille kann dann einzeln mittels eines Buttons auf den Wunschzettel gesetzt werden. Die Brillenmodelle werden nach der Kategorie gefiltert. Auf der "Home"-Seite wird eine Kategorie ausgewählt, alle Brillenmodelle, die diese Kategorie haben, werden angezeigt. Jedes Brillenmodell hat einen Preis, nach belieben können die Brillen nach dem Preis auf- oder absteigend sortiert werden.

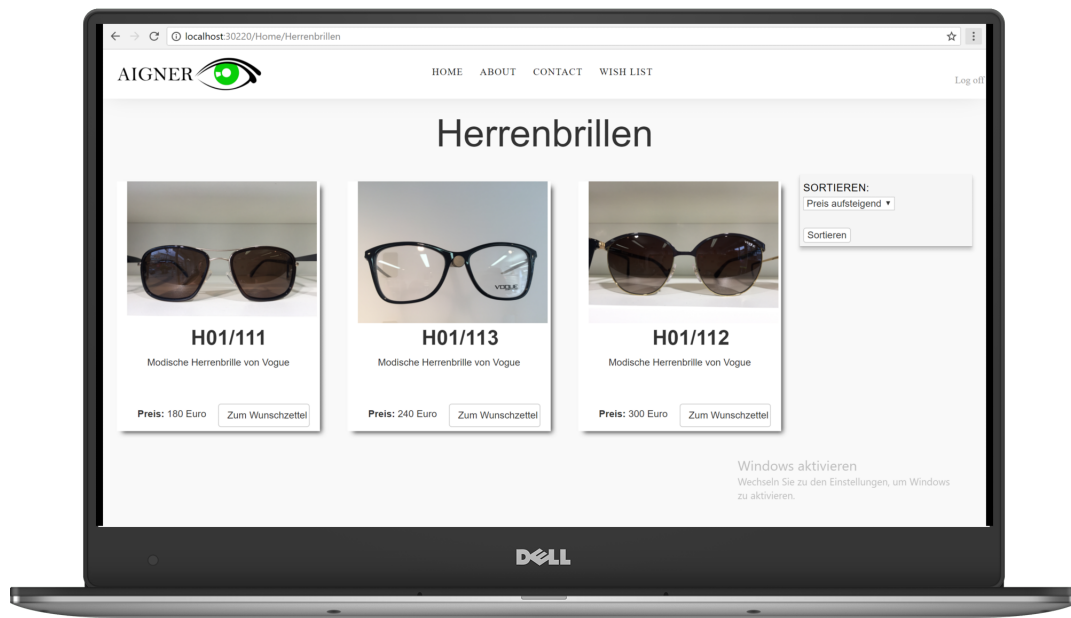


Abbildung 3.2: Model-View-Controller Konzept

3.2 Wunschzettel

Unter dem Punkt "Wunschzettel" der Navigation bar findet man alle Brillenmodelle die man zuvor zu diesem hinzugefügt hat. Hier kann man einzelne Produkte wieder vom Wunschzettel entfernen. Außerdem kann man den Verkäufer über seine Wunschobjekte informieren.

3.2.1 Kaufinteresse

Über den Wunschzettel kann man dem Verkäufer sein Kaufinteresse mitteilen. Hierzu muss man zuerst seinen Namen eingeben und dann bestätigen, dass man dem Verkäufer sein Interesse über die ausgewählten Modelle mitteilen möchte. Wird dies bestätigt wird ein neuer Eintrag in einer Tabelle erstellt, in welcher alle Brillenmodelle gelistet sind an denen Kunden interessiert sind. Außerdem verschickt man durch das Bestätigen auch eine E-mail an den Verkäufer welche in darüber Benachrichtigt das neue Modelle in die Tabelle eingetragen wurden.

3.3 Administration

3.3.1 Administratoren

Befindet man sich auf der Hauptseite findet man rechts oben einen "Einloggen"-Button wird dieser gedrückt wird man weitergeleitet auf eine Login Seite auf der sich der Administrator mittels seiner E-Mail Adresse und dem Passwort einloggen kann. Wurden die korrekten Daten vom Administrator eingegeben wird er dann wieder auf die Hauptseite zurückgeleitet dort ist erscheint nun ein Button "Administrationsbereich". In diesem bekommt man eine Übersicht aller Administratoren man kann hier dann neue hinzufügen und aktuelle Administratoren löschen. Außerdem befindet sich auf der Seite auch noch ein Button "Brille" über diesen gelangt man zu einer Übersicht aller Brillen.

3.3.2 Brillen

Der Administrator erhält eine Übersicht aller Brillen und hat die Möglichkeit Brillen zu löschen, zu bearbeiten oder neue hinzuzufügen. Beim klicken des Buttons "Neue Brille hinzufügen" wird man auf die Seite "Brille hinzufügen" weitergeleitet. Neben jeder Brille befinden sich 2 Buttons "Löschen" und "Bearbeiten". Beim drücken des "Löschen"-Buttons wird das Brillenmodell gelöscht und eine aktualisierte Liste dargestellt. Beim betätigen des "Bearbeiten"-Buttons wird man auf die Seite "Brille bearbeiten" weitergeleitet.

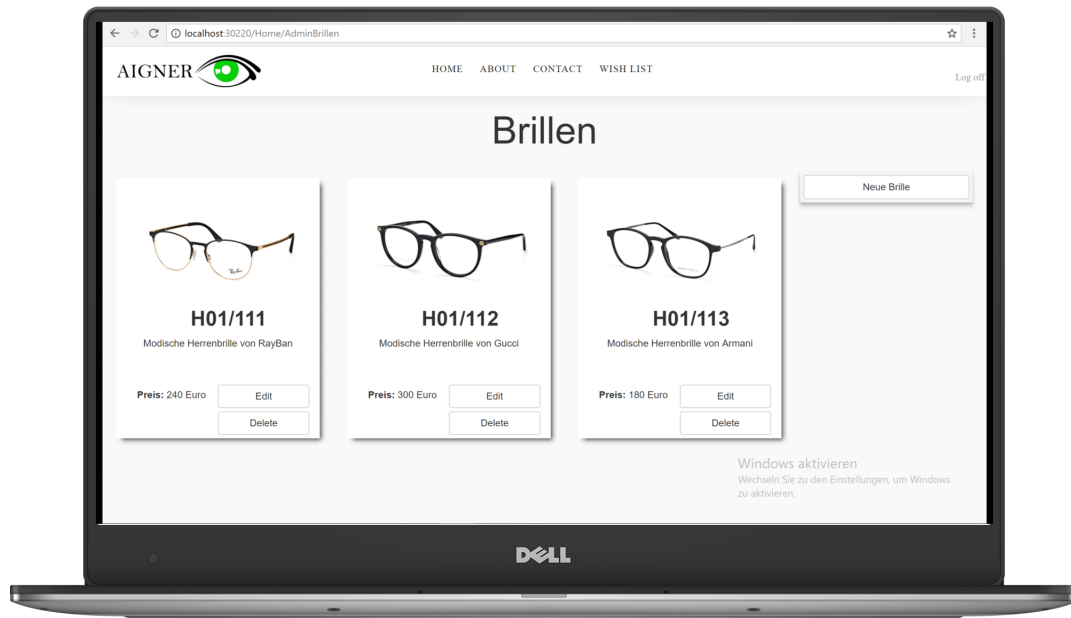


Abbildung 3.3: Model-View-Controller Konzept

Brille hinzufügen

Auf dieser Seite kann der Administrator neue Brillenmodelle einfügen dazu muss er den Titel und den Preis eingeben und ein Foto der Brille einfügen, dann kann er auf "Brille hinzufügen" drücken. Dies führt dazu, dass die Brille in der Datenbank eingefügt wird und dass er auf die Seite "Brillen" zurückkehrt wo die aktualisierte Liste der Brillen zu sehen ist.

Brille bearbeiten

Auf dieser Seite kann der Administrator den eingegebenen Titel, den Preis oder auch das Bild einer bereits vorhandenen Brille ändern. Nachdem die Änderungen vorgenommen wurden und er diese mit dem "Brille bearbeiten" Button bestätigt wird der Administrator automatisch auf die Seite "Brillen zurückkehren" wo die aktualisierte Liste aller Brillen zu sehen ist.

3.4 Allgemeine Informationen

Um allgemeinen Informationen über das Unternehmen zu erhalten, gibt es einen Menüpunkt "About", hier werden alle Informationen wie die Öffnungszeiten, Fax-, Mobil- und die Telefonnummer angezeigt, außerdem wird der Standort des Unternehmens auf einer eingebunden Google Maps Karte dargestellt.



Abbildung 3.4: Model-View-Controller Konzept

3.5 Kontakt

Bei Fragen über die Website, das Unternehmen, die Brillenmodelle und allem anderen kann der Verkäufer mittels eines Formulars kontaktiert werden. Um zum Kontaktformular zu gelangen gibt es in der Navigation bar den Punkt "Kontakt". Dort gibt man seinen Namen, die eigene E-mail Adresse und die Nachricht ein und das Formular kann dann gesendet werden. Die eingegeben Nachricht wird dann über einen SMTP Server an die E-Mail Adresse des Verkäufers gesendet. Die Mail beinhaltet die Nachricht sowie den eingegebenen Namen und auch die E-Mail Adresse somit kann der Verkäufer auf die

erhaltene Nachricht antworten.

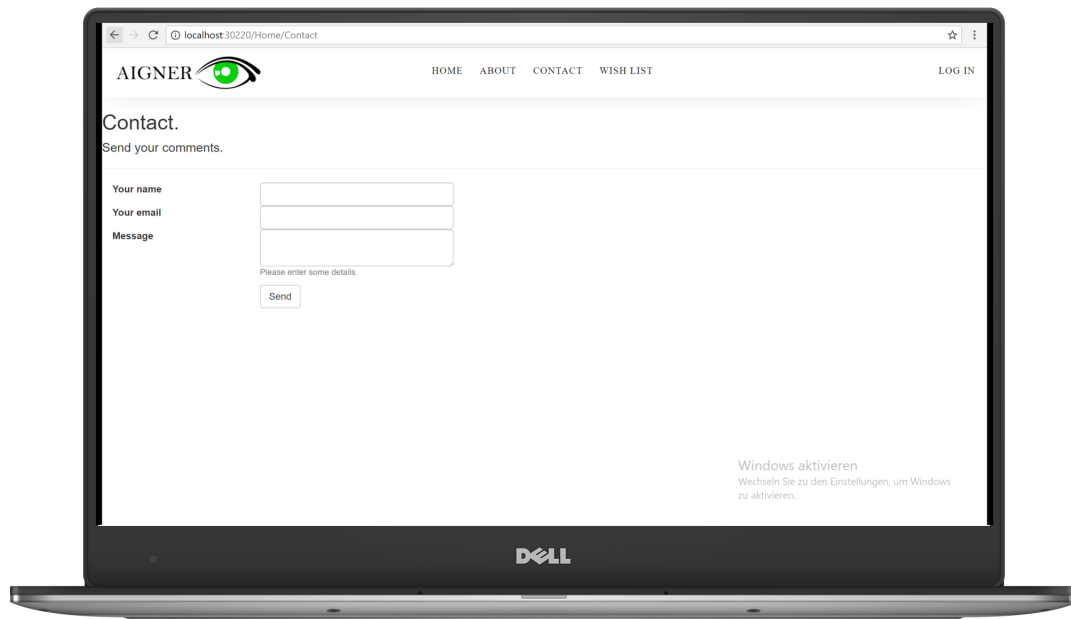


Abbildung 3.5: Model-View-Controller Konzept

Kapitel 4

Selbstevaluation

Kapitel 5

Zusammenfassung

Aufgabenstellung: Es ist eine responsive Website zu entwickeln. Auf dieser sollen neben der Lage und den Öffnungszeiten auch die verschiedenen Brillenmodelle angezeigt werden. Der User kann über die Website dem Verkäufer sein Kaufinteresse mitteilen. Der Verkäufer kann sich auf der Website anmelden und, dann seine Tabellen ansehen und bearbeiten um zum Beispiel von zu Hause eine neue Brille einzutragen. Die Website soll für Kunden ansprechend sein und eine gute Übersicht über die Brillenmodelle bieten außerdem sollte der Administrator der Website die Möglichkeit haben von zu Hause seine Brillenmodelle zu verwalten.

Realisierung: Die aktuelle Website ist nicht mobile responsive und sie der einzigen Funktionen sind den Verkäufer zu kontaktieren und allgemeine Daten anzusehen. Die Website hätte man in verschiedenn Programmiersprachen umsetzen können. Sie wurde aufgrund der Vorkenntnisse mittels Asp .net entwickelt. Es wurde das Mvc-Pattern umgesetzt.

Aufgabenstellung: Es ist ein Programm zu entwickeln, welches alle Kunden, Brillen- und Kontaktlinsenaufträge, den Lagerbestand und alle Lieferanten verwaltet. Dieses Programm soll auf mehreren Rechnern synchronisiert aufrufbar sein und mit einer Datenbank verbunden werden.

Realisierung: Das aktuelle System ist veraltet und ist in der Bedienung sehr umständlich und zeitaufwendig. Es läuft auf mehreren Rechnern unter dem Betriebssystem Windows

Literaturverzeichnis

[Wik17] Wikipedia. .net, March 2017. Page Version ID: 163867861. URL: <https://de.wikipedia.org/w/index.php?title=.net&oldid=163867861>.

Abbildungsverzeichnis

2.1	Model-View-Controller Konzept	6
3.1	Brillenkategorien	9
3.2	Model-View-Controller Konzept	10
3.3	Model-View-Controller Konzept	12
3.4	Model-View-Controller Konzept	14
3.5	Model-View-Controller Konzept	15

Tabellenverzeichnis

Anhang A

Additional Information