

## **Bachelor-Thesis**

zur Erlangung des akademischen Grades

Bachelor of Science (B. Sc.)

an der Hochschule für Technik und Wirtschaft des Saarlandes

im Studiengang Kommunikationsinformatik

der Fakultät für Ingenieurwissenschaften

## **Migration eines webbasierten Bestellungssystems in eine .Net-Umgebung mit Content-Management Funktionalität**

vorgelegt von

Bozhidar Aleksandrov

betreut und begutachtet von

Prof. Dr. Helmut Folz

Thomas Beckert, M.Sc.

Saarbrücken, 30. September 2018



# Selbständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit (bei einer Gruppenarbeit: den entsprechend gekennzeichneten Anteil der Arbeit) selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich erkläre hiermit weiterhin, dass die vorgelegte Arbeit zuvor weder von mir noch von einer anderen Person an dieser oder einer anderen Hochschule eingereicht wurde.

Darüber hinaus ist mir bekannt, dass die Unrichtigkeit dieser Erklärung eine Benotung der Arbeit mit der Note „nicht ausreichend“ zur Folge hat und einen Ausschluss von der Erbringung weiterer Prüfungsleistungen zur Folge haben kann.

*Saarbrücken, 30. September 2018*

---

Bozhidar Aleksandrov



# Zusammenfassung

Kurze Zusammenfassung des Inhaltes in deutscher Sprache, der Umfang beträgt zwischen einer halben und einer ganzen DIN A4-Seite.

Orientieren Sie sich bei der Aufteilung bzw. dem Inhalt Ihrer Zusammenfassung an Kent Becks Artikel: <http://plg.uwaterloo.ca/~migod/research/beck00PSLA.html>.



*We have seen that computer programming is an art,  
because it applies accumulated knowledge to the world,  
because it requires skill and ingenuity, and especially  
because it produces objects of beauty.*

— Donald E. Knuth [2]

## Danksagung

Hier können Sie Personen danken, die zum Erfolg der Arbeit beigetragen haben, beispielsweise Ihren Betreuern in der Firma, Ihren Professoren/Dozenten an der htw saar, Freunden, Familie usw.





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Genutzte Technologie und Services</b>	<b>3</b>
2.1	HTML . . . . .	3
2.2	CSS . . . . .	3
2.3	JavaScript . . . . .	3
2.4	JQuery . . . . .	3
2.5	Ajax . . . . .	3
2.6	AngularJS . . . . .	4
2.7	Umbraco . . . . .	4
2.8	1&1 Website Check . . . . .	5
2.9	HTTP Obeservatory . . . . .	5
<b>3</b>	<b>IST-Analyse</b>	<b>7</b>
3.1	Beschreibung des User Interface (UI) des Bestellsystems . . . . .	7
3.2	Beschreibung der Funktionalität . . . . .	8
3.2.1	Anmelde- und Registerformular . . . . .	9
3.2.2	Auftraggeber-Verwaltung . . . . .	9
3.3	Testen . . . . .	16
3.4	Warum ist eine Migration notwendig? . . . . .	18
<b>4</b>	<b>Anforderungsanalyse</b>	<b>19</b>
4.1	Kundenverwaltung . . . . .	19
4.1.1	Kundenerfassung . . . . .	19
4.1.2	Kundenansicht . . . . .	19
4.1.3	Auftragsgeber-Ansicht . . . . .	20
4.1.4	Kommunikation . . . . .	20
4.2	Artikelverwaltung . . . . .	20
4.2.1	Artikel erfassen, ändern und löschen . . . . .	20
4.3	Auftragsverwaltung . . . . .	20
4.3.1	Übersicht . . . . .	20
4.3.2	Detailansicht . . . . .	20
4.4	E-Mail-Verwaltung . . . . .	21
4.5	Umsatzverfassung . . . . .	21
<b>5</b>	<b>Konzeption und Implementierung</b>	<b>23</b>
5.1	Aufbau vom Umbraco . . . . .	23
5.2	Kundenverwaltung . . . . .	25
5.2.1	Kundenerfassung . . . . .	25
	<b>Literatur</b>	<b>27</b>
	<b>Abbildungsverzeichnis</b>	<b>29</b>
	<b>Tabellenverzeichnis</b>	<b>29</b>

<b>Listings</b>	<b>29</b>
<b>Abkürzungsverzeichnis</b>	<b>31</b>
<b>A Erster Abschnitt des Anhangs</b>	<b>35</b>

# 1 Einleitung

Seit mehreren Jahrzehnten zeichnet sich eine immer weiter voranschreitende Digitalisierung der Gesellschaft ab. Manche Wirtschaftsbetriebe und Einrichtungen haben sich schon früh damit arrangiert und entsprechende Produkte angeschafft. Dies umfasst fertig käufliche Hard- und Softwareprodukte, sowie individuell entwickelte Lösungen. Diese sind, wie die meisten digitalen Güter, einem schnellen Alterungsprozess unterworfen. Das können Schnittstellen sein, welche vom Hersteller nicht mehr unterstützt werden, oder eine neue Betriebssystem Version die nicht mehr Unterstützt wird. Wenn die Kunden direkt mit einer Software interagieren, ist Userexperience und Design ein nicht zu unterschätzender Faktor.

Die meisten Softwareinfrastrukturen wachsen meist mit einem Unternehmen und seinem Bedarf nach digitalen Lösungen. Das bedeutet, dass eine verworrene Struktur aus Abhängigkeiten entstehen kann. Das ist dabei eher die Regel als die Ausnahme.

Elektronische Datenverarbeitung (EDV)-Systeme haben nicht die klassischen Verschleißerscheinungen, so wie man sie von klassischen Betriebsmitteln kennt. Allerdings entsteht auch so ein Interesse nach einem gewissen Nutzungszeitraum die bestehende Software zu ersetzen. Man bezeichnet diesen Alterungsprozess, welchen man eingehend in der Softwarequalitätsforschung untersucht, als Softwarealterung. Software ist „weich“ und man sollte annehmen sie sei leicht änderbar und wartbar. Dies kann allerdings mit fortschreitendem Alter teurer sein als die Migration zu einem neuen System. Firmen die nicht mit der Zeit gehen werden schnell als alt und uninnovativ wahrgenommen. Dies kann sich schnell auf den Umsatz eines Unternehmens auswirken. Deshalb sind Migrationen gerade im Bereich des Web-Developments besonders häufig, bei denen man von Grund auf ein neues System erstellt.

Es gibt mehrere Fälle von Softwaremigrationen. In manchen Fällen kann Software Hardware ersetzen. In andere Fällen ersetzt wiederum Software Hardware. Wesentlich häufiger ist jedoch das alte Software durch neue ersetzt wird, sowie alte Hardware durch ihre neueren Pendants. In dieser Arbeit wird die Umstellung eines Bestellsystems eines Catering Services abgebildet werden. Dies bedeutet, das es sich bei dem Thema der Arbeit um eine Software zu Software Migration handelt. Genauer um eine modellgetriebene Softwaremigration. Dabei wird das aus dem beginn dieses Jahrtausend stammende Bestellsystem, welches auf einem Windows Imaging Components (WIC)-Plugin basiert und einem Active Server Pages (ASP)-Backend.

Da die damals verwendeten Technologien vom Hersteller Microsoft seit geraumer Zeit End-of-Life gesetzt wurden, ist eine Migration zu einem aktuelleren Technologiestack zwingend erforderlich. Das Ziel des Migrationsprojekt ist es, das aktuelle Bestellsystem, welches immernoch auf ASP basiert, durch einen modernen Technologiestack zu ersetzen.

Dazu zählt die neue Kozeptionierung des Frontend. Dies soll zur Verbesserung und Erleichterung der Bedienbarkeit führen. Dazu wird eine neue Seite, welche auf Umbraco und ASP.NET basiert, erstellt.

Im nachfolgenden Kapitel werden alle verwendeten Technologien erörtert und kurz erklärt. Danach wird im darauf folgenden Kapitel der technische und optische Stand der aktuellen Internetpräsenz analysiert.

Im darauf folgenden vierten Kapitel erfolgt die Erfassung und Anforderungsanalyse der Problemstellung deren theoretischen Lösungen. Im nachfolgenden Kapitel wird

## *1 Einleitung*

darauf aufgebaut und es erfolgt die praktische Umsetzung der Lösungssätze. Im letzten Kapitel wird die Qualität der Umsetzung erörtert und persönliche Designentscheidungen begründet. Danach folgt ein Ausblick auf weitere Verbesserungsmöglichkeiten.

## 2 Genutzte Technologie und Services

In diesem Kapitel beschäftigt man sich mit den verwendeten Technologien und Webservices. Diese sind in der Webentwicklung sehr entscheidend, da davon die Userexperience und Wartbarkeit abhängt. Besonderer Augenmerk wurde dabei auf die möglichst weite Verbreitung der verwendeten Technologien gelegt. Dies ist ein Vorteil, da daraus resultiert das es eine große Community gibt, die die Projekte aktuell hält.

### 2.1 HTML

Die Grundlegende Sprache für das erstellen und Rendern von Internetseiten ist Hypertext Markup Language (HTML). Es ist im Grunde DIE Schlüsseltechnologie um Internetseiten aus dem World Wide Web (WWW) anzuzeigen. Natürlich kann man HTML auch für das erstellen und webbasierten lokalen Graphical User Interface (GUI) genutzt werden. Weitere Informationen findet man unter [\[mozilla-html\]](#).

### 2.2 CSS

Cascading Style Sheets (CSS) ist eine moderne Technologie zur Gestaltung von Internetseiten. Dies dient der Gestaltung von HTML-Texten und liefert auch teilweise dynamische Funktionalität. Weitere Informationen findet man unter [\[mozilla-css\]](#).

### 2.3 JavaScript

JavaScript ist eine objektorientierte Interpretersprache, die Webseiten erst dynamisch macht. Sie wird in diesem Kontext im Webbrowser des Anwenders ausgeführt und stellt die Client-Seite einer Applikation dar. [Link](#)

### 2.4 JQuery

JQuery ist ein weit verbreitetes JavaScript-Framework. Es stellt Funktionen zur Verfügung welche ein leichtes manipulieren des Seiteninhalts ermöglicht. So kann die Entwicklung von komplexen Webprojekten bedeutend beschleunigt werden.

### 2.5 Ajax

Ajax ermöglicht es Daten asynchron zwischen Browser (Client) und Server zu übertragen. Dies ermöglicht partielle Veränderungen der Internetseite, ohne das ein kompletter Refresh des Seiteninhalts vorgenommen werden muss.

### 2.6 AngularJS

Hierbei handelt es sich um ein JavaScript Framework, welches hochdynamische WebAnwendungen oder lokale Anwendungen mit Webtechnologie ermöglicht.

### 2.7 Umbraco

Umbraco ist ein Content Management System (CMS). Es dient zum Erstellen, Bearbeiten und zur Verwaltung dynamischer Webseiten. Umbraco basiert auf C# und auf der ASP.Net-Technologie. Heutzutage werden Microsoft SQL Server, My SQL, VistaDB, Peta Poco und weitere Datenbanken verwendet. Dieses CMS ist Open Source und die erste Version ist vom dänischen Software-Entwickler Niels Hartving im Jahr 2000 veröffentlicht worden.

Aus folgenden Gründen hat man sich bei der Umsetzung für Umbraco entschieden [1, S. 12]:

- Umbraco ist ein flexibles CMS. Es gibt keine unnötigen Optionen und Schaltflächen. Alles ist einfach zu benutzen und zu verstehen.
- Der intuitive Editor ermöglicht es jede Art von Content einfach einzupflegen. Seiten sind einfach zu bearbeiten oder zu aktualisieren und wird nach dem selben eingängigen Paradigma dargestellt. Es ist ohne Bedeutung mit welchem Gerät auf die Webseite zugegriffen wird - Umbraco ist immer responsiv.
- Es gibt keine Einschränkung welche Webentwicklungsprogrammiersprache man nutzen muss. Umbraco ist sehr anpassungsfähig.
- Sehr gut angepasst für agile Prozesse: „Im Vergleich zu anderen Systemen geht der Livegang Deines Projekts damit sehr schnell. Umbraco unterstützt die agilen Prozesse der modernen Digitalbranche, bei denen es essenziell ist, dass Editoren immer und überall Content publizieren können, ohne damit "Content Freeze" zu verursachen. Gleichzeitig sollen auch Entwickler Bugfixes und Features schnell einbauen können. Umbraco sorgt dafür, dass der Flow nie endet“.
- Umbraco CMS ist integrierbar. Man kann E-Commerce-Plattform, CRM (Custom Relationship Management) System oder 3rd-Party Personalisation Engine verwenden. Ohne Probleme können individuelle Systeme integriert werden. Wegen Application-Programm-Interface (API) werden alle Daten mit sichtbarem Content mit dem Umbraco Front- und Backend vernetzt.
- Die Community vom Umbraco ist groß. Freundliche und aktive Umbraco Nutzer helfen jeder Zeit gegenseitig bei der Verbesserung des Codes. Viele aktive Tester sorgen dafür, dass Umbraco ständig verbessert wird.
- Das Modell der Lastverteilung ist in ASP.NET integriert. Jede ASP.NET-Webseite besitzt eigene Session-Verwaltung, die so konfiguriert werden kann, dass sie die Daten auf den SQL Server verlegen kann. So lassen sich Daten in einem gemeinsamen Datenspeicher ablegen. So ist es möglich, dass jeder Server auf den Datenstand eines Nutzers zugreifen kann.
- Volle Versionskontrolle, volle Integration in vorhandene Strukturen, zeitgesteuertes Veröffentlichen, Workflow-Orientierte Seitenverwaltung, schnelle Seitenvorschau

vor der Veröffentlichung, Mehrsprachigkeit, Papierkorb zum einfachen Wiederstellen von gelöschten Elementen und seine Lizenzkostenfreiheit sind weitere Eigenschaften, weshalb man sich für Umbraco als CMS entschieden hat.

## 2.8 1&1 Website Check

Diese Anwendung überprüft, wie gut die betrachtete Webseite ist und was noch optimiert werden kann. Vier Aspekten werden geprüft (6):

- Darstellung der Webseite
- Auffindbarkeit in Suchmaschinen
- Darstellung der Webseite
- Sicherheit der Webseite
- Geschwindigkeit der Webseite

Wenn die Internetadresse eingegeben wurde, wird Webseite aufgerufen. Danach wird der Quellcode analysiert.

## 2.9 HTTP Observatory

Mozilla HTTP Observatory ist ein Set von Tools, die zur Analyse und als Informationsquelle für Verbesserungen der Website dienen.





### 3 IST-Analyse

In dieser Arbeit wird das Bestellsystem von der Party-Service-Website "jungekueche" betrachtet. Dazu zählen auch das Verwaltungssystem des Auftraggebers und die Verwaltungsseite des Kunden. Zuerst wird das Userinterface aus der Sicht von allen drei Akteuren beschrieben. Danach wird die Funktion des Systems evaluiert.

#### 3.1 Beschreibung des User Interface (UI) des Bestellsystems

Die Anmeldung und die Registrierung für neue Nutzer befindet sich auf einer gemeinsamen Seite. Das Anmeldeformular hat zwei Felder, in denen der Benutzer seinen PIN und seine E-Mail eingeben muss. Neben den Feldern steht ein Hinweis, welcher den Kunden mit den nötigen Informationen versorgt, wie er sich anmelden oder registrieren kann.

Abbildung 3.1: Die bisherige Eingabemaske für das Kundenlogin

Falls der Kunde keine PIN hat, muss er sich registrieren. Dieses Formular besteht aus fünf Abschnitten. Im ersten muss der Kunde seine persönlichen Daten angeben. In dem zweiten steht die Informationen des Auftrags. Menü Auswahl ist der nächste Abschnitt. Im vorletzten wird die Zubehörauswahl und Serviceauswahl angezeigt. Der letzte Abschnitt ist die Erklärung zu dem Registrierungsprozess.

Abbildung 3.2: Die bisherige Eingabemaske für das Registrierungsformular

Im Fall einer Registrierungsanfrage kann der Auftraggeber sie bestätigen oder löschen. Dem Auftraggeber stehen drei Optionen zur Verfügung; Auftragsverwaltung, E-Mail Verwaltung und Umsatzverwaltung. Zuerst betrachtet man die Auftragsverwaltung. Dort befinden sich 11 Optionen, 9 davon leiten zu eigenen Unteroptionen weiter. Die anderen Zwei sind Speichern und Äbmelden". Durch diese Optionen kann der Auftraggeber die Kundendaten einsehen, editieren, bearbeiten und löschen. Außerdem kann er die dazugehörigen die Nachrichten lesen und löschen. Auch kann er neue und alte Aufträge einsehen und bearbeiten oder löschen.

Abbildung 3.3: Die bisherige Eingabemaske für die Auftragsverwaltung

Wenn die Anfrage des Kunden bestätigt wird, bekommt der Kunde einen PIN per E-Mail. Mit diesem PIN und seiner E-Mail kann der Kunde sich bei "Jungekueche anmelden. Nach dem Login-Prozess, findet er sich auf einer persönlichen Nutzerseite wieder. Dort hat er eine Übersicht über seine aktuellen und alten Bestellungen, ein Kontaktfenster, in dem er die Nachrichten einsehen oder schreiben kann, sowie eine Art Newsletter. Außerdem gibt es eine Option, durch die der Kunde neue Bestellungen aufgeben kann.

Abbildung 3.4: Die bisherige Eingabemaske für die Auftragsverwaltung

## 3.2 Beschreibung der Funktionalität

Einer Anmeldung muss immer eine Registrierung vorausgehen. Diese Funktionalität wird mit ASP und ASP.NET realisiert. Die verwendeten Hilfsprogramme sind JQuery, JavaScript, AJAX, HTML und CSS.

Zuerst wird betrachtet, wie die Kommunikation zwischen den verschiedenen Technologien funktioniert. Das ganze Programm besteht aus Front- und Backend. Das Frontend

bezeichnet den Teil des Programms, welches der Nutzer verwendet. Im Backend werden die Daten verarbeitet. Die Verbindung zwischen den beiden wird über JavaScript und JQuery realisiert. Wenn der Benutzer eine Tätigkeit ausführt, werden die eingegebenen Daten über JavaScript Methoden zu dem Backend weitergeleitet. Dort werden sie verarbeitet und über JavaScript Methode aufgerufen. Dort werden sie bearbeitet und wieder zu dem Frontend zugeschickt. Das Backend entsteht aus ASP-Funktionen, in denen sich verschiedenen Methoden befinden, sowie Datenbanken. In den Datenbanken werden die eingegebenen Daten gespeichert.

Abbildung (neue Diagramm.. die Kommunikation zwischen Front- und Backend bzw. Client -Server)

Nach der grundlegenden Kommunikation zwischen Front- und Backend, wird erläutert wie die einzelnen Kommunikationsabläufe zwischen Front- und Backend koordiniert werden.

### 3.2.1 Anmelde- und Registerformular

#### Hier wurde noch nicht korrigiert

Es wird mit dem Bestellseite angefangen, bzw. Anmeldeformular. Wenn die Kunde "Anmeldung" drückt, wird eine HTML Post-Methode aktiviert. So werden JavaScript- und ASP- Methoden aufgerufen. Die JavaScript-Methoden werden benutzt, damit die Eingabe geprüft wird, ob sie korrekt ist, und ASP-Methoden werden dann aufgerufen.

Der Registerformular funktioniert auf demselben Prinzip. Wenn die Eingaben eingegeben sind, werden sie geprüft, ob alles korrekt ist, falls alles in Ordnung ist, werden die Eingaben zu dem zugeordneten Daten, bzw. Methoden zugeschickt.

mehr im Anhang - Funktionen

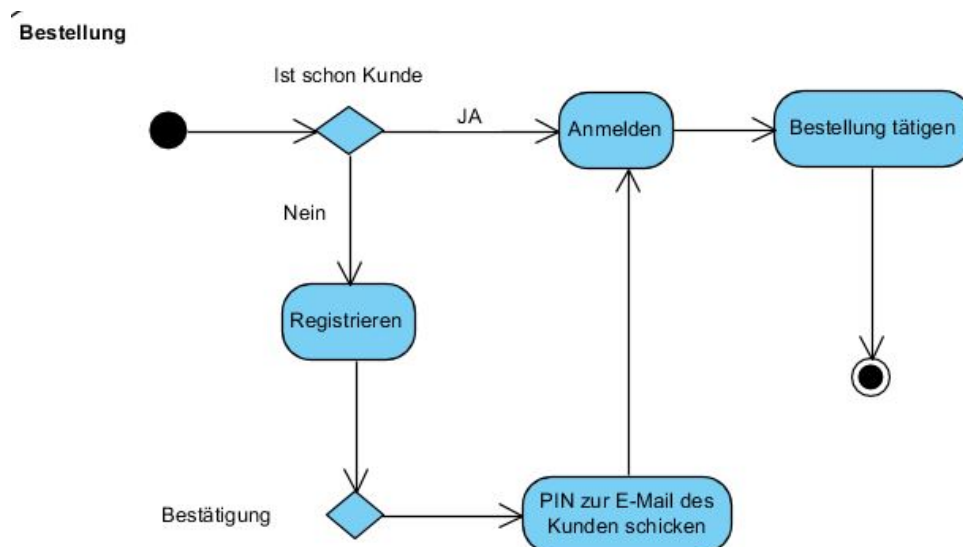


Abbildung 3.5: Die bisherige Eingabemaske für den Vorgang der Bestellung/Registrierung

### 3.2.2 Auftraggeber-Verwaltung

Hier wird man sich nur mit dem Auftragsverwaltung beschäftigen. Die Auftragsverwaltung, wie es schon beschrieben wurde, besteht auf verschiedenen Optionen. Jetzt lässt sich jeweiliges betrachten, wie es funktioniert.

#### 3.2.2.1 Nachricht-Sektion

ermöglicht dem Auftraggeber die Information zu lesen oder löschen. Nach zugeordnetem Wahl wird die Information in der Datenbank "kommentar" entweder aufgerufen oder entfernt.

Neue Nachricht

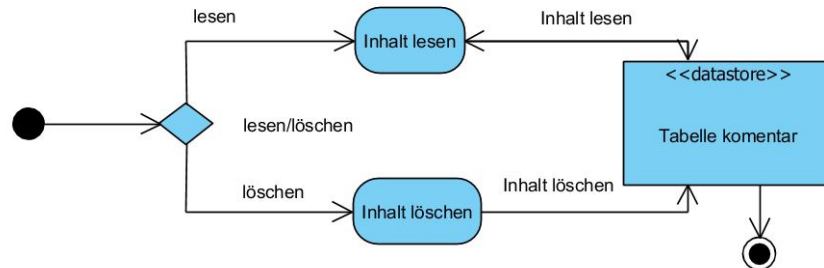


Abbildung 3.6: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

#### 3.2.2.2 Aufträge: Neue, alte und bearbeitende

Die Funktionalität von allen drei ist dieselbe. Deswegen es wird allgemein erklärt. Der Auftraggeber kann den Auftrag sehen, editieren oder löschen. Alle von diese Aktivitäten ist mit mehrere Datenbanken verbunden, die verschiedenen Informationen über die gewählten Artikel haben. In den Abbildungen 3.7 und 3.8 betrachtet man die jeweilige Funktionalität.

Auftragverwaltung

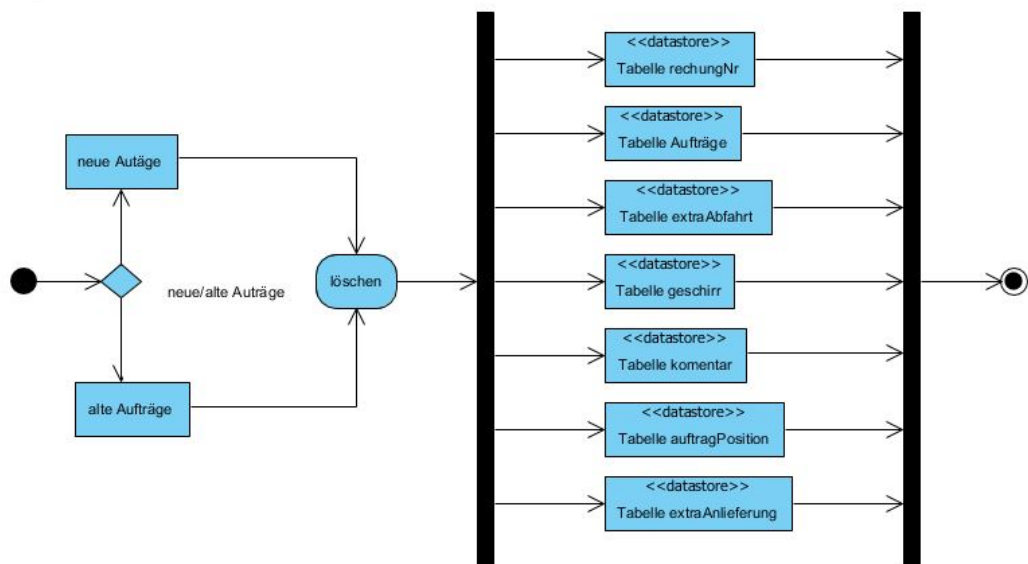


Abbildung 3.7: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

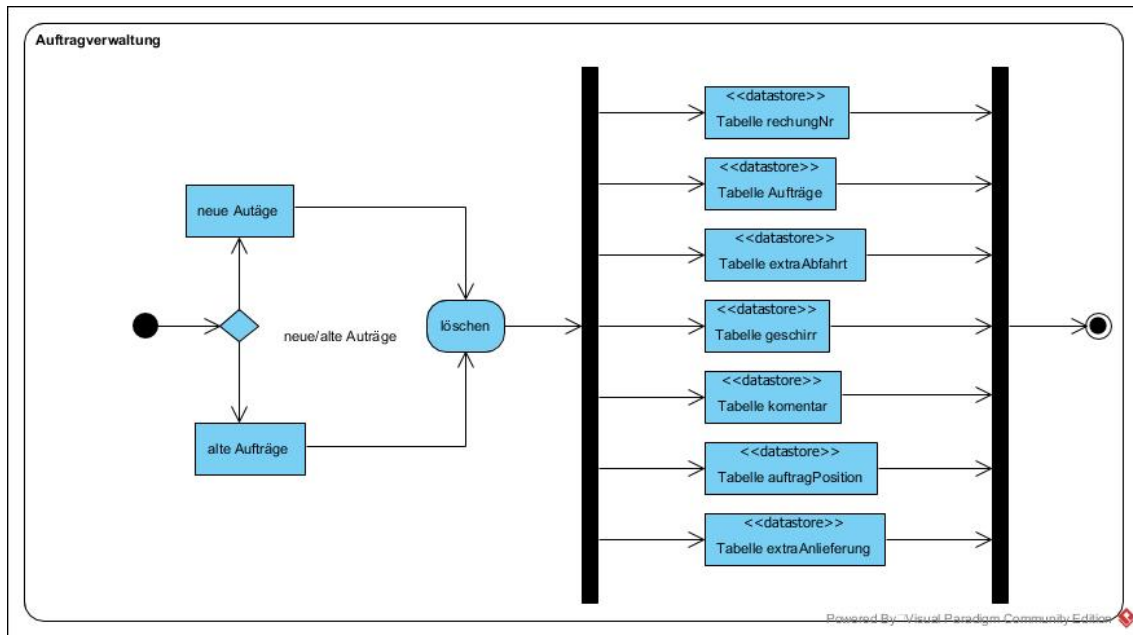


Abbildung 3.8: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

### 3.2.2.3 Kundendatei

Hier befindet sich die Information über die Kunden. Da kann man die Information editieren oder löschen. Um der bestimmten Kunde schneller zu finden, steht eine Suchmaschine zur Verfügung. Eine bessere Übersicht wird in der Abbildung 3.9 dargestellt.

Abbildung 3.9: Die bisherige Eingabemaske für die Kundendatei

Wenn man editieren oder löschen will, wird es über JavaScript Funktion passiert. Diese Funktion ruft die Methoden auf, die sich in der zugeordneten ASP-Datei befindet. Im Editor-Fenster kann man die Daten des Kunden editieren, Login-Daten wie Email an Kunden übermitteln, Wichtige Information für alle Kunden senden, V.I.P-Status des Kunden geben, spezifische Warnungen jeweiligem Kunde aufgeben, Information schreiben, sowie Unterschrift editieren und etc. Besser Verständnis zum diesen Fenster ergibt uns Abbildung 3.10

### 3 IST-Analyse

**Kundendaten**

Name:

Firma:

Adresse:

PLZ:

Ort:

e-Mail:

Telefon:

Handy:

Kunden Pin:

Facebook:

☐ Login-Daten via Email an Kunden übermitteln

**Wichtige Informationen für alle Kunden:**

Bitte beachten (Kundenspezifisch):

Erfahren:

Info:

<input <="" td="" type="button" value="Unterschrift?"/> <td><input <="" td="" type="button" value="Auswahl?"/><td><input 265="" 561="" 578"="" 778="" button"="" data-label="Caption" type="button" value="Geduld&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;div data-bbox="/><p>Abbildung 3.10: Die bisherige Eingabemaske für das Kunden-Editor</p></td></td>	<input <="" td="" type="button" value="Auswahl?"/> <td><input 265="" 561="" 578"="" 778="" button"="" data-label="Caption" type="button" value="Geduld&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;div data-bbox="/><p>Abbildung 3.10: Die bisherige Eingabemaske für das Kunden-Editor</p></td>	<input 265="" 561="" 578"="" 778="" button"="" data-label="Caption" type="button" value="Geduld&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;div data-bbox="/> <p>Abbildung 3.10: Die bisherige Eingabemaske für das Kunden-Editor</p>
--	--	---

Durch JavaScript - Methode wird eine zusätzlichen Möglichkeit zu dem Kunde Nachricht geschrieben. Sie wird aktiviert, wenn der Cursor auf dem bestimmten Kunde, der in der Liste des Adressbuch ist, steht. Die Kommunikation zwischen dem Auftraggeber und dem Kunde wird auch im Datenbank "kommentar" gespeichert.

Nach allem, was geschrieben ist, soll man sich ein vertieftes Verständnis aufbauen. In folgenden Abbildungen 3.11 und 3.12 kann man sehen, wie genau die Aktivitäten des Auftraggeber passieren und wie funktioniert die Nachricht-Kommunikation.

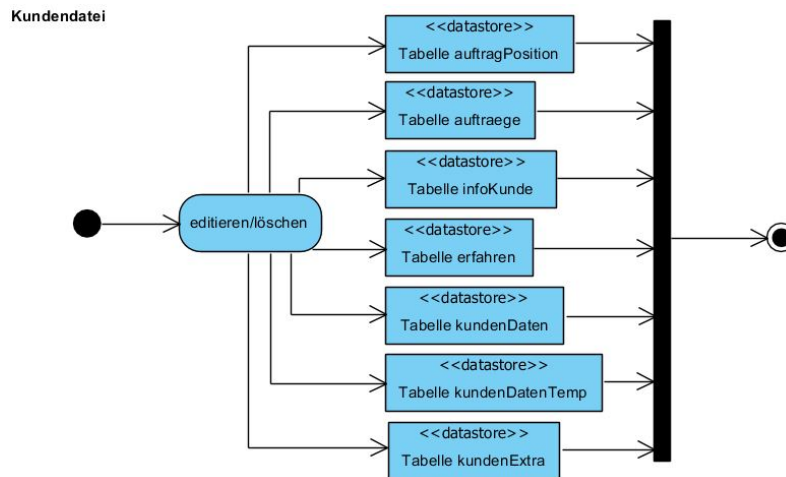


Abbildung 3.11: Die bisherige Eingabemaske für das Editieren/Löschen Funktion

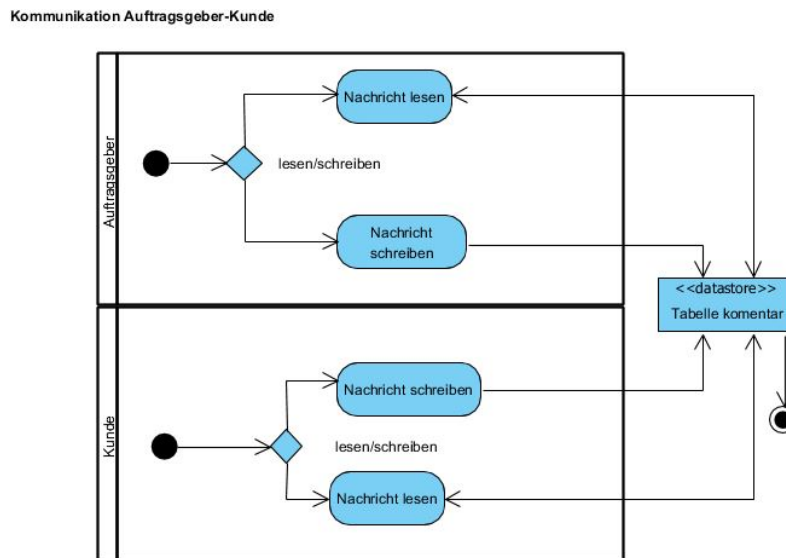


Abbildung 3.12: Die bisherige Eingabemaske für die Kommunikation zwischen dem Auftragsgeber und dem Kunde

### 3.2.2.4 Online-Editor

Durch diese Option kann man die neuen Artikel zu erstellt, editieren und löschen werden können. Es gibt „Artikel – Arragements“ und „Artikel – Standard“. Wenn schon ein Artikel erstellt wurde, kann er entweder gelöscht, editiert oder zugeordnet werden. JavaScript Funktionen werden benutzt, um die Eingabe zu prüfen, ob alles korrekt ist. Wenn alles in Ordnung ist, werden die obengenannten Methoden aus den jeweiligen Dateien aufgerufen. In der folgenden Abbildung 3.13 ist die Übersicht der Seite zu sehen.

### 3 IST-Analyse

**Online Angebot hinzufügen / löschen**

Artikel	Einzelpreis	
<input type="checkbox"/> ** UNPLUGGED ** kalt warmes Gourmet Buffet 26,90 € zzgl. MwSt.	0,00	€
<input type="checkbox"/> ** EVERGREEN ** kalt warmes Gourmetbuffet 25,90 € zzgl. MwSt.	0,00	€
<input type="checkbox"/> ** WIESENGRUND ** kalt warmes Buffet 16,90 € + MwSt.	0,00	€
<b>Standard-Artikel die bei jeder Bestellung vorhanden sind</b>		€
<input type="checkbox"/> Anlieferung - Aufschlag 2015 (siehe Website): 12,00 € zzgl. 19 % MwSt.	14,28	€
<input type="checkbox"/> Anlieferung - Aufschlag siehe Website: 10,00 € zzgl. 19 % MwSt.	11,90	€
<input type="checkbox"/> Anlieferung (bitte beachten Sie die Infos auf der Website) ?? km á 0,80 € =	0,00	€

Artikel - Arrangements	Einzelpreis	
<input type="text"/>	<input type="text"/>	
<b>Angebot hinzufügen</b>		

Artikel - Standard	Einzelpreis	
<input type="text"/>	<input type="text"/>	
<b>Angebot hinzufügen</b>		

Artikel - Arrangements \*\* UNPLUGGED \*\* kalt warmes Gourmet Buffet 26,90 € zzgl. MwSt.

**Editieren**

Artikel - Standard Anlieferung - Aufschlag 2015 (siehe Website): 12,00 € zzgl. 19 % MwSt.

**Editieren**

Abbildung 3.13: Die bisherige Eingabemaske für die Übersicht das Menü

Wenn die Option „Editieren“ zu dem „Artikel-Arrangements“ ausgewählt wird, wird neue Seite geöffnet, in der es vielfältige Möglichkeiten gibt, die Inhalt geändert werden kann. Abbildung 3.14 zeigt uns wie diese Seite aussieht.



### 3.2 Beschreibung der Funktionalität

**buffet / menü - daten editieren**

**\*\* UNPLUGGED \*\***
**kalt warmes Gourmet Buffet**
2€
0,00

Kunde kann wählen

<br><br> 3 warmer Spezialitäten mit Gemüsebuffet und 3 Beilagen<br><br>
1 Gourmetsuppe (zzgl. 1,50 €)<br><br>
11 kalten Spezialitäten, feine Salate der saison,<br><br> 4 Desserts + Kaseauswahl + Brotbuffet + Dipsaucen <br><br>
(Auf Wunsch können Sie sich auch gerne aus dem Buffet Evergreen oder Buffet Saarmania warme Spezialitäten aussuchen, bitte per Telegramm mitteilen)

Änderungen durchführen

**Warme Spezialitäten**

Ganze Kategorie löschen

aa zarte Médallions von Schweinelendchen mit Pfifferlingsahne

editieren  
löschen

ab feiner Wildlachs & Filet vom Saint Pierre in Champagnersauce

editieren  
löschen

**Dipsaucen**

Ganze Kategorie löschen

ai 5 Dips der Saison

editieren  
löschen

**Kalte Spezialitäten**

Ganze Kategorie löschen

aj Roastbeef rosa gebraten mit gefüllten Eiern

editieren  
löschen

aj gegrilltes Poulardencarpaccio mit grünem Spargel und Ananas

editieren  
löschen

**Brotbuffet**

Ganze Kategorie löschen

az Partyrad, verschiedene Flutes und kleine Partyweckchen

editieren  
löschen

**feine Salate**

Ganze Kategorie löschen

ba komplettes Salatbuffet der Saison (siehe Website)

editieren  
löschen

**Dessert**

Ganze Kategorie löschen

bm hausgemachte Panna cotta mit Himbeermus im Glas

editieren  
löschen

bn Weiße Mousse mit exotischen Früchten

editieren  
löschen

**Käse**

Ganze Kategorie löschen

bs Französische Edelkäse mit Trauben und Salzgebäck

editieren  
löschen

**position hinzufügen**

Warme Spezialitäten

hinzufügen

Abbildung 3.14: Die bisherige Eingabemaske für die Übersicht des Menü-Editor-Arrangements

Artikel-Standard“ hat nur eine Möglichkeit, editiert zu werden. Sie wird in der Abbildung 3.15 dargestellt.

<b>Artikel - Standard</b>	<b>Einzelpreis</b>	
Anlieferung - Aufschlag 2015 (siehe Website): 12,00 € zzgl. 19 % MwSt.	14,28	
<b>Angebot editieren</b>		

Abbildung 3.15: Die bisherige Eingabemaske für die Übersicht des Menü-Editor-Standard

In Abbildung 3.16 ist der Zusammenhang zwischen die verschiedenen Funktionen des „Online-Editor“ zu sehen.

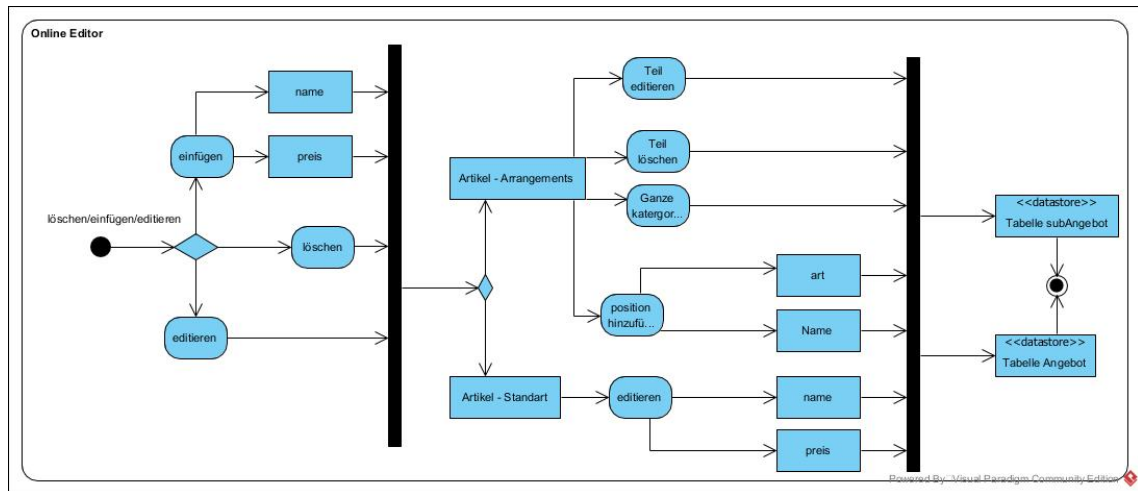


Abbildung 3.16: Die bisherige Eingabemaske für die Funktionalität des Menü-Editor-Standard

## 3.3 Testen

Für diese Arbeit wird ein webbasierten Sicherheit-Checker von 1&1. Nach eine vertiefte Recherche, zeigte sich, dass die Webseite auf eine mittlere Sicherheitsposition steht. Die Webseite ist nach vier Kriterien ohne Beanstandung abgesichert.

- Secure Sockets Layer (SSL) Verschlüsselung – Über SSL wird sichergestellt, dass zwischen User und Server die übertragenen Daten nicht gelesen werden können.
- Cookies sind auch sicher und so ist der Browser von Dritten über JavaScript unlesbar.
- Apache-Status ist verboten. Die Status-Seite ist öffentlich nicht erreichbar. So wird den Schutz von potenziellen Angreifern erhöht.
- Server Version ist nicht sichtbar. Die Server Version ist öffentlich nicht einsehbar. Damit kann ein Angreifer nicht einfach bekannte Schwachstellen ausnutzen.

Wie man von der Abbildung 3.16 sehen kann, ist die Webseite gesichert, aber es gibt einige Bereiche, die eine Optimierung benötigen würde. Sie lädt sich langsam, wird im Browser relativ oft gefunden, aber ist sie gesichert.

Über noch ein Web-Checker werden die weitere Funktionalitäten der Wrbseite analysiert. Observatory[1, postnote] von Mozilla analysiert der Browser im vier Schritten.

- Hypertext Transfer Protocol (HTTP)
- Transport Layer Security (TLS)
- Secure Shell (SSH)
- **Third-Party Tests! (Third-Party Tests!)**

Erste zwei werden betrachtet, weil nur sie wichtig zu dieser Arbeit sind.

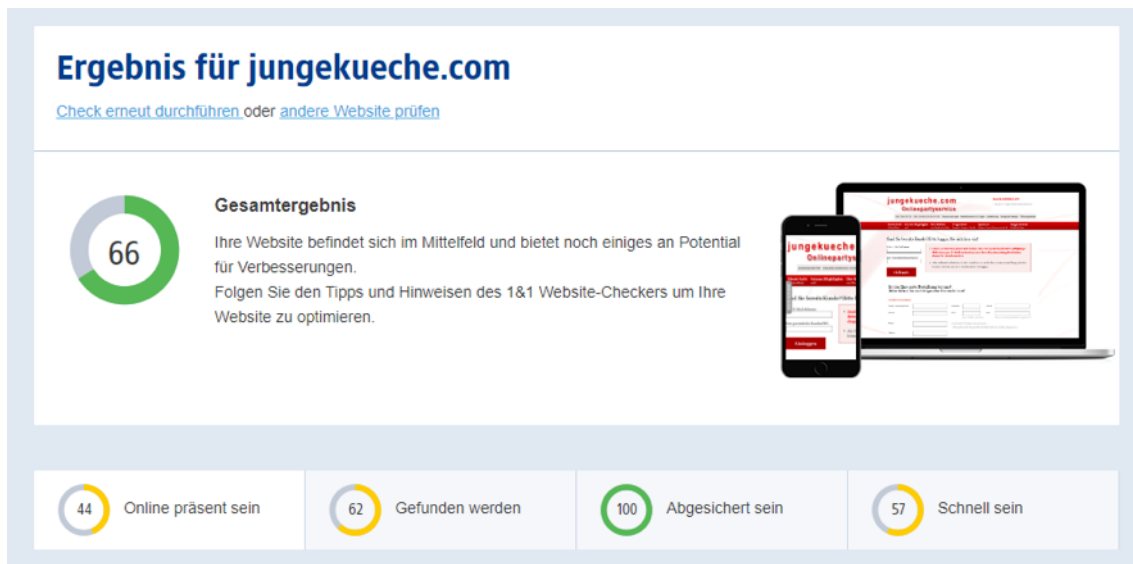


Abbildung 3.17: Ergebnis vom 1zu1

### HTTP Observatory

Hier werden aktuelle Sicherheit der Webseite im öffentlichen Internet beobachtet. Alle Anfragen sind über POST- oder GET-Methoden durchgeführt und die Rückmeldungen sind im JSON-Format. Nach dieser Analyse wird gezeigt, dass **CPS!** (**CPS!**) nicht implementiert wurde. CSP verhindert Weitbereich von Cross-Site-Scripting (XSS)- und „clickjacking“-Attacken. Cookies sind teilweise gesichert. Hier fehlt es „Secure“-Attribut, sowie „SameSite“ und „Prefix“. Durch „Secure“ werden die Cookies davon abgehalten, dass sie über nicht gesicherten HTTP versendet werden. „SameSite“ hält die Cookies vom Cross-Site-Request-Forgery (CSRF) Attacken und Cross-Site-Übersendung ab, d. h. es ist möglich, dass es einen Hacker-Angriff auf Computersystem des Benutzers durchgeführt werden [1, postnote]. Es wird keine \_Host- und \_Secure-Prefixes auf den Namen der Cookies verwendet. Das bedeutet, dass sie nicht gegen Überschreiben geschützt sind. Man kann in der Abbildung 3.18 die genaue Bewertung von der Webseite. stellt das Resultat von der durchgeführten Analyse dar.

Test Scores				
Test	Pass	Score	Explanation	
<a href="#">Content Security Policy</a>	✗	-25	Content Security Policy (CSP) header not implemented	Ⓜ
<a href="#">Cookies</a>	✗	-40	Session cookie set without using the <code>Secure</code> flag or set over HTTP	Ⓜ
<a href="#">Cross-origin Resource Sharing</a>	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers	Ⓜ
<a href="#">HTTP Public Key Pinning</a>	–	0	HTTP Public Key Pinning (HPKP) header not implemented (optional)	Ⓜ
<a href="#">HTTP Strict Transport Security</a>	✗	-20	HTTP Strict Transport Security (HSTS) header not implemented	Ⓜ
<a href="#">Redirection</a>	✓	0	Initial redirection is to HTTPS on same host, final destination is HTTPS	Ⓜ
<a href="#">Referrer Policy</a>	–	0	Referrer-Policy header not implemented (optional)	Ⓜ
<a href="#">Subresource Integrity</a>	✗	-5	Subresource Integrity (SRI) not implemented, but all external scripts are loaded over HTTPS	Ⓜ
<a href="#">X-Content-Type-Options</a>	✗	-5	X-Content-Type-Options header not implemented	Ⓜ
<a href="#">X-Frame-Options</a>	✗	-20	X-Frame-Options (XFO) header not implemented	Ⓜ
<a href="#">X-XSS-Protection</a>	✗	-10	X-XSS-Protection header not implemented	Ⓜ

Abbildung 3.18: Ergebnis vom HTTP Observatory

### TLS

### 3 IST-Analyse

Die betrachtenden von uns Webseite verwendet Signaturverfahren SHA-256-With-RSA. Das bedeutet, dass eine Kombination zwischen SHA256 (kryptologischen Hashfunktion, die Hashwerte mit einer Länge zwischen 256 und 512 Bits erzeugt) und Rivest-Shamir-Adleman (RSA) (ermöglicht symmetrische Schlüssel jeder üblichen Länge verschlüsselt zu werden). In der Abbildung 3.19 stellt dar



Certificate Information	
Common name:	jungekueche.com
Alternative Names:	
First Observed:	2018-08-06 (certificate #187995580)
Valid From:	2018-07-27
Valid To:	2018-10-25
Key:	RSA 2048 bits
Issuer:	Let's Encrypt Authority X3
Signature Algorithm:	SHA256WithRSA

Abbildung 3.19: Ergebnis vom TLS Observatory

### 3.4 Warum ist eine Migration notwendig?

ASP ist eine nicht Objekt-Orientierte Skripte Sprache. Es ist anfällig zu Unterbrechung der Applikationen, z. B. wenn Internet Information Services (IIS) gestoppt und neu gestartet wird. Wenn eine ASP-Seite aufgerufen wird, wird der Text linear analysiert. Die Frontend-Verwendung des Bestellsystems ist verschachtelt und es gibt redundante Optionen. Auf einer Seite befindet sich Login- und Registrierung Menü, sowie das Bestellformular. Das verlangsamt die Eröffnung der Seite. Es gibt unnötigen Quellcode, der entfernt werden kann. Ein anderer Nachteil des Systems ist, dass es nicht dynamisch orientiert ist, d.h. die Umgebung ist nicht Benutzerfreundlich. Der Auftragsgeber muss sich für jede nötige Änderung bei dem Entwickler melden. Das macht den Benutzer abhängig von dem Hersteller des Programms. Heutzutage gibt es vielfältige Auswahl von CMS, durch die der Auftragsgeber mit verbreiteten Möglichkeiten zu Änderungen seiner Website verfügt. Im folgenden Kapitel wird einer dieser CMS erläutert.

## 4 Anforderungsanalyse

In der durchführenden Analyse werden wir uns mit ein neues Bestell- und Verwaltungssystem Konzept beschäftigen. Auf der Basis der IST-Analyse wird es eine Verbesserung mit aktuellen Webtechnologie und Verfahren durchführen. In diesem Kapitel wird man die entstehenden Probleme und Optimierungspotenzial näher anschauen. Das neue Bestellsystem soll nicht kompliziert sein, sondern Benutzerfreundlich. Die Optionen sollen klar gezeigt werden und einfach zu sein. Sicherheit ist einen wichtigen Teil der guten webbasierte Entwicklung, deswegen werden zusätzliche Prüfungen in dieser Richtung entwickelt. Die Auftragsverwaltung und Kundenübersicht werden mit denselben Funktionen bleiben. Das ganze Konzept wird aus einer Umbraco-Instanz heraus verwaltet. In den folgenden Unterkapitel werden wir uns mit einer detaillieren Erweiterung von oben beschriebenen Anforderungen beschäftigen.

### 4.1 Kundenverwaltung

In diesem Unterkapitel werden wir uns mit den Anforderungen beschäftigen, die Auftragsgeber gewünscht hat, um er die alle Aktivitäten zum Kunde kontrollieren zu können. Eine eigene Seite wird auch erstellt, in der den Kunde seine Bestellungen erstellen und einsehen kann, sowie Nachrichten zuschicken oder lesen kann.

#### 4.1.1 Kundenerfassung

Hier werden wir die Anforderungen zu den Onlinebestell-System und Kunden Übersicht beschreiben.

1. Der Kunde kann sich bei der Erstbestellung registrieren und bekommt eine PIN zugeschickt. Bei jeder weiteren Bestellung kann er sich mit seiner E-Mail und der PIN einloggen. Das was beachtet werden muss ist, dass beim Bestellen die Mail geprüft werden muss und ein Hinweis ausgegeben werden soll, wenn die Mail schon existiert. Neukunden kommen in der Bestellübersicht vom Auftraggeber in eine gesonderte Übersicht und müssen von ihm bestätigt oder übernommen werden. Erst dann kann sich der Kunde auch einloggen.

#### 4.1.2 Kundenansicht

1. Der Kunde kann sich bei der Erstbestellung registrieren und bekommt eine PIN zugeschickt. Bei jeder weiteren Bestellung kann er sich mit seiner E-Mail und der PIN einloggen. Das was beachtet werden muss ist, dass beim Bestellen die Mail geprüft werden muss und ein Hinweis ausgegeben werden soll, wenn die Mail schon existiert. Neukunden kommen in der Bestellübersicht vom Auftraggeber in eine gesonderte Übersicht und müssen von ihm bestätigt oder übernommen werden. Erst dann kann sich der Kunde auch einloggen.
2. Eine flexible Gestaltung für den Auftraggeber muss auch anwesend sein. So kann er hier auch Informationen platzieren, sowohl allgemein als auch kundenspezifisch.

3. Die Registrierung muss über die Member in Umbraco abgebildet werden

### 4.1.3 Auftragsgeber-Ansicht

1. Der Auftraggeber hat eine Kundenübersicht, die er filtern kann. In der Detailansicht sieht er die Infos zum Kunden wie Bestellverlauf und kann von der Ansicht heraus E-Mails an den Kunden senden (Mail-Vorlage oder frei gestaltbar).

### 4.1.4 Kommunikation

1. Über die Option „neue Nachrichten“ kann der Auftraggeber mit seinen Kunden kommunizieren und Absprachen zu den Aufträgen treffen.
2. Eine übersichtlichere Kommunikationsmethode muss mit Filtern (gelesen, nach Kunden suchen usw.) integriert werden. Der „Chat“ sollte auch aus der Kundenkartei-Ansicht funktionieren und die Kommunikation soll an einen Auftrag gebunden sein.

## 4.2 Artikelverwaltung

### 4.2.1 Artikel erfassen, ändern und löschen

1. Der Auftraggeber kann in einer recht einfachen Maske Artikel verwalten (online-editor). Es gibt nur zwei Kategorien: Arrangements und Artikel-Standard. Im Moment sind die Arrangements noch nicht mit den Webseiten, die sie beschreiben, gekoppelt.
2. Ein Arrangement besteht aus Kategorien (fest vorgegeben) und Positionen und kann sich darin unterscheiden, ob Kunden Positionen auswählen dürfen oder nicht.
3. Umsetzungsvorgabe: Artikel sollen einfacher verwaltet werden können und die Artikelseite der Webseite soll direkt mit dem Artikel gekoppelt sein.

## 4.3 Auftragsverwaltung

### 4.3.1 Übersicht

1. Der Auftraggeber hat eine Übersicht über seine aktuellen Aufträge. Für jede Art von Auftrag (neu, bearbeitet, alte und fällig) gibt es ein Select-Befehl.
2. Diese Ansicht soll in einer eigenen Umbraco-Section umgesetzt werden. Die Artikeldatenbank soll von Access nach SQL transportiert werden. Es muss eine neue Zuordnung Kunde zu Umbraco-Member geben.

### 4.3.2 Detailansicht

1. In der Detailansicht kann der Auftragsgeber den Auftrag bearbeiten, den Status ändern, Positionen editieren / hinzufügen und löschen, dem Kunden Freigaben erteilen (z.B. Aussuchen der Positionen) und eine Rechnungsnummer vergeben.
2. Der Auftrag muss ausdrückbar auf vier Seiten ausdrückbar (genau vorgegeben) sein.

3. Die Begrifflichkeiten von zweitem Blatt müssen für Auftragsgeber zu Verfügung gelassen werden, damit er sie selbst ändern kann.
4. Quick-Icons erleichtern die Kommunikation mit dem Kunden und die Anpassung an dessen Auswahl zu Auftragsbeginn (z.B. Änderung Serviceauswahl).
5. Diese Ansicht soll in einer eigenen Umbraco-Section umgesetzt werden.

#### 4.4 E-Mail-Verwaltung

1. Hier kann der Auftraggeber E-Mail Vorlagen mit Platzhaltern definieren, die er dann in der Kundenkommunikation auswählen kann.
2. Es gibt Terminanfragen über die Webseite. Diese erzeugen eine Mail an den Auftraggeber. Der muss in der Mail nur einen Link betätigen, um die Anfrage zu bestätigen. Dies hängt auch mit dem E-Mail Verwaltungssystem zusammen.
3. Diese Ansicht soll in einer eigenen Umbraco-Section umgesetzt werden.

#### 4.5 Umsatzverfassung

1. Hier kann der Auftraggeber die Umsätze der letzten Monate / Jahre sich anschauen, die über die Aufträge zustande gekommen sind. Dabei ist es wichtig, dass diese Datensätze nicht direkt an die Aufträge gekoppelt sind, sondern aus einer extra Tabelle kommen, die der Auftraggeber auch selbst noch editieren kann.
2. Diese Ansicht soll in einer eigenen Umbraco-Section umgesetzt werden.





## 5 Konzeption und Implementierung

Das neue Konzept, das wird im diesen Kapitel erklärt, wird auf zwei großen Paketen unterteilt – Paket A und Paket B. Im ersten Paket wird dem Auftragsgeber User Interface (UI) aufgebaut. Das Ziel ist maximale Flexibilität des Website-Besitzers zu haben und durch einfache Tätigkeiten, bestimmten Zweck zu erreichen. Man kann editieren, ändern und Text stilisieren. Umbraco verfügt mit sogenannte „Grid“. Es dient dafür, dass man das Design der Seite manipulieren und auch die schon besprochenen Möglichkeiten ausnutzen kann. Es werden auch eigene Macros benutzt, in den ein Quell-Code der SHOP – Komponenten hingeschrieben wird. Somit kann der Auftragsgeber Macros in beliebigen Teilen der Seiten hinstellen. Umbraco – Forms werden benutzt als Formulare, damit man selber einordnen kann. Beide Paketen werden als Startknoten aus einer Umbraco-Instanz heraus verwaltet. Paket B ist als Kern der Seite dargestellt. Das ist eigentlich das Online-Bestellsystem. Hier ist es wichtig, dass eine unkomplizierte, bedienbare, reibungslose und flexibel Umgebung aufgebaut wird. Das System besteht aus drei Hauptkerne:

- Bestellsystem
  - Kundenverwaltung (Anmeldung, Kundenbereich, Kommunikation)
  - Artikelverwaltung
  - Auftragsverwaltung
- E-Mail-Verwaltung
  - E-Mail-Vorlagen anlegen, editieren, löschen
- Umsatzerfassung
  - Umsatzübersicht nach Monat und Jahr

### 5.1 Aufbau vom Umbraco

Wie oben schon erklärt wurde, ist Umbraco ein Content Management System, das flexibel und Benutzerfreundlich ist. Damit man bessere Verständnis hat, worum es geht, wird es im nachträglichen Unterkapitel erläutern.

User Interface von Umbraco ist auf drei Teilen verteilt. Erste Teil ist Hauptfunktion bzw. Setcion bezeichnet. Da befinden sich die Hauptoptionen: Content, Media. Settings, Developer, Users, Members, Forms. Damit einen klaren Unterschied zwischen „Member“ und „User“ gemacht wird, werden die beiden Begriffe erklärt. „User“ ist jemand, der Zugriff zu „Umbraco-Backoffice“ hat und dort hat bestimmte Rechte.

Ein Member wird von Umbraco für die Registrierung und Authentifizierung eines externen Besuchers benutzt. Das sind Leute, die nur Front-End benutzen dürfen. Man kann auch „Custom Setcion“ erstellen, womit wir uns im weiteren Kapitel beschäftigen.

Nächste Teil ist Unternavigation oder auch Tree genannt. Alle Unteroptionen stehen da. Jede Hauptfunktion hat ihre eigenen Unteroptionen. Es wird zum jeweiligen Setcions zugehörige Funktionalität der Unteroptionen (Trees) durchgehen.

- **Trees vom Content-Section:** In diesem Tree befinden sich alle Seiten, die im Website Front-End er-schienen werden können. Da steht auch Recycle Bin, oder auch Papierkorb genannt. So kann man die gelöschte Seite zurücksetzen.
- **Trees von der Media-Section:** Hier können alle Videos und Bilder stehen
- **Trees vom Settings-Section:** Mithilfe von (PartialView), HTML, CSS, JavaScript, Dokument- und Medientypen, das Erstellen und das Einsetzen von Templates und Spracheinstellungen werden die Webseiten einfach und schnell eingerichtet.
- **Trees vom Developer-Section:** Hier stehen zur Verfügung Trees, die es den Entwickler ermöglichen bereits erstellte Seiten zu entwickeln. Das wird durch Data Typ, Macros, Packages, Relation Types, XSLT Files und Partial View Macro Files ermöglicht.
- **Trees vom User:** In diesem Bereich stehen die Benutzer, die mit bestimmtem Rechten vom Umbraco die Front-End der Webseite benutzen dürfen. Zu einem bestimmten User können verschiedene Rechte abgegeben werden.
- **Trees vom Member-Section:** Hier sind Benutzer, die sich in der fertigen Webseite registriert haben.
- **Trees vom Forms-Section:** Hier kann man leicht verschiedene Arten von Formularen erstellen.

Der dritte Teil vom Umbraco-Backoffice ist der Editierbereich. Dort werden alle Eigenschaften von jeweiligen Tree-Optionen dargestellt.

Im Abbildung 5.1 kann man sehen, wie Umbraco-Backoffice aussieht.

### Aufbau des Backends

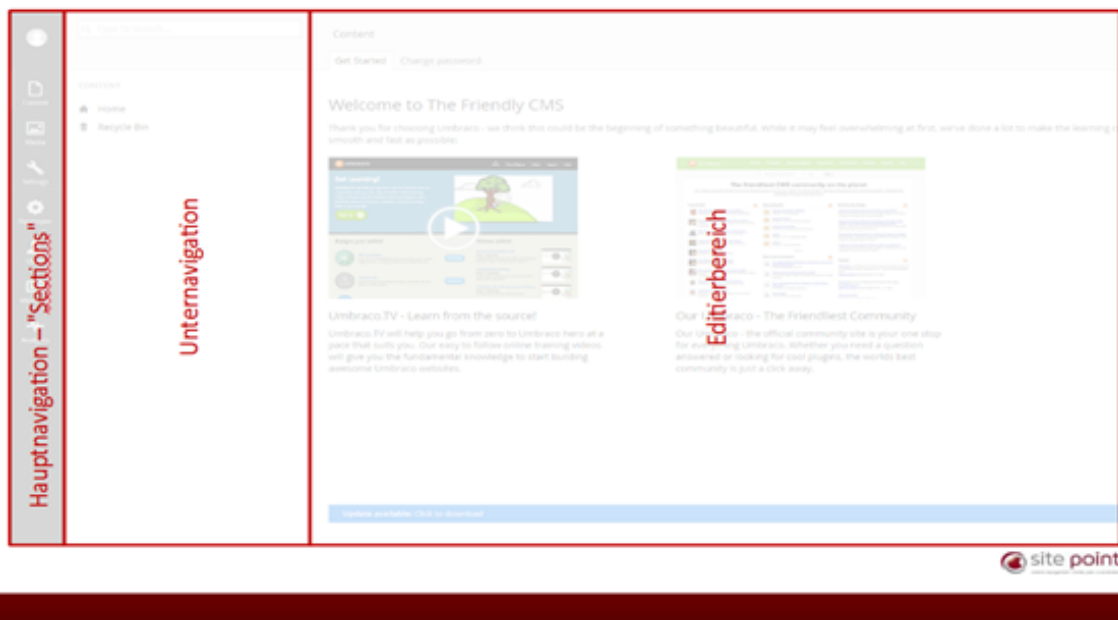


Abbildung 5.1: Übersicht vom Umbraco-Backoffice

## 5.2 Kundenverwaltung

Die Kundenverwaltung fasst den ganzen Bereich um, der sich um die Kunden bezieht. Mehr wird in nächsten Unterkapitel erfahren.

### 5.2.1 Kundenerfassung

1. Das angestrebte Ziel ist, dass der Kunde sich registrieren kann. Die Registrierung muss gesichert und individuell sein. Damit die Registrierung zuverlässig ist, muss nach den folgenden Regeln gehalten werden: Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität. Ein Personal-Identification-Number (PIN) lässt sich in E-Mail des Kunde zusenden. So wird sichergestellt, dass die Kunde eine gültige E-Mail hat. Nach der Registrierung wird überprüft, ob die E-Mail schon existiert. Wenn ja, das Prozess wird storniert und eine Nachricht erscheint, in der geschrieben ist, dass der Kunde diese E-Mail schon verwendet hat. Der Auftragsgeber entscheidet, ob der sich registrierten Kunde bestellen darf oder nicht. Danach kann der Benutzer sich mit dem PIN und die E-Mail anmelden. Die Kundenverfassung wird durch Member Application Programming Inter-face (API) von Umbraco realisiert. Wie schon erläutert wurde, ist Umbraco auf ASP.NET basiert. Damit kann man vom CMS unterschiedliche „Services“ benutzen. Ein davon ist „MemberService“. Das ist direkte Kanal zu dem MemberAPI. Es ist in den „Services property“ von dem „SurfaceController“ zu Verfügung gestellt. Die Registrierung wird via ASP.Net Code und Media API programmatisch erstellt. Das ist eine komplexe Methode, in der es eine vielfältige Datei benötigt wird (SurfaceController, Model, PartialView und Source Datei, wo Front-End Source code steht). Die benötigte Information, die wir zu der Registrierung brauchen, wird im „Model“- Datei geschrieben. In der Datei „Model“ stehen „Model Properties“. Das sind Parametern, mit denen man arbeitet. Einfach erklärt, über das Model werden die Properties von „Partial-View“ zu dem „SurfaceController“ übertragen. „SurfaceController“ ist der „Autobahn“ zu der Umbraco-Datei. Das ist ein MVC (Model View Controller), das mit dem Umbraco interagiert wird. Es wird von der Bibliothek Umbraco.Web.Mvc.SurfaceController geerbt. Über „PartialView“ werden die Verbindungen zwischen Kontakt Formular und „Model Properties“ geschehen. Das ist eigentlich eine Teilansicht, die von Umbraco Front-End benutzt wird. Dort ist Umbraco User Interface (UI). Folgende Abbildung ergibt bessere Verständnis wie die obengenannten Begriffe zu einander stehen.



# Literatur

- [1] Jon Bentley. *Programming Pearls*. Addison–Wesley, 1999.
- [2] Donald E. Knuth. „Computer Programming as an Art“. In: *Communications of the ACM* 17.12 (1974), S. 667–673.



## Abbildungsverzeichnis

3.1	Anmeldeformular . . . . .	7
3.2	Registerformular . . . . .	7
3.3	Auftragsverwaltung . . . . .	8
3.4	Kundeansicht . . . . .	8
3.5	Anmeldung/Bestellung . . . . .	9
3.6	Kommunikation . . . . .	10
3.7	AutragLoeschen . . . . .	10
3.8	auftragEinsehen . . . . .	11
3.9	Kundeansicht . . . . .	11
3.10	Kundeansicht . . . . .	12
3.11	Kundeansicht . . . . .	13
3.12	Kundeansicht . . . . .	13
3.13	Kundeansicht . . . . .	14
3.14	Kun2deansicht . . . . .	15
3.15	Kundeansicht . . . . .	15
3.16	Kundeansicht . . . . .	16
3.17	Egebniss Von 1zu1 . . . . .	17
3.18	Egebniss vom HTTP Observatory . . . . .	17
3.19	Egebniss TSL Observatory . . . . .	18
5.1	Umbraco Backoffice . . . . .	24

## Tabellenverzeichnis

## Listings





# Abkürzungsverzeichnis

<b>EDV</b>	Elektronische Datenverarbeitung
<b>WIC</b>	Windows Imaging Components
<b>ASP</b>	Active Server Pages
<b>HTML</b>	Hypertext Markup Language
<b>WWW</b>	World Wide Web
<b>GUI</b>	Graphical User Interface
<b>CSS</b>	Cascading Style Sheets
<b>CMS</b>	Content Management System
<b>HTTP</b>	Hypertext Transfer Protocol
<b>TLS</b>	Transport Layer Security
<b>SSH</b>	Secure Shell
<b>SSL</b>	Secure Sockets Layer
<b>XSS</b>	Cross-Site-Scripting
<b>CSRF</b>	Cross-Site-Request-Forgery
<b>RSA</b>	Rivest–Shamir–Adleman
<b>IIS</b>	Internet Information Services
<b>PIN</b>	Personal-Identification-Number
<b>API</b>	Application Programming Inter-face



# Anhang



## A Erster Abschnitt des Anhangs

In den Anhang gehören „Hintergrundinformationen“, also weiterführende Information, ausführliche Listings, Graphen, Diagramme oder Tabellen, die den Haupttext mit detaillierten Informationen ergänzen.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.



## **Kolophon**

Dieses Dokument wurde mit der L<sup>A</sup>T<sub>E</sub>X-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 2.1). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer, Helmut G. Folz und Martina Lehser). Daten: (F)10.95 – (B)426.79135pt – (H)688.5567pt