

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

1	Einleitung	1
2	Genutzte Technologie und Services	3
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
3	IST-Analyse	7
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	12
	Literatur	13
	Abbildungsverzeichnis	15
	Tabellenverzeichnis	15
	Listings	15
	Abkürzungsverzeichnis	17
A	Erster Abschnitt des Anhangs	21

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 Pendanten. 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

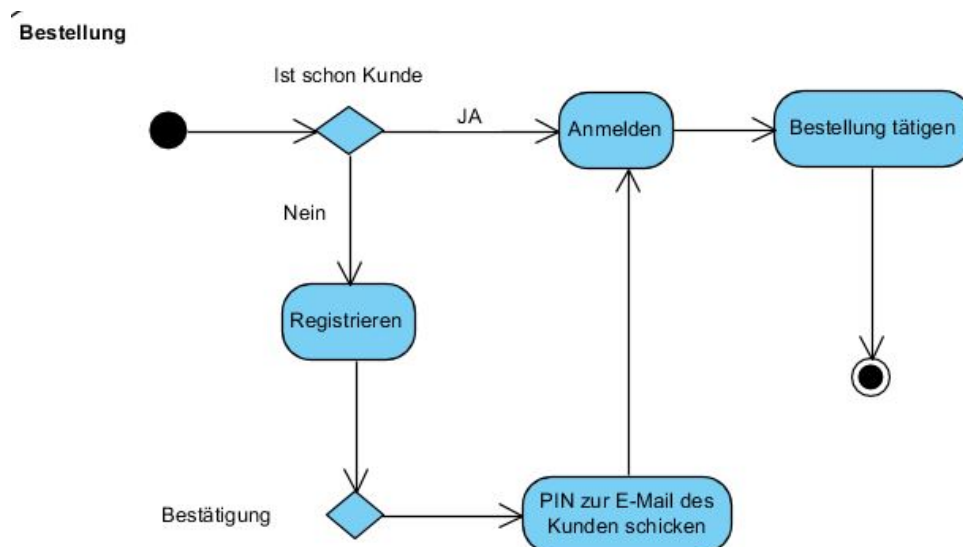


Abbildung 3.5: Die bisherige Eingabemaske für den Vorgang der Bestellung/Registrierung

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

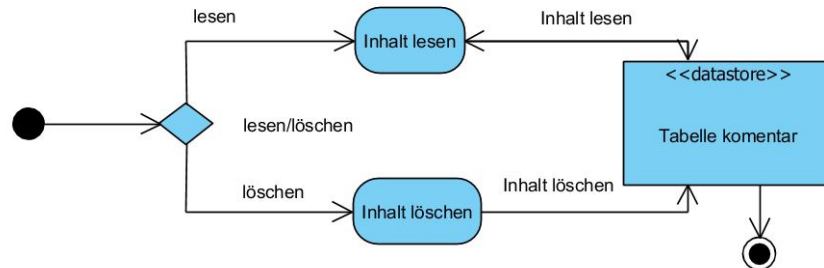


Abbildung 3.6: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

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

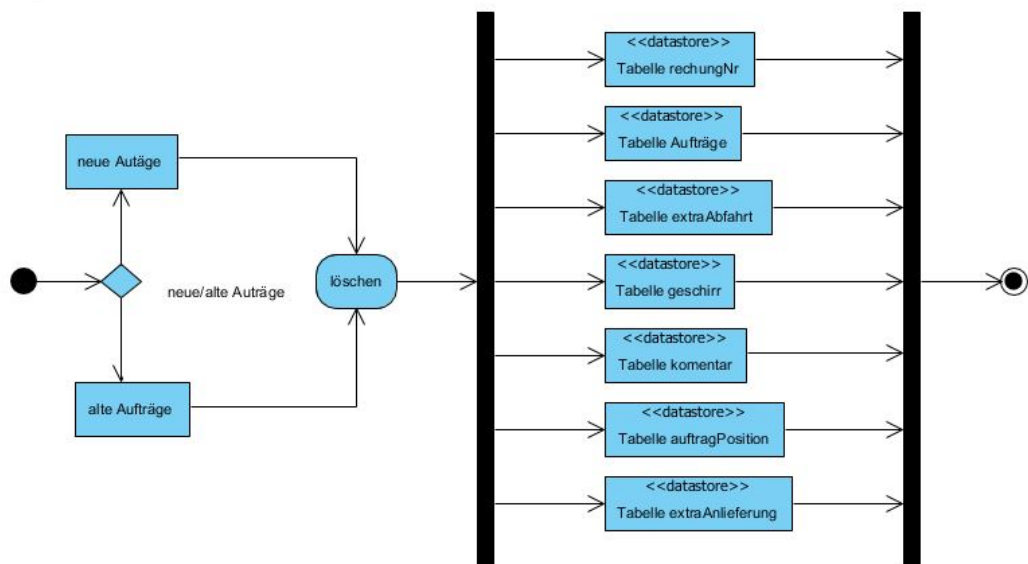


Abbildung 3.7: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

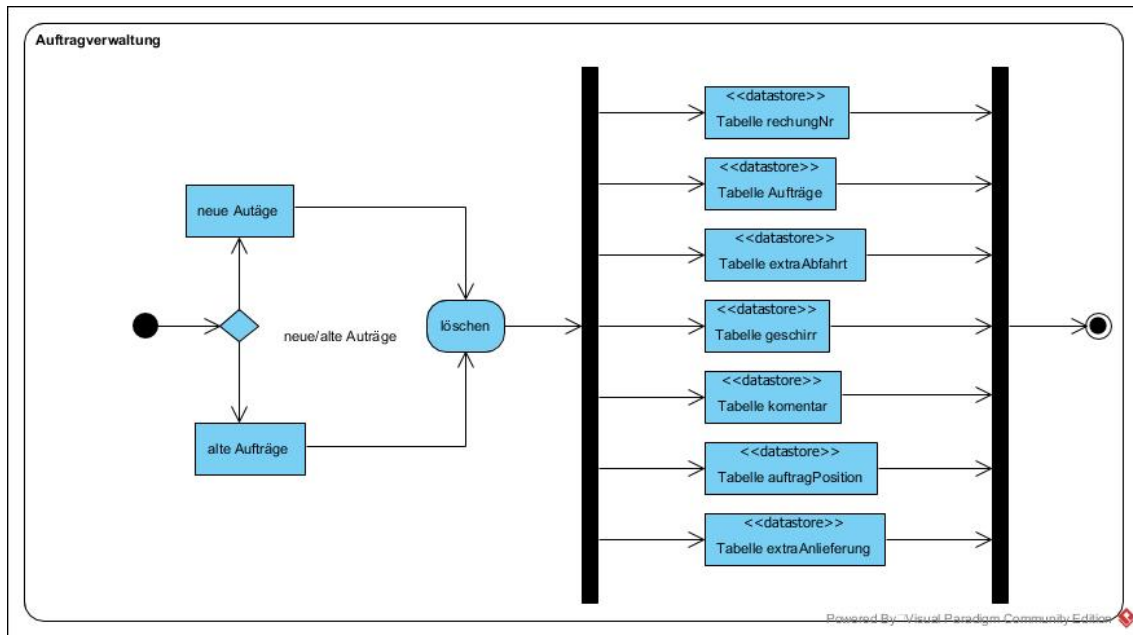


Abbildung 3.8: Die bisherige Eingabemaske für den Nachrichten lesen oder löschen

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 stellt uns die Abbildung dar.

Abbildung Adressbuch

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 3-19

Abbildung übersicht von Editor-Fenster

Durch JavaScript - Methode wird eine zusätzlichen Möglichkeit zu dem Kunde Nachricht geschrieben zu werden. 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.

Abbildung Neue Nachricht

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

Abbildung Kundendatei

Abbildung Auftraggeber-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 – Arrangements“ und „Artikel – Standard“. Wenn schon ein Artikel erstellt wurde, kann er entweder gelöscht, editiert oder zugeordnet werden. JavaS-

cript 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 ist die Übersicht der Sete zu sehen.

Abbildung Online Übersicht

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 zeigt uns wie diese Seite aussieht.

Abbildung

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

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

Abbildung Online-Editor

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.

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

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

EDV Elektronische Datenverarbeitung

WIC Windows Imaging Components

ASP Active Server Pages

HTML Hypertext Markup Language

WWW World Wide Web

GUI Graphical User Interface

CSS Cascading Style Sheets

CMS Content Management System

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 \LaTeX -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