

Hochschule Ostwestfalen-Lippe
Fachbereich 8 - Umweltingenieurwesen und Angewandte Informatik
Fachgebiet Angewandte Informatik
Projekt Realisierung betrieblicher IT-Systeme
4. Semester SoSe 2015

Ausarbeitung:
Projekt Realisierung betrieblicher IT-Systeme

**Entwicklung eines Webshops
für das Fahrradproduktions- und Vertriebsunternehmen
Global Bike Inc.**

Von
Stefan Schnürer,
Benedikt Brüntrup,
Raphael Gregarek und
Michael Dück

Erstprüfer: Prof. Dr. rer. nat. Stefan Wolf
Zweitprüfer: Prof. Dr. Ralf Hesse
Drittprüfer: Prof. Dr. Jessica Rubart
Vierprüfer: Prof. Dr. rer. nat. Burkhard Wrenger

Eingereicht am: 2. September 2015

Inhaltsverzeichnis

1 Projektaufgabenstellung	3
2 Projektplanung	3
2.1 Ideenfindung	3
2.2 Anforderungsanalyse	4
2.3 Geschäftsprozess	4
2.4 Risikoanalyse	6
2.5 Wahl der Programmiersprache	7
2.6 Wahl der Programmierentwicklungsumgebung und des Versions-verwaltungs-Plugin	7
2.7 Grundlegender Aufbau und Ordnerstruktur der Webseite	8
2.8 Die Test-Server	10
2.9 Zeitplan	10
3 Arbeitspakete	17
3.1 Einteilung der Arbeitspakete	17
3.2 Arbeitspaket von Herrn Schnürer	19
3.2.1 Arbeitspaket 1: Erster Seitenprototyp	19
3.2.2 Arbeitspaket 1 und 4: Startseite des Webshops	20
3.2.3 Arbeitspaket 3: Seitennavigation	20
3.2.4 Arbeitspaket 4: Weitere Arbeiten an der Startseite des Webshops	23
3.2.5 Arbeitspaket 4: Rechtsschutz	24
3.2.6 Arbeitspaket 3: Dynamische Seitennavigation	26
3.2.7 Arbeitspaket 4: Dynamische Artikellaufistung	29
3.2.8 Arbeitspaket 4: Detaillierte dynamische Artikelbeschreibung	30
3.2.9 Arbeitspaket 4: Startseite-Angebot des Tages	33
3.3 Arbeitspakete von Herrn Brüntrup	34
3.3.1 Arbeitspaket 5: Die Schnittstellendatenbank	34
3.3.2 Arbeitspaket 6: Registrierungs- und Anmeldungsfenster designen und programmieren	40
3.3.3 Arbeitspaket 7: Entwickeln einer Suchfunktion	48
3.3.4 Zum Teil übernommenes Arbeitspaket: Warenkorb	50
3.3.5 Zum Teil übernommenes Arbeitspaket: Bestellvorgang	53
4 Sicherheit	55
4.1 „.htaccess“-Datei	56
4.2 Schutz vor: Cross-Site-Scripting (XSS) und SQL Injection	56

5 Fazit	57
5.1 Motivation innerhalb der Gruppe	57
5.2 Zusammenarbeit mit anderen Gruppen	57
5.3 Verlauf des Projektes	57
6 Quellenverzeichnis	57

1 Projektaufgabenstellung

Die Aufgabenstellung des Projektes ist es einen Webshop für die Firma Global Bike Inc. (folgend GBI) zu erstellen. Diese Firma stellt Fahrräder her und vermarktet dieses. Die Produkte des Webshops sollen aus einem ERP-System geladen werden. Bei dem ERP-System handelt es sich um ein schon bestehendes SAP-ERP-System. Alle auf dem Webshop getätigten Bestellungen sollen an das ERP-System weitergereicht werden. Zudem soll für den Benutzer des Webshops der Stand der Bestellung einsehbar sein. Dieser Stand soll ebenfalls aus dem ERP-System übernommen werden.

Weitere Kriterien für den Webshop sind, dass dieser in eine APP für mobile Geräte einbindbar sein muss, ein Corporate-Design entworfen werden, sowie ein Marketing-Mix für die Online-Vermarktung im Großraum OWL entwickelt werden soll.

2 Projektplanung

Dieser Abschnitt erläutert, wie die Projektplanung dieses Projekts ablief. Es wird der Ablauf von der Ideenfindung bis durch Planung der Durchführung erläutert.

2.1 Ideenfindung

Der erste Schritt des Projekts bestand darin zu bestimmen, was für Funktionen der Webshop alles anbieten soll. Hierfür sind drei Onlinewebshops betrachtet worden. Es wurden die Webshops „fahrrad.de“, „amazon.de“ und „alternate.de“ betrachtet.

Vom Webshop „alternate.de“ ist der Webseitenkopf als sehr sinnvoll empfunden worden. Dieser Kopf ist am oberen Rand der Webseite fest fixiert und wird nicht mitgescrollt. Dieses bietet den Vorteil, dass der Kunde durchgehend die Menge einsehen kann, die sich im Warenkorb befindet. Grund hierfür ist, dass sich eine kleine Warenkorbanzeige mit im Kopf der Webseite befindet. Zudem kann der Benutzer sich durch die Webseite navigieren ohne zu scrollen. Da ebenso im Kopf der Webseite die Suche fixiert ist, kann der Benutzer auch ohne zu scrollen nach etwas suchen. Wegen den vielen Vorteilen wurde beschlossen den Kopf der Webseite weitgehend für den Webshop der GBI zu übernehmen.

Aus allen drei Webshops wurde zudem übernommen, dass der Webshop der GBI mit einem Warenkorb gelöst werden soll und die Benutzer sollen Kommentare zu einen Artikel schreiben und lesen können. Ebenso soll es möglich sein, dass der Kunde den Artikel bewerten kann. Der Artikel soll über eine Suche und eine Navigation zugreifbar sein. Zu einen Artikel soll in der Produktbeschreibung mehrere Produktbilder angezeigt werden, die sich der Kunde durch draufklicken größer ansehen kann. Die Produktbeschreibungsseite soll mindestens den Preis, die Bewertung, den Produktnamen und eine ausführlichen Beschreibungstext erhalten. Als negativ wurde empfunden, wenn die Startseite der Webseite „überladen“ von Produkten ist. Somit wurde beschlossen die Startseite des Webshops übersichtlich zu gestalten.

2.2 Anforderungsanalyse

Der zweite Schritt des Projekts war die Anforderungsanalyse. Hier wurde ermittelt, welche Anforderungen genau der Webshop erfüllen muss. Während der Analyse einigte man sich auf folgende Projektanforderungen:

- Man sollte sich registrieren können.
- Wenn man schon registriert ist, sollte man sich anmelden können.
- Die Daten zwischen dem ERP-System und dem Webshop sollen über eine Schnittstellendatenbank ausgetauscht werden.
- Um auf mobilen Geräten und Desktop-Computern die Webseite richtig anzeigen zu können soll eine zusätzliche CSS-Datei für mobile Geräte erstellt werden. Diese Datei soll die Style-Anweisungen überschreiben, die vom mobilen Gerät falsch angezeigt werden.
- Man sollte ein Artikel über die Navigation und die Suche erreichen.
- Die gefundenen Artikel sollten in einen Warenkorb gesammelt werden können (mit Mengenangabe).
- Nachdem der Warenkorb gefüllt ist, sollen die Artikel bestellt werden können. Hierbei sollen verschiedene Liefer- und Zahlungsmöglichkeiten auswählbar sein.
- Die Bestellung sollte mit einer Bestätigungs-Email bestätigt werden.
- Der Zustand der Bestellung sollte nachverfolgbar sein.
- Der Kunde sollte den Artikel bewerten können und Kommentare äußeren können.

2.3 Geschäftsprozess

Als nächstes wurde festgelegt, wie ein Geschäftsprozess beim Webshop aussehen soll. Der Geschäftsprozess beginnt, wenn der Kunde die Webseite aufruft. Darauf folgend registriert sich entweder der Kunde oder er logt sich mit der vorhandenen Benutzerkennung ein. Danach sucht sich der Kunde Produkte auf der Webseite aus und füllt damit den Warenkorb. Nachdem der Warenkorb gefüllt ist, führt der Kunde die Bestellung durch. Wenn die Firma GBI die Bestellung erhalten hat, wird an den Kunden eine Bestätigungsmaile versendet. Darauf folgend wartet der Webshop auf die Bezahlung der Produkte. Hat die Firma GBI die Zahlung erhalten, wird die Ware an den Kunden verschickt.

Dieser Geschäftsprozess ist nochmal in dem als „Business Process Model and Notation“ dargestellten Geschäftsmodell (*Abbildung 1*) zu sehen.

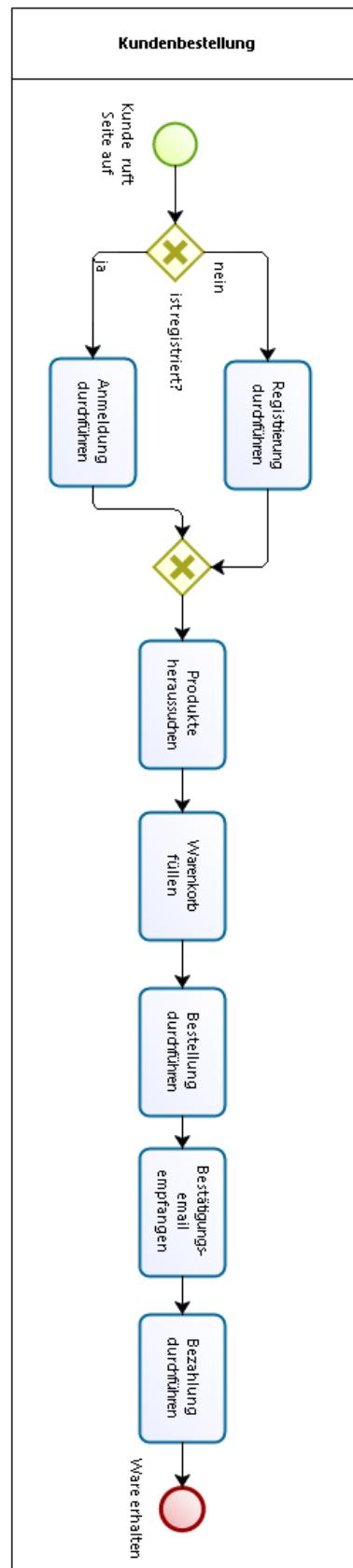


Abbildung 1: Geschäftsprozess des Webshops

2.4 Risikoanalyse

Ein wichtiger Bestandteil der Projektplanung ist die Risikoanalyse. Sie umfasst alle ermittelten Risiken, die zur Laufzeit des Projekts auftreten können und entsprechende Gegenmaßnahmen. Um die Risiken zu ermitteln wurde zuvor eine Risikoanalyse durchgeführt. Bei der Risikoanalyse werden die Risiken hinsichtlich ihrer Tragweite und Wahrscheinlichkeit bewertet. Die Bewertung reicht dabei von gering, über eher gering, eher hoch bis hoch. *Abbildung 2* zeigt die Einstufungen der Risiken zum Projekt sowie entsprechende Gegenmaßnahmen:

A	Risiken	Tragweite	Wahrscheinlichkeit	Lösungen
R2	Schnittstelle zur SAP Gruppe wird nicht fertig	hoch	eher hoch	Infomieren über direkten Zugriff
R3	Zeit zu knapp	hoch	eher hoch	Zeitplan mit Puffer
R5	Server nicht bereit	hoch	hoch	Lokalen Server
R10	Keine Berechtigungen auf fertigen Server	hoch	hoch	
B				
R4	Dinge sind nicht realisierbar	gering	eher gering	Verzichten?
R7	Zu viele Ziele	gering	eher hoch	Verzichten?
R8	Plattformunabhängigkeit nicht erreichbar	eher hoch	eher gering	
R9	Export auf fertigen Server schwierig	hoch	eher gering	Relative Pfade benutzen
C				
R1	Student fällt aus/ist nicht mehr da	hoch	gering	
R6	keine Räumlichkeiten	gering	gering	
R11	Daten gehen verloren	hoch	gering	ILIAS, Sicherheitskopien

Abbildung 2: Einstufung der Risiken und entsprechende Gegenmaßnahmen

Die Risiken werden in drei Kategorien unterteilt. Kategorie A beschreibt Risiken, die eine hohe Tragweite und Wahrscheinlichkeit aufweisen. Kategorie B beschreibt Risiken, deren Tragweite und Wahrscheinlichkeit weder zu hoch noch zu gering eingestuft wurden. Kategorie C beschreibt Risiken, deren Tragweite und Wahrscheinlichkeit eher gering eingestuft wurden. Der Kategorie A wurden die Risiken „SAP-Gruppe wird nicht mit Datenbank fertig“, „Zeit zu knapp“, „Server nicht bereit“ und „Keine Berechtigungen auf fertigen Server“ hinzugefügt. Das Risiko des Zeitmangels ist bei den meisten Projekten das größte Problem. Mangelnde Zeit bedeutet, dass das Projekt nicht im vollen Umfang fertiggestellt werden kann. Daher wurde es als eines der höchsten Risiken eingestuft. Risiken im Zusammenhang mit dem Server fallen ebenfalls unter dieser Kategorie. Ohne einen Server kann die Seite nicht betrieben werden, weshalb dieser kontinuierlich zur Verfügung stehen muss. Die SAP-Gruppe ist die wichtigste Schnittstelle, die es zu beachten gilt, da diese die Datenbank für den Webshop zur Verfügung stellt. Ohne Datenbank ist der Webshop nicht voll funktionsfähig.

Der Kategorie B wurden die Risiken „Dinge sind nicht realisierbar“, „Zu viele Ziele“, „Plattformunabhängigkeit nicht erreichbar“ und „Export auf fertigen Server schwierig“ hinzugefügt. Wenn sich bei den Zielen eines Projektes verschätzt wird und dadurch zu viele oder unrealistische Ziele gesetzt werden, kann nicht der vorher vereinbarte Umfang des Projektes gewährleistet werden. Plattformunabhängigkeit ist eines der Vereinbarten Anforderungen. Der Webshop sollte auf jeden Browser dargestellt werden können. Je mehr Browser die Webseite unterstützt, desto mehr Kunden können angesprochen werden. Der Export auf den fertigen Server könnte sich als schwierig erweisen, wenn Programme, die für den Export benötigt werden nicht auf den Server vorhanden sind.

Der Kategorie C wurden die Risiken „Student fällt aus/ist nicht mehr da“, „keine Räumlichkeiten vorhanden“ und „Daten gehen verloren“ hinzugefügt. Diese Risiken können auftreten, sind aber für das Projekt keine große Bedrohung. Das Risiko eines Datenverlustes wird durch mehrere Backups stark gemindert.

Das Risiko „Student fällt aus/ist nicht mehr da“ wurde anfangs als gering eingestuft. Zum jetzigen Zeitpunkt wird dieses Risiko allerdings als hoch eingestuft, da die Gruppenmitglieder Herr Gregarek und Herr Dück schon mehrfach ausgefallen sind.

Abbildung 3 zeigt die Ergebnisse der Risikoanalyse anhand einer Grafik:

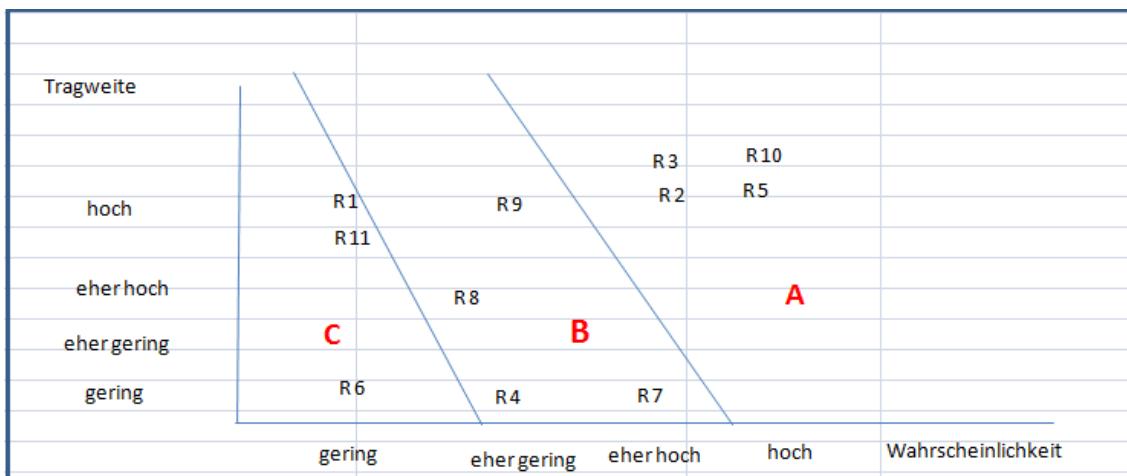


Abbildung 3: Ergebnisse der Risikoanalyse

Dabei lässt die Einteilung der Risiken in den einzelnen Kategorien besser nachvollziehen.

2.5 Wahl der Programmiersprache

Nachdem die Anforderungen an das Projekt festgelegt waren, wurde sich mit der Festlegung der Programmiersprache beschäftigt. Es stand Java Enterprise Edition und PHP zur Auswahl. Die Auswahl fiel auf PHP, da aus alten Projekten und der Webdesign-Vorlesung schon Vorwissen über die Programmierung von Webseiten in PHP vorhanden war. Zudem war auch geplant Teile des Quellcodes der alten Projekte für das neue Projekt wieder zu benutzen.

2.6 Wahl der Programmierentwicklungsumgebung und des Versionsverwaltungs-Plugin

Zum Beginn des Projekts hat jedes Gruppenmitglied einen eigenen Texteditor und an einer eigenen Version des Projekts gearbeitet. Es fiel nach kurzer Zeit auf, dass so das Zusammenfügen der verschiedenen Versionen sehr aufwendig war. Aus diesen Grund wurde nach einer Möglichkeit gesucht, das Problem zu lösen. Es wurde das „GitHub-Plugin“ für das Programm „Eclipse“ als Lösung für das Versionsproblem gefunden.

Als Entwicklungsumgebung wurde sich für Eclipse entschlossen, da es bei installierten PHP-Plugin eine gute Autovervollständigung für PHP bietet. Zudem ist es als Portable-Version

erhältlich und kann somit ohne Installation auf den Hochschulrechnern betrieben werden. Als Versionsverwaltungssystem wurde sich für „GitHub“ entschlossen, da dieses schon in den Programm „Eclipse“ integriert ist. Zudem ist „GitHub“ eine Cloud basierte Lösung, welche das gemeinsame Arbeiten an einem Projekt ermöglicht. Das komplette Projekt wird dabei auf einen Server hochgeladen. Jedes Gruppenmitglied, das einen Account bei „GitHub“ erstellt hat, kann dem Projekt beitreten, indem es bei den Gruppen-Administrator um Erlaubnis bittet. Dadurch kann ein Dokument von mehreren Gruppenmitgliedern gemeinsam bearbeitet werden. So entstehen mehrere Versionen von einem Dokument, die anschließend wieder mit Hilfe eines in GitHub enthaltenden „Merge-Tools“ zu einer gemeinsamen Version zusammengefügt werden können.

2.7 Grundlegender Aufbau und Ordnerstruktur der Webseite

Einer der ersten Schritte der praktischen Umsetzung des Projekts war die Festlegung der Ordnerstruktur der Webseite. Die Ordnerstruktur sollte einen einheitlichen Aufbau der Webseite garantieren. Mit der Festlegung sollte für jedes Projektmitglied definiert werden, wo welche Dateien abgelegt werden sollen und wie generell die programmiertechnische Struktur des Programms aussehen soll.

Als Programmieransatz für das Projekt wurde die objektorientierte Programmierung gewählt. Die Hauptgründe für die Auswahl des Programmieransatzes waren, dass die objektorientierte Programmierung durch die Vererbung die Möglichkeit bietet schon programmierte Funktionen an andere Klassen zu übernehmen. Es wird also Programmierarbeit eingespart. Ein Beispiel für die Einsparung von Programmierarbeit ist, wenn ein Fehler in einer vererbten Methode ist, muss der Fehler nur bei der Eltern-Klasse behoben werden und nicht bei der Kind-Klasse. Ein weiterer Grund ist, dass die objektorientierte Programmierung auch die Übersichtlichkeit der Webseite fördert. Für jedes Objekt der Webseite wird einfach eine Klasse angelegt, welche dieses definiert. So wird z.B. eine Klasse für den Kunden oder den Warenkorb erstellt. Diese Klasse arbeitet alle Aufgaben des Objektes ab. Der letzte Grund für den Ansatz war, dass schon objektorientierte Quellcodes aus alten Projekten zur Verfügung standen.

Da sich für die objektorientierte Programmierung entschieden wurde, hat sich die Gruppe auf folgende Ordnerstruktur geeinigt:

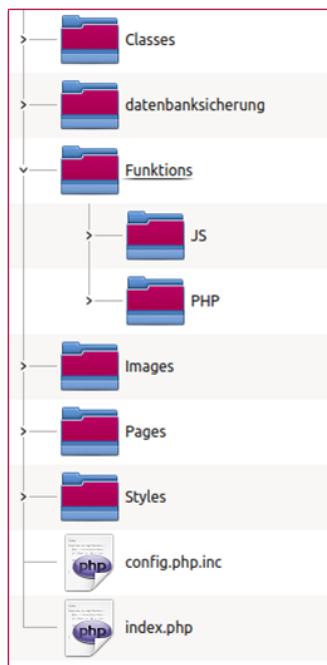


Abbildung 4: Ordnerstruktur des Projekts

Der Ordner „Classes“ beinhaltet alle PHP-Klasse des Projekts. Dieser Ordner enthält zum Beispiel Klassen zur Umsetzung des Warenkorbs, des Datenbankzugriffs, des Kundens, der Kundenbestellung und so weiter. Der Ordner „datenbanksicherung“ enthält die jeweils aktuelle Sicherung der Datenbank. Der folgende Ordner ist in zwei Unterordner aufgeteilt. Dieser Ordner wurde mit dem Namen „Funktions“ bezeichnet. Der erste Unterordner, der den Namen „JS“ trägt, enthält die clientseitig programmierten Funktionen. Die clientseitige Programmierung sollte in JavaScript umgesetzt werden. Aus diesen Grund sind in diesen Ordner JavaScript-Dateien vorzufinden. Die JavaScripts werden unter anderen dafür verwendet eine Formulareingabe vor den Absenden auf Gültigkeit zu überprüfen. Dieses sorgt dafür, dass es nicht zu unnötigen Datenverkehr zwischen Browser und Webserver kommt. Der Unterordner „PHP“ enthält die Funktionen, die zum Beispiel nach dem Absenden einer Formulareingabe an den Webserver aufgerufen werden. Diese Funktionen instanziieren die Klassen. Zudem rufen die Funktionen-Dateien die entsprechenden Methoden auf, um die von Formular empfangenen Eingaben zu verarbeiten. Dieser Ordner enthält noch zwei weitere Dateien. Die erste Datei der beiden heißt „set_page.php.inc“. Diese Datei liefert einen über die Adresszeile erhaltenden Parameter aus. An Hand des Parameters bestimmt die Datei, welcher Inhalt aktuell auf der Webseite angezeigt werden soll. Hierzu includiert die Datei die entsprechende HTML-Seite aus den Ordner „Pages“. Die zweite Datei heißt „set_control.php.inc“. Diese Datei includiert die benötigte Funktions-Datei, die z.B. die Formulareingaben bearbeiten. In diesen Projekt werden keine in PHP programmierten Funktionen und Inhaltseiten direkt aufgerufen. Diese Dateien werden immer in die Index.php includiert. Der Grund hierfür ist, dass so den Anwender der interne Aufbau der Webseite verborgen bleibt.

Der Ordner „Images“ enthält die Bilder, die auf der Webseite angezeigt werden. Der letzte Ordner mit den Namen „Styles“ enthält die CSS-Dateien, die das Design der Webseite festlegen. In der „config.php.inc“-Datei werden grundlegende Konfigurationen für den Webshop

festgelegt. In dieser Datei wird unter anderen die Zugangsdaten für die Datenbank angegeben. Die „index.php“ ist die standardmäßige Webseiteneinstiegsdatei. Über diese Datei wird die Webseite geladen.

2.8 Die Test-Server

Um bei der Entwicklung nicht erst darauf warten zu müssen bis die Server-Gruppe die benötigten Server-Dienst aufgesetzt hat, wurde sich nach einer Möglichkeit umgesehen, wie man für Testzwecke die Webseite auf den Hochschulrechnern laufen lassen könnte. Hierfür wurde das Programm XAMPP gefunden. Es ist eine Software-Sammlung, die alle benötigten Server-Dienste für PHP-Webseiten auf einen Windows-Rechner bereitstellt. Die Softwaresammlung bietet den Vorteil, dass eine portable Version erhältlich ist. Diese Version kann ohne Administrationsrechte ausgeführt werden.

2.9 Zeitplan

Nachfolgend wird der Zeitplan des Projektes dargestellt. Für jedes Arbeitspaket wurde eine Zeit von zwei Wochen eingeplant. Da jedes Gruppenmitglied 3 Arbeitspakte zugeteilt bekam, betrug die reine Arbeitszeit 6 Wochen. Herr Schnürer und Herr Brüntrup konnte diese Zeit weitestgehend einhalten. Zu der Arbeitszeit von Herrn Gregarek und Herrn Dück ist nichts Näheres bekannt. Da deren Arbeitspakte jedoch nicht vollständig bearbeitet wurden, ist deren Arbeitszeit deutlich geringer einzuschätzen. Die Arbeitszeit von 6 Wochen trug dazu bei, dass die Sommersemesterferien als Pufferzeit eingeplant werden konnten. *Abbildung 5* zeigt den Zeitplan, während *Abbildung 6* das dazugehörige Gantt-Diagramm zeigt. *Abbildung 7-13* beschreiben den aktuellen Projektstand anhand eines Ampelsystems.



Abbildung 5: Zeitplan

GANTT-DIAGRAMM

	Vorgar ▾	Vorgangsname	Dauer	Anfang	Ende	Vorgänger
1	■	Warenkorb	2 Wochen	Fre 17.04.15	Don 30.04.15	
2	■	Bestellvorgang	2 Wochen	Fre 01.05.15	Don 14.05.15	1
3	■	Marketing Mix	2 Wochen	Fre 15.05.15	Don 28.05.15	2
4	■	Nutzerdatenbank	2 Wochen	Fre 17.04.15	Don 30.04.15	
5	■	Registrierung und Anmeldung	2 Wochen	Fre 01.05.15	Don 14.05.15	4
6	■	Suche	2 Wochen	Fre 15.05.15	Don 28.05.15	5
7	■	CSS- Datei für mobile Geräte	2 Wochen	Fre 17.04.15	Don 30.04.15	
8	■	Bestätigung-E-Mail	2 Wochen	Fre 01.05.15	Don 14.05.15	7
9	■	Artikelbewertung	2 Wochen	Fre 15.05.15	Don 28.05.15	8
10	■	Inhalt der Seite	2 Wochen	Fre 17.04.15	Don 30.04.15	
11	■	CSS- Datei für Desktop Rechner	2 Wochen	Fre 01.05.15	Don 14.05.15	10
12	■	Navigation	2 Wochen	Fre 15.05.15	Don 28.05.15	11
13						
14	■	Alles Ferig	0 Tage	Fre 26.06.15	Fre 26.06.15	

Abbildung 6: Zeitplan

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Projektstand

Teil von Stefan

- CSS- Datei für Desktop Rechner
 - Geplante Bearbeitungszeit: 2 Wochen
 - Grundlegendes Design fertig
 - Muss eventuell noch angepasst werden nach Umsetzung anderer Pakete
- Inhalt der Seite
 - Geplante Bearbeitungszeit: 2 Wochen
 - Artikelseite in Arbeit (bisher auch erst mal noch statisch)
 - Startseite statisch fertig muss noch dynamisch umgesetzt werden

Abbildung 7: Zeitplan

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Projektstand

Teil von Stefan

■ Navigation



- Geplante Bearbeitungszeit: 2 Wochen
- Weitgehend fertig
- Werte werden schon aus der Datenbank geladen
- Muss noch mit der Artikelübersicht verknüpft werden

■ Rechtliches (war im Projektplan vorher nicht vorgesehen)

- AGB, Impressum, Datenschutz
- Seiten wurden angelegt und ins Projekt eingebunden

Abbildung 8: Zeitplan

Projektstand

Teil von Benedikt

- Nutzerdatenbank / Schnittstellendatenbank
 - Geplante Bearbeitungszeit: 2 Wochen
 - Fertig
 - Wird hin und wieder an die Anforderungen des SAP-Systems angepasst
 - Ständige Kommunikation mit der SAP-Gruppe

- Registrierung, Anmeldung und Benutzerprofil bearbeiten
 - Geplante Bearbeitungszeit: 2 Wochen
 - Registrierung und Anmeldung fertig umgesetzt
 - Die Möglichkeit das Benutzerprofil zu bearbeiten wurde erst mal zurückgestellt
 - Die Behebung der Probleme beim Bestellvorgang und Warenkorb wird als wichtiger angesehen



Abbildung 9: Zeitplan

Projektstand

Teil von Benedikt

■ Suche



- Geplante Bearbeitungszeit: 2 Wochen
- Normalsuche mit Autovervollständigung fertig
- Erweiterte Suche fertig (Formular und Suchalgorithmus)
- Zur Anzeige der Suchergebnisse wurde eine Artikelübersichtsseite entwickelt

Abbildung 10: Zeitplan

Projektstand

Teil von Raphael

■ Warenkorb



- Geplante Bearbeitungszeit: 2 Wochen
- Eine PHP-Klasse wurde angelegt mit Methoden für den Warenkorb
- Anscheinend gibt es Probleme bei der Umsetzung
- Wegen Kommunikationsmangel nichts weiter bekannt
- Keine Änderung am Quellcode bei GitHub seit 2 Wochen vorhanden

■ Bestellvorgang



- Geplante Bearbeitungszeit: 2 Wochen
- Noch nicht angefangen?
- Wegen Kommunikationsmangel nichts bekannt
- Wurde noch kein Quellcode bei GitHub hochgeladen

Abbildung 11: Zeitplan

Projektstand

Teil von Raphael

■ Marketing Mix

- Noch nicht angefangen?
- Wegen Kommunikationsmangel nichts bekannt



Abbildung 12: Zeitplan

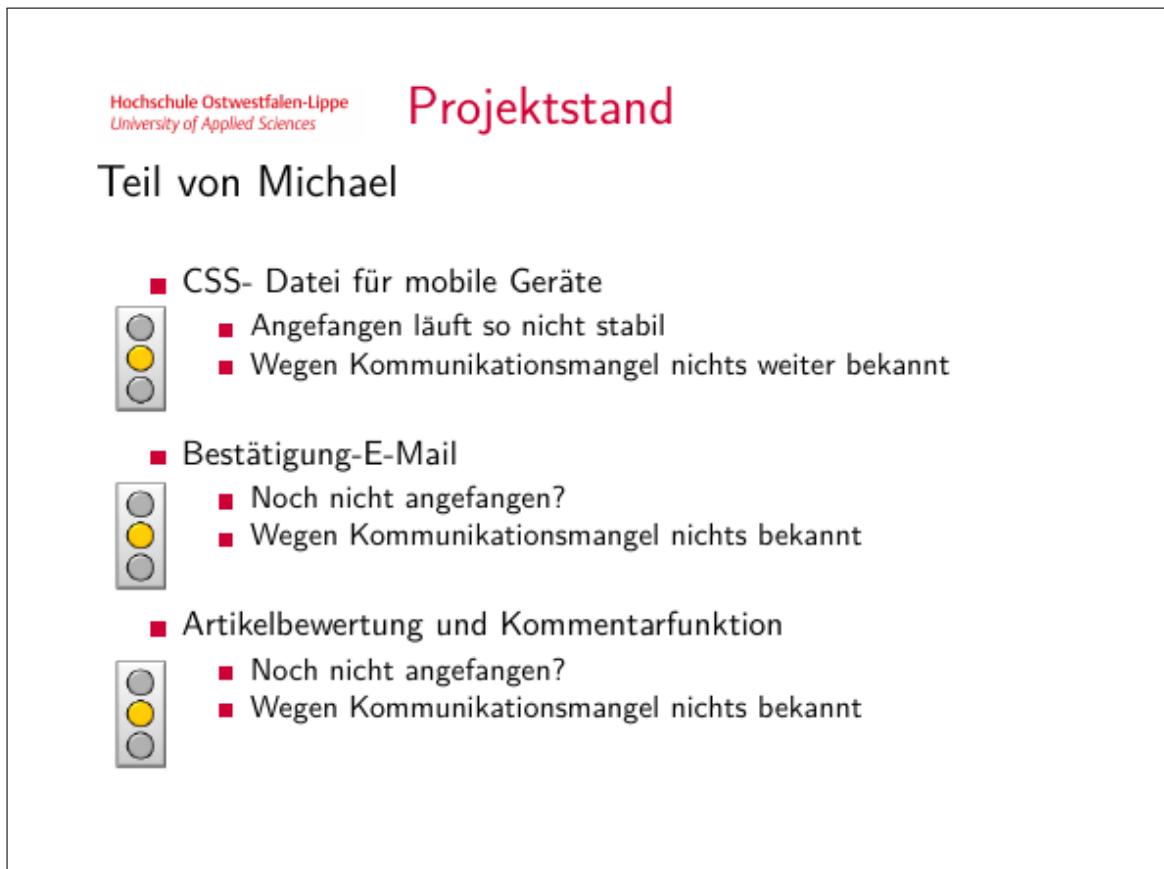


Abbildung 13: Zeitplan

3 Arbeitspakete

Dieser Abschnitt beschreibt die Unterteilung des Projektes in Arbeitspakete und wie die Gruppenmitglieder die zugeteilten Arbeitspakete abarbeitet haben. Es wird auch auf den unvorhergesehenen Arbeitsbereitschaftsmangel der Gruppenmitglieder Herr Gregarek und Herr Dück eingegangen.

3.1 Einteilung der Arbeitspakete

Zunächst wurden die Arbeitspakete aus der Anforderungsanalyse abgeleitet. Aus dieser Ableitung sind folgende Arbeitspakete definiert worden:

1. Entwicklung des grundlegenden Webseitendesigns für Desktop-Rechner
2. Anpassung des Designs für mobile Geräte
3. Entwicklung der Webseitenavigationsleiste
4. Entwicklung der inhaltlichen Seiten des Webshops:
 - Startseite

- Produktseite
 - Rechtliche Seiten:
 - ◊ Impressum
 - ◊ Datenschutz
 - ◊ AGB
5. Entwicklung einer Schnittstellendatenbank zum Datenaustausch zwischen Webshop und SAP-System:
- Ansprechpartner der SAP-Gruppe für die Schnittstelle zum Webshop
 - Entwurf von Tabellen für die Datenhaltung des Webshops
6. Registrierungs- und Anmeldungsfenster designen und programmieren
- Möglichkeit den Kunden bieten die Anmeldeinformationen später verändern zu können
7. Entwickeln einer Suchfunktion:
- Autovervollständigung
 - Erweiterte Suchfunktion, wo die Suche genauer eingegrenzt werden kann
 - Auflistung der Suchergebnisse
8. Warenkorb
- Auflistung der Produkte, die sich im Warenkorb befinden
 - Formular mit Mengenfeld, um ein Produkt in den Warenkorb zu packen
 - Möglichkeit zur nachträglichen Änderung der Produkte im Warenkorb:
 - ◊ Änderung der Bestellmenge
 - ◊ Entfernen des Produkts aus dem Warenkorb
9. Bestellvorgang:
- Bestellroutine mit folgenden Schritten:
 - ◊ Wahl der Versandart
 - ◊ Nachträglichen Änderung der Bestellmenge der Produkte
 - ◊ Wahl Zahlungsart
 - ◊ Wahl Lieferadresse
 - ◊ Bestellübersichtseite zur Kontrolle vor den Absenden der Bestellung
 - Einsicht des aktuellen Status der Bestellung
 - Möglichkeit die Bestellung zu stornieren
10. Erstellung eines Marketing Mix
11. Bewertungsfunktion von Artikel (mit Kommentarfunktion)

12. Email-Versand

- Klasse für den Email-Versand entwerfen
- Designen der Bestätigungs-Emails

Die oben genannten Arbeitspakete hat sich die Gruppe untereinander aufgeteilt. Stefan Schnürer hat Arbeitspakete 1,3 und 4 übernommen. Benedikt Brüntrup übernahm die Arbeitspakete 5, 6 und 7. Michael Dück hatte sich bereit erklärt die Arbeitspakete 2, 11 und 12 zu bearbeiten. Raphael wollte die Arbeitspakete 8, 9 und 10 abarbeiten.

3.2 Arbeitspaket von Herrn Schnürer

Dieser Abschnitt beschreibt den Aufgabenteil von Herrn Schnürer und wie dieser den Seitenprototyp, den statischen und dynamischen Inhalt und Aufbau der Seite sowie die CSS-Datei zur Anzeige des Webshops auf Desktop Rechnern entwickelt hat.

3.2.1 Arbeitspaket 1: Erster Seitenprototyp

Der erste Prototyp der Seite basiert auf den Standardverfahren einer Webseite. Dabei wird der Inhalt der Seite in HTML geschrieben, während eine CSS Datei für das Seitenlayout der Webseite verantwortlich ist. *Abbildung 14* zeigt einen ersten groben Designentwurf der Startseite, anhand dessen orientiert sich auch der erste Seitenprototyp von Herrn Schnürer:

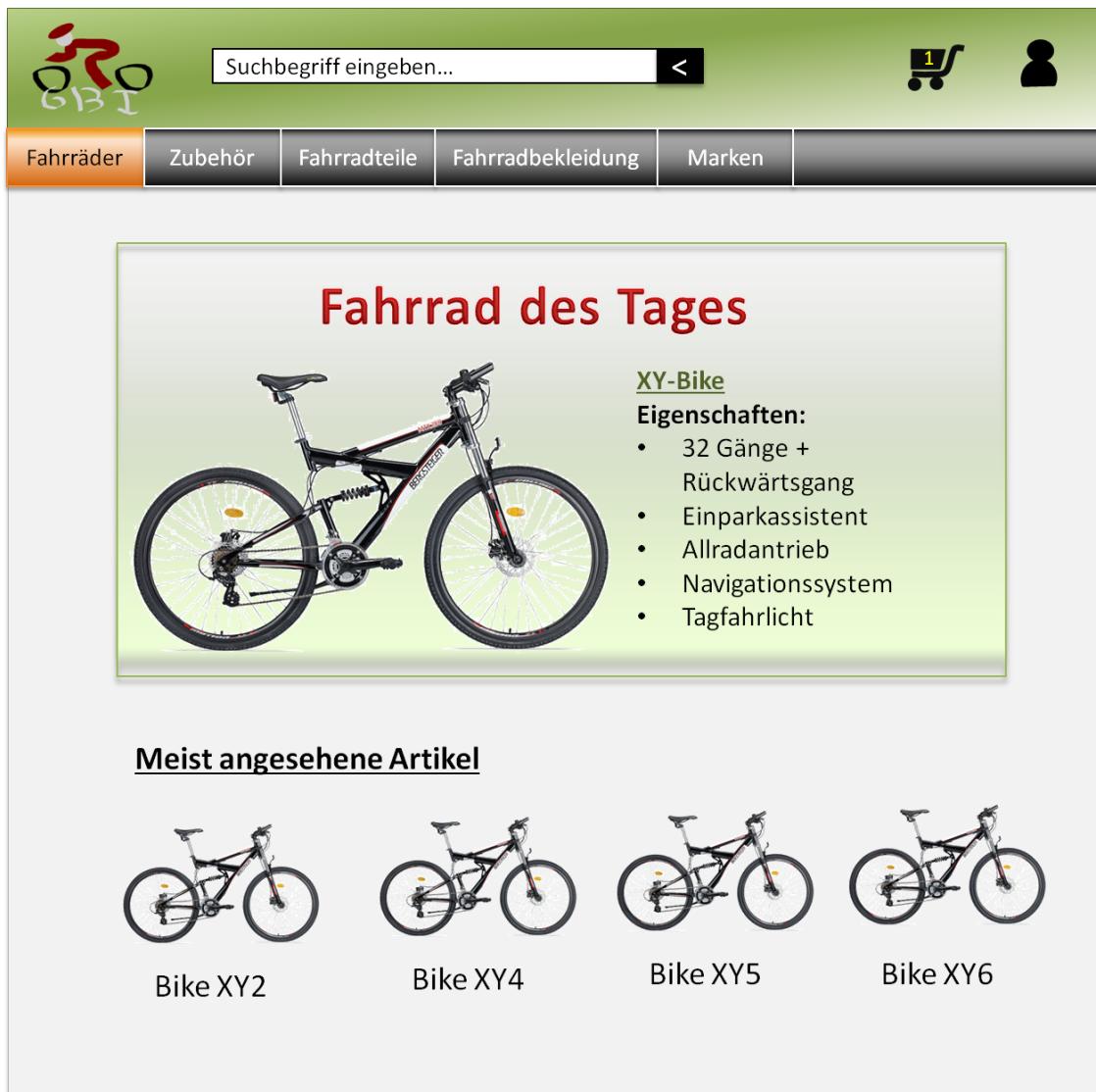


Abbildung 14: Erster grober Designentwurf der Startseite

3.2.2 Arbeitspaket 1 und 4: Startseite des Webshops

Die Startseite wird zunächst mit zwei einfachen Artikeln versehen. Sie sollen eine grobe Vorstellung liefern, wie die fertige Startseite aussehen könnte.

3.2.3 Arbeitspaket 3: Seitennavigation

In den darauf folgenden Wochen liegt der Fokus auf die Seitenavigation in der Kopfzeile (nachfolgend Header) der Seite. Die Seitenavigation wird mithilfe von CSS an den Designentwurf angepasst. Es werden die Hauptkategorien „Fahrräder“, „Zubehör“, „Fahrradteile“, „Fahrradbekleidung“, „Marken“ und eine Rubrik „HowTo“ angezeigt. Letztere verlinkt auf eine Internetquelle, mit dessen Hilfe die Seitenavigation umgesetzt wurde. Diese Kategorie dient den anderen Gruppenmitgliedern als Referenz und wird in späteren Versionen des

Webshops wieder entfernt. Fährt der Nutzer mit der Maus über die Kategorie „Fahrräder“ öffnet sich ein Drop-Down-Menü, welches den Nutzer verschiedene Fahrradmodelle auswählen lässt. Die komplette Seitennavigation wurde in CSS3 umgesetzt. CSS3 kann von nahezu jeden aktuellen Browser gelesen werden und bietet zudem den Vorteil, dass der entsprechende Code in der bisherigen CSS-Datei eingefügt werden kann. Da CSS3 unabhängig von Java Script ist, lässt sich die Seite zudem problemlos bedienen, wenn Java Script auf den jeweiligen Browsern deaktiviert ist. Dies begründet die Entscheidung CSS3 für die Seitenavigation zu verwenden.

Abbildung 15 und *16* zeigen den bisherigen Seitenprototypen. *Quellcode 1* veranschaulicht die Umsetzung des Dropdown-Menüs in CSS:

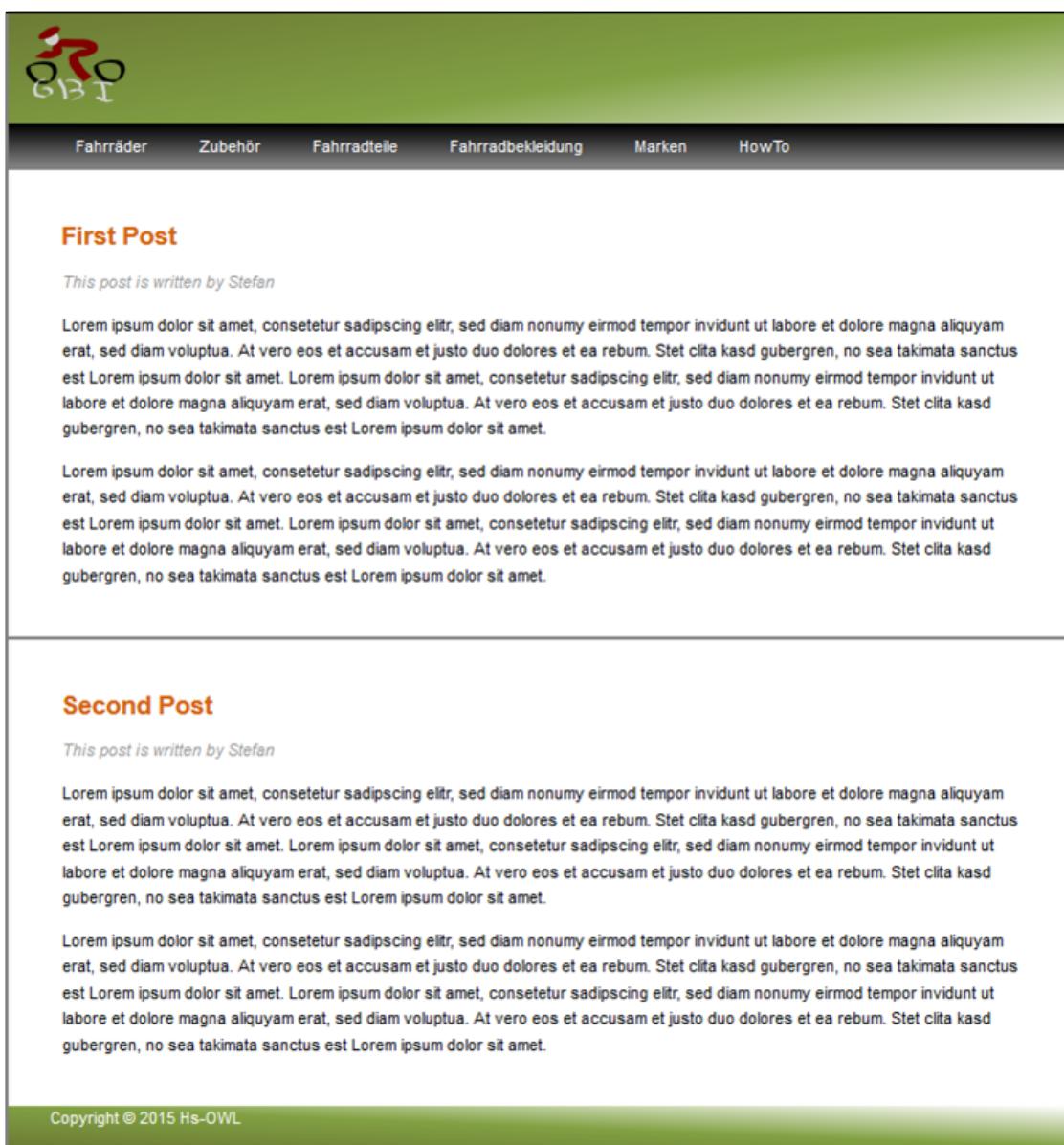


Abbildung 15: Seitenprototyp

The screenshot shows a website prototype for 'B3T'. At the top is a green header bar with the 'B3T' logo. Below it is a dark navigation bar with tabs: 'Fahrräder' (highlighted in orange), 'Zubehör', 'Fahrradteile', 'Fahrradbekleidung', 'Marken', and 'HowTo'. The main content area has a white background. It displays a grid of four rows and four columns of items labeled 'Modell 1' through 'Modell 8'. To the right of the grid is a large amount of placeholder text in Latin: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.' Below this is another section of placeholder text: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.' At the bottom left is a green footer bar with the text 'Copyright © 2015 Hs-OWL'.

Abbildung 16: Seitenprototyp mit Drop-Down-Menü

```
1  /*
2   * Der Quellcode zeigt nur einen kleinen Ausschnitt aus der kompletten CSS
3   * Datei.
4   * Der nachfolgende Quellcode wurde von Herrn Schnürer generiert.
5   */
6
7  /* Drop-Down Menü */
8  .navigation li:hover > ul {
9      /* Blendet das Dropdown- Menü ein, wenn die Maus über eine Kategorie
10     ist. */
11    visibility: visible;
12    padding: 0;
13 }
14
15 .navigation ul, .navigation ul li ul {
16     /* verbirgt das Dropdown- Menü wenn die Maus über keine Kategorie ist. */
17     visibility: hidden;
18     list-style: none;;
19     border-top : 12px solid #6c7b35 ;
20     /* Farbverlauf */
21     background: linear-gradient(to bottom, gray , black);
22     position: absolute;
23 }
24
25 .Untermenue{
26     width: 100px;
27     text-align: center;
28 }
29
30 .navigation .Spalte {
31     width: 150px;
32     height: auto;
33     float: left;
34     display: inline;
35 }
```

Quellcode 1: Auszug aus der CSS Datei - Die Seitenavigation

3.2.4 Arbeitspaket 4: Weitere Arbeiten an der Startseite des Webshops

Im Laufe der nächsten Woche wird die Startseite des Webshops gemäß dem Designentwurfs erstellt (siehe Abbildung 17). Auf der Startseite ist das „Angebot des Tages“ zu sehen. Es gibt eine Kurzbeschreibung zu dem Artikel mit den wichtigsten Eigenschaften. Der untere Teil der Startseite zeigt die meist angesehenen Artikel des Nutzers an.

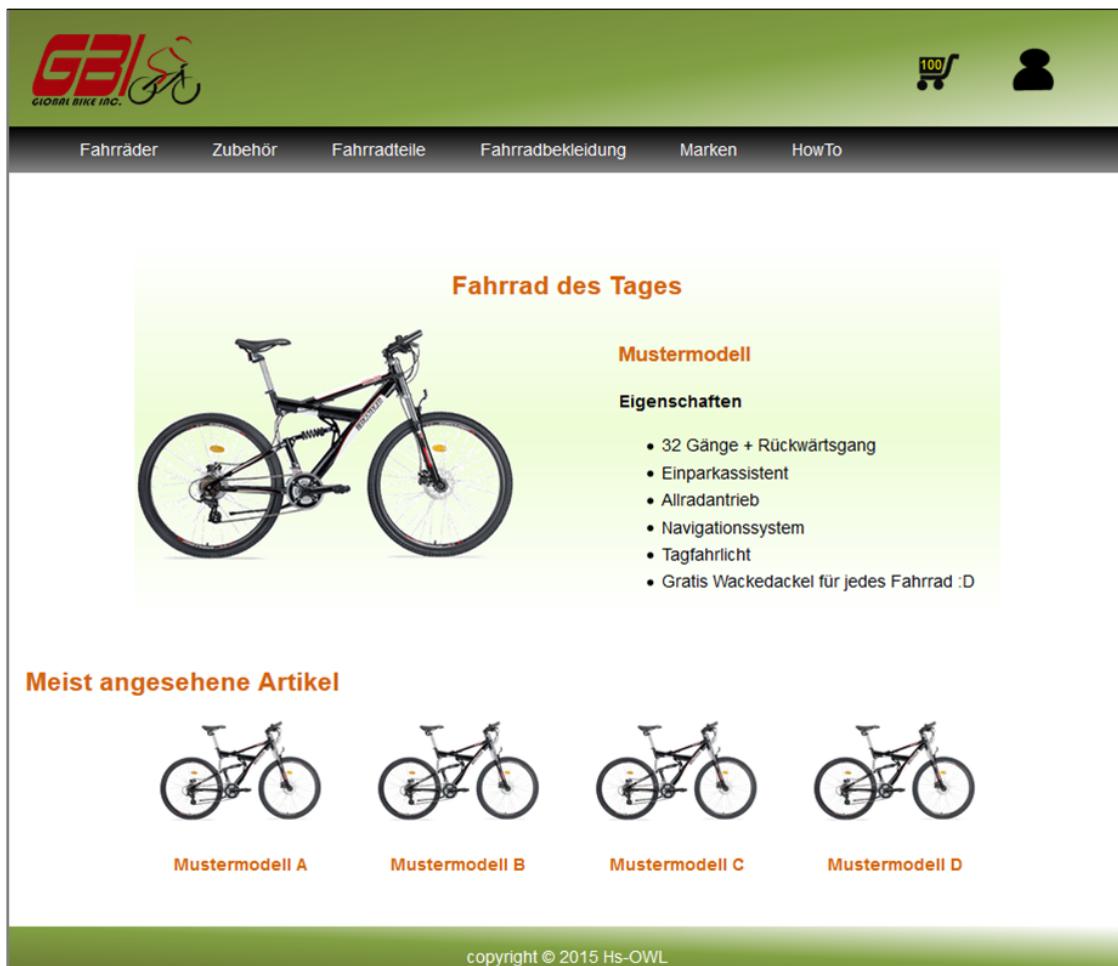


Abbildung 17: Startseite des Webshops

3.2.5 Arbeitspaket 4: Rechtsschutz

In den nächsten Tagen wird das Thema Rechtsschutz behandelt. Dem Webshop werden ein Impressum, die AGB sowie Hinweise zum Datenschutz hinzugefügt:

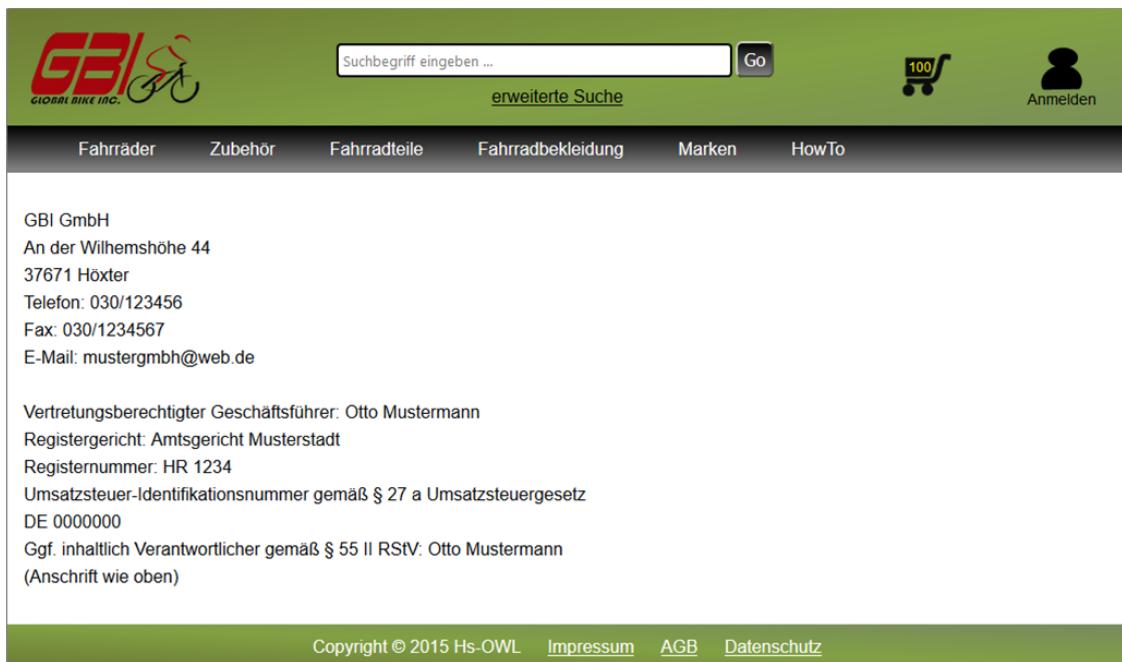


Abbildung 18: Impressum des Webshops



Abbildung 19: AGB des Webshops



Abbildung 20: Datenschutz des Webshops

3.2.6 Arbeitspaket 3: Dynamische Seitennavigation

In den nächsten Wochen wird der statische Inhalt des Webshops durch dynamischen Inhalt ersetzt. Das bedeutet, dass die Daten fortan aus einer Datenbank geladen werden. Diese Datenbank wird mit der Datenbank aus dem SAP- System synchronisiert. Zunächst wird die Navigation des Webshops durch dynamische Inhalte ersetzt. Wird nun eine neue Haupt- oder Unterkategorie hinzugefügt oder entfernt, ändert sich die Navigation entsprechend. Zum besseren Verständnis ein kurzes Beispiel: Angenommen die GBI würde jetzt auch Helmkeramas anbieten, so würde in der Datenbank eine neue Kategorie „Helmkeramas“ erstellt und zur ihr Kameramodelle von „GoPro“ und „Rollei“. Die Navigation hätte jetzt die Hauptkategorien „Fahrräder“, „Zubehör“, „Fahrradteile“, „Fahrradbekleidung“ und die Hauptkategorie „Helmkeramas“. Unter letzterer würden die beiden Unterkategorien „GoPro“ und „Rollei“ aufgeführt. Abbildung 21 zeigt die dynamische Navigation. Quellcode 2 zeigt den entsprechenden Quellcode der die Methoden der Klasse Navigation aufruft, um diese dynamisch zu erstellen:

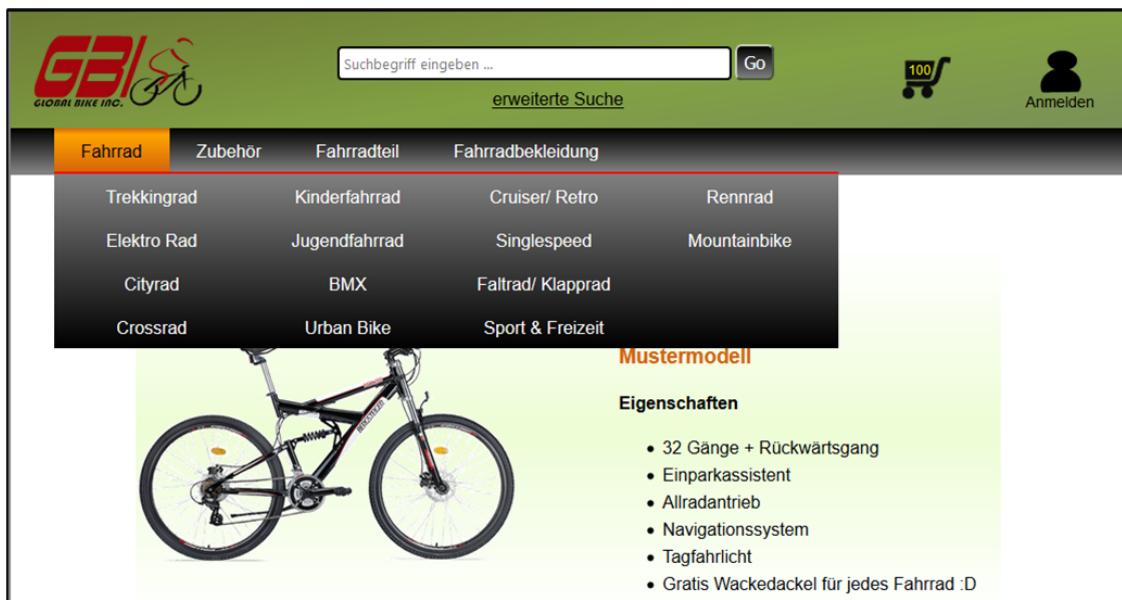


Abbildung 21: Dynamische Navigation

```
1 <?php
2 // Der Quellcode zeigt nur einen kleinen Ausschnitt aus der kompletten Klasse
3 // Navigation.
4 // Der nachfolgende Quellcode wurde von Herrn Schnuerer geeriert.
5 //
6 //
7 //
8
9 // Erzeugt das Drop-Down-Menü der Navigation
10    public function FahrradbauartAuflisten($bauart , $seitenzahl){
11        $query = $this->anfangszeilenSQLQuery(); //Vordefinierten Kopf
12            abspeichern
13            //Verbindung zur Datenbank aufbauen
14            $db = datenbank_verbindung_aufbauen();
15            if ( !($db->db_connect()) ) die ("Es konnte keine Verbindung zur
16                Datenbank aufgebaut werden");
17
18            $bauart=$db->db_real_escape_string($bauart); //Sonderzeichen aus
19                $bauart entfernen
20
21            //Anfrage ergänzen
22            $seite_sql = ($seitenzahl-1)*10;
23            $query.= " AND b.BId ='$bauart' ORDER BY p.Bezeichnung asc limit
24                $seite_sql, 10";
25
26            $result = $db->db_query($query); //Abfrage ausführen
27
28            $werte = array();
29            $i = 0;
30
31            //wird benötig um einzelne Elemente auf der Seite mit Namen
32            anzusprechen
33            while ($temp = $db->db_fetch_object($result)) {
34                $werte[$i] = $temp;
35                $i++;
36            }
37
38            $query= "select
39                    count(distinct p.Bezeichnung)
40                    from
41                        produkte p
42                        left join bauart b on (b.BId = p.bauart)
43                        left join produktkategorie pk on (pk.PTId
44                            = p.produktkategorie)
45                    where
46                        b.BId ='$bauart'";
```

```
41         $result= $db->db_query($query);
42
43         $gesammtseitenzahlen = 0;
44
45         while ($temp = $db->db_fetch_array($result)) {
46             $gesammtseitenzahlen = $temp[0];
47         }
48         $gesammtseitenzahlen = ceil($gesammtseitenzahlen/10); //Bestimmt
49             wie viele Seiten benötigt werden, wenn jede Seite nur 10
50             Einträge enthält
51
52         $this->erstelle_auflistung_aus_datenbank_reslut_array($werte, (
53             $seitenzahl), $gesammtseitenzahlen);
54
55     }
56 ?>
```

Quellcode 2: Auszug aus der Klasse Navigation - Die dynamische Seitennavigation

3.2.7 Arbeitspaket 4: Dynamische Artikelaufstellung

Ein paar Tage später erstellt Herr Schnürer eine dynamische Artikelaufstellung. In ihr werden alle Fahrräder einer bestimmten Kategorie aufgelistet. Wählt der Nutzer in der Seitennavigation die Hauptkategorie „Fahrrad“ und anschließend die Unterkategorie „Cityrad“ werden ihm alle „Cityräder“ aufgelistet, wählt er die Unterkategorie „Mountainbike“ werden ihm alle „Mountainbikes“ aufgelistet usw. *Abbildung 22* zeigt die dynamische Artikelaufstellung aller „Cityräder“ (Anmerkung: Es wird an dieser Stelle nur ein Cityrad aufgelistet, da sich nur ein Produkt mit der Kategorie „Cityrad“ in der Datenbank befindet. Man müsste nur weitere Cityräder in der Datenbank hinterlegen, damit entsprechend mehr Produkte aufgelistet werden):

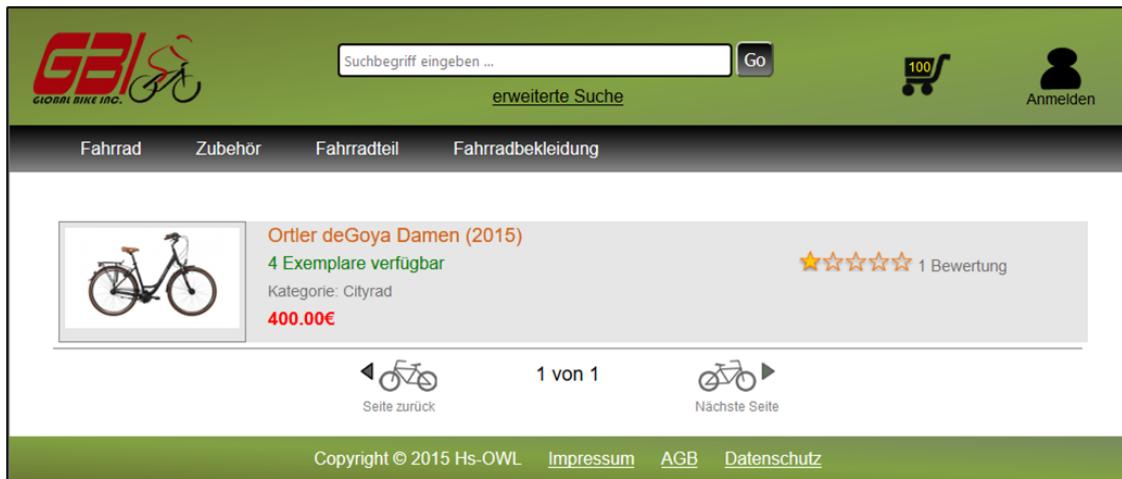


Abbildung 22: Dynamische Artikelaufstellung aller Cityräder

3.2.8 Arbeitspaket 4: Detaillierte dynamische Artikelbeschreibung

Eine Woche später wird eine detaillierte und dynamische Beschreibung für die Artikel erstellt. Wählt der Nutzer einen Artikel aus, erscheint eine detaillierte Beschreibung dessen. Der oberer Teil ist dabei an die Artikelbeschreibung der Startseite angelehnt und zunächst noch statisch, der untere Teil beschreibt den Artikel näher und ist bereits dynamisch umgesetzt. So erscheint unter der Überschrift „Artikelbeschreibung“ die jeweilige Beschreibung des zuvor ausgewählten Artikels. *Abbildung 23* zeigt eine solche Beschreibung für das Fahrrad „Ortler deGoya Damen (2015)“:

Serious Rockaway 26

Detailansichten:



Mountainbike

2 Exemplare verfügbar

Menge: **In den Warenkorb**

Eigenschaften

- 32 Gänge + Rückwärtsgang
- Einparkassistent
- Allradantrieb
- Navigationssystem
- Tagfahrlicht
- Gratis Wackedackel für jedes Fahrrad :D

Artikelbeschreibung

Mit dem De Goya lässt sich auf Ausflugstour bequem durch malerische Landschaften cruisen oder komfortabel kurze Wege in der Stadt bewältigen. Die Federgabel dämpft selbst gröbere Unebenheiten locker weg und schont Wirbelsäule sowie Gelenke. Auf dem Touren Komfortsattel sitzt man in moderat aufrechter Sitzposition auch über längere Strecken gut; zumal die Sitzposition über den verstellbaren Vorbau je nach Bedarf angepasst werden kann. 7 Gänge erlauben bequemes Fahren, auch wenn es etwas bergan gehen sollte. Bergab sorgen gut greifende V-Brakes in Kombination mit der beliebten Rücktrittbremse für Sicherheit. Der Fahrspaß und die Sicherheit sind starke Argumente für ein De Goya, nicht zuletzt aber besticht es mit einer sehr ästhetischen Optik der Anbauteile, einem Hauch von Nostalgie durch den Sattel und die Griffe sowie zurückhaltendem Understatement des Rahmens in schlichtem schwarzmatt. Im Detail ist das de Goya mit hochwertiger Schwalbe Pannenschutzbereifung, rostfreien Speichen, stabilen Hohlkammerfelgen und wartungsfreiem Innenlager auch da erstklassig ausgestattet, wo man es nicht auf dem ersten Blick sieht, sich aber langfristig besonders auszahlt.

Abbildung 23: Detaillierte dynamische Artikelbeschreibung

Wie bereits erwähnt ist der obere Teil zunächst nur statisch programmiert worden, weshalb die Überschrift noch nicht zum ausgewählten Artikel passt. Der Text „Artikelbeschreibung“ hingegen wird bereits aus der Datenbank geladen.

In den Sommersemesterferien hat Herr Schnürer auch den oberen Teil dieser Seite dynamisch erstellt. Die Produktbilder und Eigenschaften des Artikels werden nun ebenfalls aus der Datenbank geladen. Zudem wird noch ein kleines Extra eingebaut: Klickt der Nutzer eines der Produktbilder an, wird mithilfe eines Java Scripts eine „Lightbox“ aufgerufen. Diese zeigt die Produktbilder in ihrer normalen Auflösung und ermöglicht das einfache Durchklicken der Produktbilder. *Abbildung 24* zeigt den fertigen oberen Teil der Artikelbeschreibung, während *Abbildung 25* die „Lightbox“ veranschaulicht. *Quellcode 3* zeigt, wie dabei vorgegangen wurde:

Serious Rockaway 26"

Mountainbike

2 Exemplare verfügbar
279.99 €

Menge: In den Warenkorb

Eigenschaften

- 32 Gänge + Rückwärtsgang
- Einparkassistent
- Allradantrieb
- Navigationssystem
- Tagfahrlicht
- Gratis Wackedackel für jedes Fahrrad:D

Detailansichten:



Abbildung 24: Abbildung 12 Fertige dynamische Artikelbeschreibung



Abbildung 25: Abbildung 13 „Lightbox“ zur besseren Darstellung der Produktbilder

```
1 // Der Quellcode zeigt nur einen kleinen Ausschnitt aus der kompletten Index
2 // Datei.
3 //
4 //
5 <?php include_once 'Classes/ArtikelDetails.php.inc';?>
6 <div class = "startseite">
7 <article class="detailContent">
8
9     <?php
10
11         $artikelDetails = new ArtikelDetails();
12
13         $artikel = $artikelDetails->beschreibungAnzeigen($_GET['pid']);
14
15
16         echo '<h2>' . htmlspecialchars($artikel->Bezeichnung, ENT_QUOTES, 'UTF-8') . '</h2>';
17
18     ?>
19 //...
20 //...
21 //...
```

Quellcode 3: Auszug aus der Seite Artikel Detailansicht

3.2.9 Arbeitspaket 4: Startseite-Angebot des Tages

Zum Ende des Projektes wurde die Startseite des Webshops fertiggestellt. Sie zeigt das „Angebot des Tages“ in der oberen Hälfte. In der unteren Hälfte werden die meist angesehenen Artikel des Nutzers angezeigt. Abbildung 26 zeigt noch einmal die fertige Startseite:

Fahrrad des Tages

Detailansichten:



-
-
-
-
-



Mountainbike

2 Exemplare verfügbar

279.99 €

Eigenschaften

- 32 Gänge + Rückwärtsgang
- Einparkassistent
- Allradantrieb
- Navigationssystem
- Tagfahrlicht
- Gratis Wackedackel für jedes Fahrrad:D

Meist angesehene Artikel



Ortler deGoya Damen
(2015)

Abbildung 26: Abbildung 14 Startseite-„Angebot des Tages“

3.3 Arbeitspakete von Herrn Brüntrup

In diesem Abschnitt wird beschrieben, wie Herr Brüntrup seine Arbeitspakete gelöst hat. Zudem wird in diesem Abschnitt noch beschrieben welche Arbeitspakete wegen Zeitmangel und fehlender Arbeitsbereitschaft zusätzlich noch übernommen wurden.

3.3.1 Arbeitspaket 5: Die Schnittstellendatenbank

Die Datenbank stellt die Schnittstelle zwischen dem SAP-System und dem Webshop da. Aus diesem Grund wurde die Datenbankstruktur in der ersten Woche nach der Arbeitspaketverteilung zusammen mit Mitgliedern der SAP-Gruppe entworfen.

Gründe für die Schnittstellendatenbank

Die Gründe warum die Kommunikation mit einer Schnittstellendatenbank gelöst wurde, wird in den folgenden Sätzen genannt. Der erste Grund war die Verfügbarkeit. Der Webshop sollte

auch erreichbar sein, wenn mal keine Verbindung zum SAP-System besteht. Der SAP-Server steht extern bei einer Hochschule in Magdeburg. Es könnte also schon mal dazu kommen, dass das System nicht erreichbar ist. Ein anderer Grund war die Flexibilität. Die Datenhaltung sollte flexibel erweiterbar sein. Soll die Webseite z.B. durch eine 360°-Ansicht des Fahrrads erweitert werden, können einfach die benötigten Felder zur Datenbank hinzugefügt werden. Beim SAP-System ist das nur beschränkt möglich. Der Zugriff auf das SAP-System wird mit vordefinierten Funktionen gelöst. Diese Funktionen werden „Business Application Programming Interface (BAPI)“ genannt. Da die SAP-Gruppe beim SAP-System keine Berechtigungen hat neue BAPIs zu erstellen, wäre somit die Erweiterbarkeit des Webshops sehr eingeschränkt.

Aufbau der Schnittstellendatenbank

Nun wird der zu Beginn geplante Aufbau der Schnittstellendatenbank erläutert. Die zu Beginn geplante Datenbank bestand aus den Relationen „Kunde“, „Bestellungen“, „Produkte“ und „Produktbilder“. Die Relation „Kunde“ enthält die Login-Daten des Kunden sowie dessen Adresse. Die Relation „Bestellungen“ enthält alle Angaben zur Bestellung. Angaben sind z.B. die Zahlungs- und Versandart sowie der Status der Bestellung. Die vorletzte genannte Relation beinhaltet alle Angaben zu den Produkten, die beim Webshop verkauft werden sollen. Ihr wurde deswegen auch der Name „Produkte“ gegeben. Die Relation „Produktbilder“ enthält Pfadangaben, wo auf den Server die Produktbilder abgelegt sind. Da zu einem Produkt mehrere Produktbilder abgelegt werden können, wurde sich auf eine extra Tabelle für die Produktbilder geeinigt.

Eine Relation, die die Produkte einer Bestellung enthält, wurde zunächst vergessen umzusetzen. Das Vergessen der Relation fiel erst auf, als die SAP-Gruppe mit der Umsetzung des Bestellvorgangs bei der SAP-Schnittstelle beginnen wollte. Zur Lösung des Problems fügte die SAP-Gruppe nun nachträglich eine Relation „Bestellprodukte“ zur Datenbank hinzu. Diese Relation steht in Beziehung mit den Relationen „Bestellungen“ und „Produkte“.

Als Datenbank-Server wurde sich auf MYSQL geeinigt. Dieser Server lässt sich problemlos auf Linux installieren. Es war wichtig, dass die Datenbank unter Linux läuft, da die Server-Gruppe einen Linux-Server aufsetzen wollte. Zudem bietet MYSQL den Vorteil, dass auf eine MYSQL-Datenbank mit der Programmiersprache PHP gut zugegriffen werden kann. Die SAP-Gruppe wollte die Datenbank mit einer selbst geschriebenen JAVA-Anwendung füllen. Auch von JAVA aus kann gut auf eine MYSQL-Datenbank zugegriffen werden. Ein weiterer Grund für die Nutzung der MYSQL-Datenbank war, dass die Softwaresammlung XAMPP einen MYSQL-Server beinhaltet. Die Softwaresammlung XAMPP wurde von der Webshop-Gruppe als Testserver zum Testen der Webseite auf den Hochschulrechnern verwendet.

Die Abbildung 27 zeigt das „Entity-Relationship-Diagramm (ERD)“. Dieses Diagramm ist die erste Version der Datenbank, die zusammen mit der SAP-Gruppe entworfen wurde.

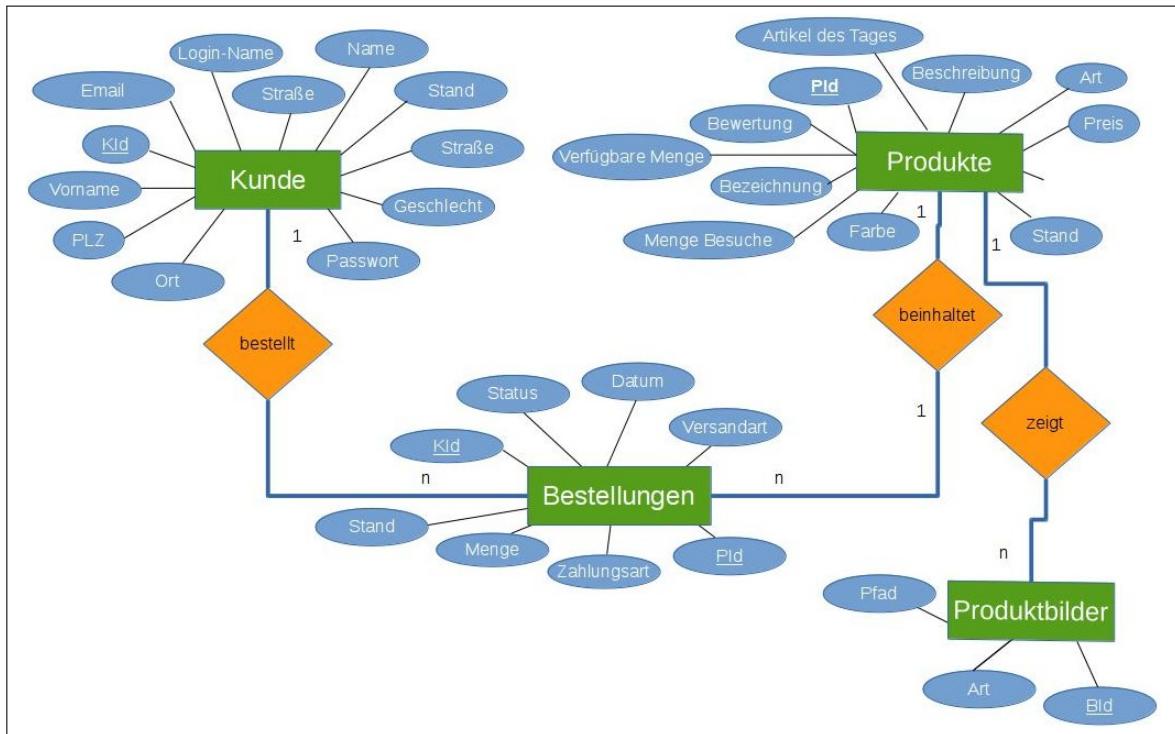


Abbildung 27: Erster Datenbankentwurf

Da es ursprünglich bemängelt wurde, dass das Projekt noch nicht den für das Modul benötigten Zeitaufwand aufweisen wurde, wurde beschlossen das Arbeitspaket „Bewertung“ mit einer Kommentarfunktion zu erweitern. Somit sollte der Kunde auch ein Kommentar für seine Bewertung äußern können. Dafür musste die Datenbank etwas angepasst werden. Vorher war einfach ein Attribut „Bewertung“ in der Relation „Produkte“ vorzufinden. Um die Kommentare für die Bewertung speichern zu können, wurde die Datenbank um eine Relation „Kommentare“ erweitert. Diese Relation beinhaltet ein Datenfeld, wo die Bewertung des Kunden (von 1-5 Sterne) gespeichert wird. Ein weiteres Datenfeld der Relation enthält den Bewertungstext.

Als mit der Suchfunktion und der Navigation begonnen wurde, fiel schnell auf, dass benötigte Attribute und Tabellen für die Navigation vergessen wurden. Somit wurde eine neue Relation „Produktkategorie“ zur Datenbank hinzugefügt. Diese Relation steht mit der Relation „Produkte“ in Beziehung und beinhaltet alle Produktkategorien, die der Webshop anbietet soll. Unter Produktkategorien wird beim Webshop die Festlegung verstanden, ob das Produkt ein Ersatzteil, ein Fahrrad, Zubehör usw. ist. Eine zweite Relation, die mit der Relation Produkte in Beziehung steht, ist die Relation „Bauart“. Dieses Relation wird nur verwendet, wenn das Produkt ein Fahrrad ist. In dieser Relation sind alle beim Webshop angebotenen Bauarten aufgelistet. Unter Bauart wird bei diesen Webshop verstanden, ob es sich um ein Mountainbike, Trekkingbike usw. handelt.

Die Abbildung 28 zeigt die endgültige Struktur der Schnittstellendatenbank mit allen nachträglich hinzugefügten Relationen.

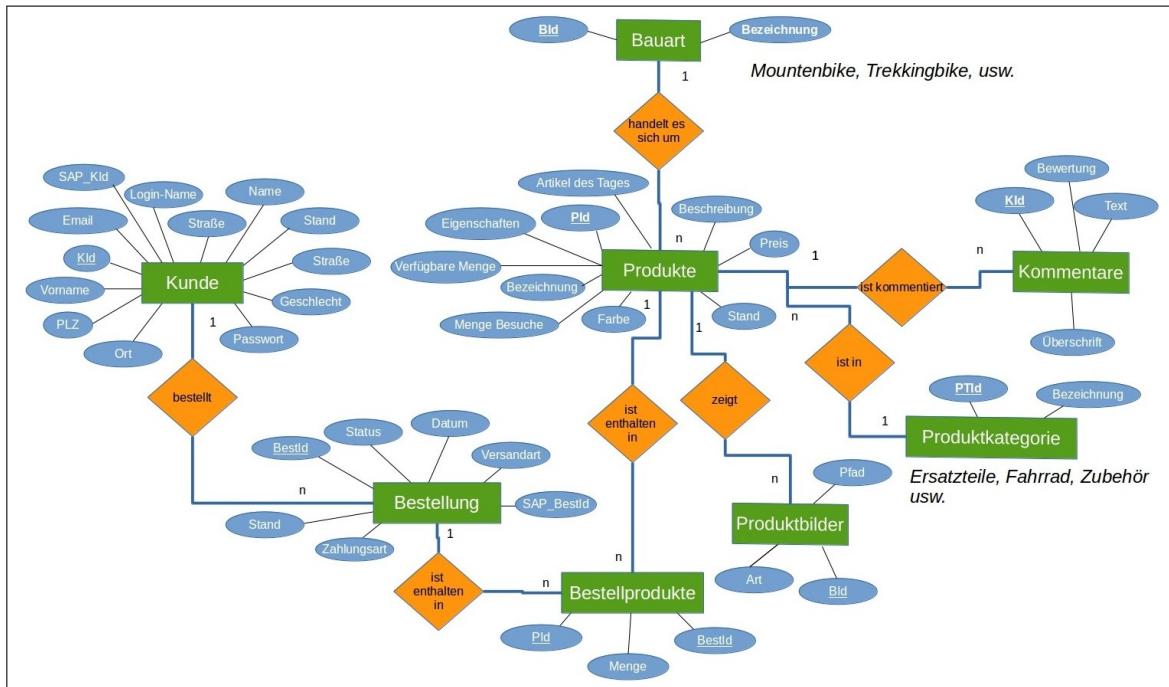


Abbildung 28: Endgültiger Datenbankentwurf

Wahl der PHP-API zum Datenbankzugriff

Um von PHP aus auf eine MYSQL-Datenbank zugreifen zu können, gibt es mehrere Möglichkeiten. In dieser Ausarbeitung werden die Zugriffsmöglichkeiten kurz verglichen und dann erläutert, welche davon für das Projekt benutzt wurde.

Die erste Möglichkeit ist die PHP-API „ext/mysql“. Diese API ist veraltet und wird nicht mehr weiterentwickelt. Sie unterstützt noch nicht alle MYSQL 5.1+ Funktionalitäten und ist stellenweise mit neuen PHP-Versionen nicht mehr nutzbar. Diese Erweiterung nutzt den prozeduralen Ansatz. Die nächste Möglichkeit ist die PHP-API „PDO“. Diese API unterstützt fast alle Funktionalitäten von MYSQL 5.1+. Sie bietet zudem den Vorteil, dass die API auch zum Zugriff auf andere SQL-basierten Datenbanken genutzt werden kann. Diese API hat einen objektorientierten Aufbau. Die letzte Zugriffsmöglichkeit bietet die API „ext/mysqli“. Sie unterstützt alle MYSQL 5.1+ Funktionalitäten und kann objektorientiert und prozedural programmiert werden.

Die Auswahl fiel auf die API „ext/mysqli“. Die Gründe hierfür waren, dass die API alle Funktionalitäten der neusten MYSQL-Version unterstützt, regelmäßig auf die neuesten MYSQL-Funktionen angepasst wird und schon Quellcodes von alten Projekten übernommen werden konnte. Die Abbildung 29 zeigt die Eigenschaften der drei APIs nochmal in einer Tabelle zusammen gefasst.

	ext/mysql	PDO_MySQL	ext/mysql
PHP version introduced	5.0	5.1	2.0
Included with PHP 5.x	Yes	Yes	Yes
Development status	Active	Active	Maintenance only
Lifecycle	Active	Active	Deprecated
Recommended for new projects	Yes	Yes	No
OOP Interface	Yes	Yes	No
Procedural Interface	Yes	No	Yes
API supports non-blocking, asynchronous queries with mysqlnd	Yes	No	No
Persistent Connections	Yes	Yes	Yes
API supports Charsets	Yes	Yes	Yes
API supports server-side Prepared Statements	Yes	Yes	No
API supports client-side Prepared Statements	No	Yes	No
API supports Stored Procedures	Yes	Yes	No
API supports Multiple Statements	Yes	Most	No
API supports Transactions	Yes	Yes	No
Transactions can be controlled with SQL	Yes	Yes	Yes
Supports all MySQL 5.1+ functionality	Yes	Most	No

Wahl

Abbildung 29: Überblick der PHP-APIs zum Datenbankzugriff

Die Konfigurationsdatei

Um auf eine MYSQL-Datenbank zugreifen zu können, werden üblicherweise mehrere Zugangsdaten benötigt. Die Zugangsdaten, die mindestens benötigt werden, sind der Datenbankbenutzername, das Passwort zum Benutzernamen, der Datenbankname und die IP-Adresse des Servers. Damit diese Zugangsdaten nicht bei jeden Datenbankzugriff im Quelltext angegeben werden müssen, wurden diese Angaben in eine Konfigurationsdatei ausgelagert. Wird die Webseiten später mal auf einen anderen Webserver betrieben, so müssen die Datenbankzugangsdaten nur in dieser Datei geändert werden. Die Konfigurationsdatei ist extra gesichert worden, damit sie nicht direkt im Browser geöffnet werden kann. Der Datei wurde als Dateiendung „*.php.inc“ angehängt. Mit einer bestimmten Zeile in einer Konfigurationsdatei für den Webserver wurde das Öffnen von „*.inc“-Dateien den Browser untersagt. Die Konfigurationsdatei trägt den Namen „.htaccess“ und befindet sich im Home-Vezeichnis der Webseite. Mehr zur „.htaccess“-Datei kann im Abschnitt „Sicherheit“ gelesen werden. Der *Quellcode 1* zeigt die Konfigurationsdatei des Webshops. In dieser Datei sind die benötigten Datenbankzugangsdaten und die festgelegte Email-Adresse zu lesen, von der alle Bestätigungs-E-mails versendet werden sollen. Zudem sind in der Konfigurationsdatei die Bankdaten definiert, die dem Kunden mitgeteilt werden, wenn er als Zahlungsart „Vorkasse“ ausgewählt hat.

```
1 <?php
2 //Angaben zur Verbindung mit der Datenbank
3 $benutzername = 'root';
4 $passwort = 'Passwort';
5 $datenbank = 'webshop';
6 $server = '127.0.0.1';
7
8 //Festlegung der Webshop-Emailadresse
9 $webshop_email_adresse = "webshop@fahrrad-gbi.de";
10 $webshop_email_name = "Webshop Fahrrad GBI";
11
12 //Konto-Daten der GBI für die Bestätigungs-Email
13 //((wenn Vorkasse von Kunden gewählt)
14 $bankdaten_gbi = array(
15     "inhaber" => "Global Bike Inc.",
16     "iban" => "DE08700901xxxxDemo",
17     "bic" => "1234567890B");
18 ?>
```

Quellcode 4: Die Konfigurationsdatei

Die Datenbankzugriffsklasse

Beim Webshop wird nicht direkt über die „MySQLi“-API auf die MySQL-Datenbank zugegriffen. Es wurde für den Datenbankzugriff extra eine Klasse entwickelt, die als Schnittstelle zwischen den Webshop und der „MySQLi“-API fungiert. Diese Klasse bestand schon von alten Projekten und wurde aus folgendem Grund entwickelt. Sie wurde entwickelt, damit nur an einer Stelle das Projekt verändert werden muss, wenn zu einem späteren Zeitpunkt mal ein anderes API verwendet werden soll. Dieses ist z.B. der Fall, wenn zu einer anderen Datenbank verwendet werden soll.

3.3.2 Arbeitspaket 6: Registrierungs- und Anmeldungenfenster designen und programmieren

Dieser Abschnitt beschreibt, wie beim Webshop die Registrierung und Anmeldung umgesetzt wurde. Zudem wird die Umsetzung einer Funktion des Webshops beschrieben, wo der Kunde seine Profildaten ändern kann.

Der Registrierungsvorgang

Auf dieser Webseite läuft die Registrierung wie in den folgenden Sätzen beschrieben ab. Zunächst tippt der Neukunde die von der Webseite verlangten persönlichen Informationen in ein Formular ein und bestätigt die Eingabe durch einen Klick auf den Bestätigungsbutton. Wenn beim Webbrower des Neukunden JavaScript nicht deaktiviert ist, überprüft der Webbrower mit einem JavaScript die Eingabe bevor sie zum Webserver weitergeleitet wird. Die Eingabe wird nur weitergeleitet, wenn diese vom JavaScript aus als gültig anerkannt wurde. Bei ungültiger Eingabe wird auf der Webseite eine Fehlermeldung ausgegeben. Der JavaScript ist keine endgültige Eingabeüberprüfung, dieser dient nur dafür unnötigen Datenverkehr zu minimieren. JavaScript besitzt den Nachteil, dass dieser beim Webbrower deaktiviert werden kann. Bei deaktivierten JavaScript wird somit die Eingabe unüberprüft an den Webserver weitergeleitet. Somit muss die Eingabe nochmal beim Webserver auf Gültigkeit überprüft werden. Der Neukunde wird also erst in die Kunden-Tabelle der Datenbank geschrieben, wenn die serverseitig programmierten Überprüffunktionen die Eingabe als gültig angesehen haben. Wurde die Eingabe als ungültig anerkannt, so wird auch bei der serverseitigen Programmierung eine Fehlermeldung auf der Webseite angezeigt. Nachdem der Kunde erfolgreich in der Kundendatenbank angelegt wurde, bekommt der Neukunde eine Bestätigungsmaile zu gesendet. Der Neukunde kann sich erst erfolgreich am System anmelden, wenn dieser den in der Email enthaltenden Link ausgeführt hat.

Das Passwort des Kunden wird nicht in Klartext in die Datenbank gespeichert. Bevor es in die Datenbank geschrieben wird, wird das Passwort mit einem „MD5-Hash-Algorithmus“ unkenntlich gemacht. Der MD5-Hashwert ist nicht mit einen Schlüssel wieder invertierbar. Er kann zur Speicherung von Passwörtern verwendet werden, da beim „MD5-Hash-Algorithmus“ der gleiche Input immer den gleichen Output erzeugt. Somit wird beim Login-Check einfach auch das eingegebene Passwort unkenntlich gemacht und verglichen, ob es mit den Wert in der Datenbank übereinstimmt.

Die Abbildung 30 zeigt den Registrierungsvorgang nochmal als BPMN dargestellt.

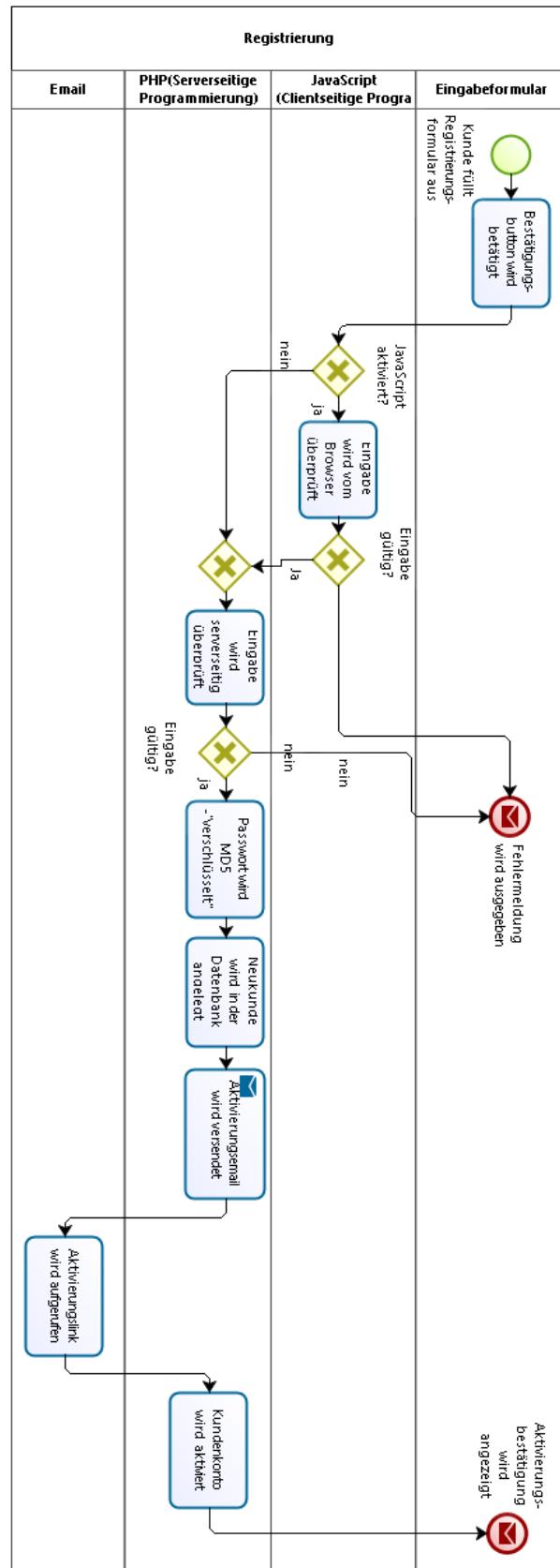


Abbildung 30: Der Registrierungsvorgang

Der Anmeldungsvorgang

Der Anmeldungsvorgang läuft ähnlich wie der Registrierungsvorgang ab. Der Kunde tippt als Benutzernamen seine Email-Adresse und als Passwort das bei der Registrierung angegebene Passwort in das Anmeldeformular ein. Wie bei der Registrierung überprüft ein JavaScript die Eingabe bevor diese an den Server weitergeleitet wird. Die vorherige Überprüfung ist wieder nur bei aktivierten JavaScript im Webbrower möglich. Deswegen wird die Überprüfung bei der Anmeldung auch nochmal vom Server überprüft. Hat der Kunde eine gültige Eingabe getätigt, so wird die Email-Adresse und das Passwort mit den Datensätzen in der Kunden-Tabelle der Datenbank verglichen. Um den Vergleich ausführen zu können wird bei der Anmeldung auch das Passwort durch den „MD5-Hash-Algorithmus“ unkenntlich gemacht. Wurde in der Tabelle ein Datensatz gefunden, wo Email-Adresse und Passwort übereinstimmen, so wird der Kunde eingeloggt. Wurde keine Übereinstimmung gefunden, so wird auf der Webseite eine Fehlermeldung ausgegeben.

Der Einlogvorgang wurde wie folgt umgesetzt. Die Anmelde-Email-Adresse wird in einer SESSION-Variable gespeichert. Der SESSION-Variable wurde der Name „\$_SESSION[‘angemeldet’]“ gegeben. An Hand dieser Variable erkennt der Webshop, ob der Kunde angemeldet ist oder nicht. Wurde die Variable angelegt und enthält einen Wert, so wird dieses als angemeldet interpretiert. Die Abmeldung erfolgt einfach durch das Löschen dieser Variable. Eine SESSION-Variable behält die enthaltenden Daten über mehrere Seitenaufrufe. Wenn nicht mehr auf die Variable zugegriffen wird, werden die Daten dieser Variable automatisch nach einer bestimmten Zeit gelöscht. Ob der Kunde angemeldet ist, muss in einer SESSION-Variable zwischengespeichert werden, da das HTTP-Protokoll verbindungslos ist. Es wird also jede HTTP-Anfrage isoliert betrachtet. Ohne die SESSION-Variable könnte sich somit der Webserver nicht merken, dass der Kunde angemeldet ist. Einer Session wird in PHP eine eindeutige „SESSION-ID“ zugeordnet. Diese SESSION-ID wird entweder als Cookie im Browser abgespeichert oder immer der Seiten-URL mit übergeben. Wird die SESSION-ID in einem Cookie im Browser abgespeichert, so wird dieser beim Schließen des Browsers gelöscht. Die Daten, die in der SESSION-Variable abgespeichert sind, werden auf der Serverseite zwischengespeichert und nicht im Browser. Die SESSION-ID wird von Browser bei jeder Anfrage an den Server weitergeleitet. An Hand dieser SESSION-ID kann der Server die SESSION-Variable zu den entsprechenden Browser zuordnen.

Die clientseitige Überprüfung

Wenn JavaScript aktiviert ist, werden, wie oben schon erwähnt, die Formulareingaben zuvor im Browser überprüft. Dafür wird beim Formular den Attribut „onsubmit“ eine JavaScript-Funktion übergeben. Die Formulareingabe wird hierbei nur an den Server weitergeleitet, wenn diese Methode als Rückgabewert „true“ zurückgibt. Die JavaScript-Funktion überprüft bei der Eingabe, ob überhaupt ein Text in die Textfelder eingegeben wurde und die Eingabe gültig ist. Zum Beispiel darf eine Postleitzahl nur aus 5 Ziffern bestehen. Die clientseitige Programmierung findet bei der Anmeldung in der Datei „/Funktions/JS/anmeldung.js“ und bei der Registrierung in der Datei „/Funktions/JS/registrierung.js“ statt.

```

1  <!-- JavaScript zur Überprüfung der Eingabe wird zur Webseite importiert -->
2  <script type="text/javascript"
3      src=".//Funktions/JS/vorcheck_std_funktionen.js" >
4  </script>
5  <script type="text/javascript" src=".//Funktions/JS/anmeldung.js" >
6  </script>
7
8  <form method="post" action="..." name="form_login" id="login"
9      onsubmit="return vorcheckEingabe_anmeldung('form_login');">
10
11     <!-- Textfeld zur Eingabe der Login-Email-Adresse -->
12     <label for="anmeldung_email">Email:</label>
13     <input type="text" name="anmeldung_email" maxlength="50"
14         class="lang_reg" />
15
16     <!-- Textfeld zur Eingabe des Passworts -->
17     <label for="anmeldung_passwort">Passwort:</label>
18     <input type="password" name="anmeldung_passwort" maxlength="50"
19         class="lang_reg" />
20
21
22     <!-- Fehlermeldungen bei ungültiger Eingabe -->
23     <p class="text_fehlermeldung" id="fehlermeldung_login">
24         <?php /* Fehlermeldungstext */ ?>
25     </p>
26
27     <!-- Der Submit-Button -->
28     <input type="submit" value="Login" title="Login"/>
29 </form>

```

Quellcode 5: Login-Formular (vereinfacht)

Der *Quellcode 2* zeigt in vereinfachter Version das Login-Formular. Beim „form“-Tag kann das „onsubmit“-Attribut gefunden werden. Dieses Attribut startet beim Klick auf den Submit-Button die enthaltende JavaScript-Funktion. Das Attribut lässt nur die Durchführung des Submits zu, wenn die JavaScript-Funktion als Rückgabewert den Wert „true“ zurücklieft. Um unnötigen Quelltext zu umgehen, wurde ein Teil des Eingabeüberprüfungsquellcode in eine andere JavaScript-Datei ausgelagert. Der ausgelagerte Quellcode wird bei jeden Formular benötigt. Dieser Quellcode überprüft, ob alle Pflichtfelder ausgefüllt sind. Die ausgelagerte Funktion trägt den Namen „sindAlleFelderAusgefuellt“ und befindet sich in der Datei „/Funktions/JS/vorcheck_std_funktionen.js“. Der Funktion werden beim Aufruf drei Parameter übergeben. Der erste Parameter erhält ein Array. Dieses Array enthält die Feldnamen der Textfelder, wovon die Eingabe überprüft werden soll. Der zweite Parameter enthält wiederum ein Array. Dieses Array enthält die Meldungswörter, die in der Fehlermeldung für das jeweilige Textfeld eingesetzt werden sollen. Die Meldungswörter werden in die Fehlermeldung eingesetzt, wenn das Textfeld nicht gefüllt ist. Beim letzten Parameter wird der Name des Formulars angeben, worauf sich die Textfelder befinden. Sind nicht alle Felder ausgefüllt,

erstellt die Funktion eine Fehlermeldung und gibt diese als Rückgabewert aus. In die Auslagerungsdatei wurden zudem Funktionen zur Überprüfung der Email-Adresse oder ob die Eingabe eine Zahl ist abgelegt. Die Funktionen in den Dateien „/Funktions/JS/anmeldung.js“ und „/Funktions/JS/registrierung.js“ rufen die ausgegliederten Funktionen auf und führen noch eine genauere Überprüfung durch. Die Funktionen testen zum Beispiel, ob das Passwort mit der Passwortwiederholung übereinstimmt.

Serverseitige Programmierung

Wie bei der clientseitigen Programmierung wurde auch bei der serverseitigen Programmierung die Methode zur Überprüfung ob alle Pflichtfelder ausgefüllt sind ausgelagert. Bei der serverseitigen Programmierung wurde hierfür die Technik der Vererbung genutzt. Serverseitig wurde der Webshop objektorientiert programmiert. Für die zuvor genannte Methode wurde die Klasse „EingabeCheckGrundlegend“ erstellt. Diese Klasse wurde an alle anderen Klassen vererbt, die Formulareingaben entgegen nehmen müssen. Für die Registrierung, Anmeldung und Änderung des Kundenprofils ist die Klasse „Kunde“ entwickelt worden. Sie enthält Methoden, mit der die Formulareingabe überprüft, ein neuer Kunde angelegt, ein Kunde verwaltet und die Anmeldung durchgeführt werden kann.

Instanziert wird die Klasse durch Skripte des Ordners „/Funktions/PHP“. Registriert sich ein neuer Kunde am Webshop, so wird von den Registierungsformular der Skript „registrierung_durchfuehren.php.inc“ des zuvor genannten Ordners aufgerufen. Dieser Skript instanziert die Klasse „Kunde“, überprüft die Formulareingabe und legt bei gültiger Eingabe den Kunden an. Zur Überprüfung der Eingabe und zum Anlegen des Kunden ruft der Skript die benötigten Methoden der Klasse „Kunde“ auf.

Weitere Skripte, die mit der Klasse „Kunde“ arbeiten, sind zum Beispiel die Skripte „anmeldung_durchfuehren.php.inc“, „kunde_aktivieren.php.inc“ und „kunde_aendern.php.inc“. Der erste der Skript der Liste ist für die Anmeldung des Kunden zuständig, der zweite aktiviert den Kunden, wenn der Aktivierungslink der Aktivierungsemail angeklickt wurde und der letzte Skript ist für das Ändern der Anschrift oder Benutzerkennung des Kunden zuständig. Die Klasse „Kunde“ legt alle Daten in der Datenbank ab. Wird z.B. die Instanzmethode „get_vorname_from_kid(\$kid)“ aufgerufen, so fragt die Methode über einen Datenbank-Select den Vornamen des Kunden von der Datenbank ab. Das Java-Schnittstellenprogramm der SAP-Gruppe lauscht die ganze Zeit, ob sich etwas an der Datenbank ändert. Kam es zu einer Änderung in der Datenbank, so wird die Änderung im SAP-System eingetragen.

Der Loginbereich im Header

Der Loginbereich im Header wurde mit einen sogenannten „Flyout“ umgesetzt (siehe Abbildung 31). Ein Flyout blendet ein Fenster auf der Webseite ein, wenn über ein bestimmtes Symbol mit der Maus gefahren wird. Der Loginbereich bei der Webseite wird also erst angezeigt, wenn sich die Maus auf den Anmeldesymbol befindet.

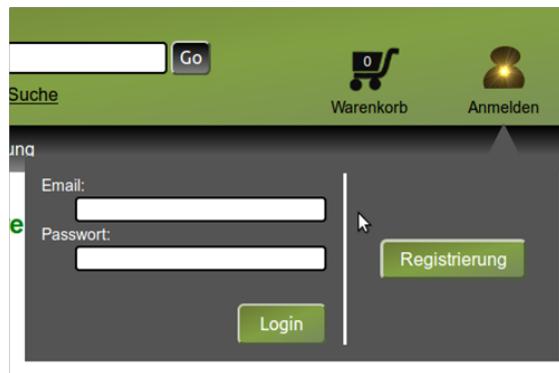


Abbildung 31: Flyout: Loginbereich

Die Besonderheit bei den Flyout des Webshops ist, dass es auch bei deaktivierten JavaScript funktioniert. Das Flyout wurde mit den in CSS 2.0 eingeführten „Kindselektor“ umgesetzt. Mit diesen „Kindselektor“ ist es eingeschränkt möglich mit CSS eventbasiert zu programmieren. Um diese CSS-Funktion nutzen zu können, muss aber folgende Bedingung erfüllt sein: Das Fenster, dass angezeigt werden soll muss ein Kindelement des Elements sein, welches das Ereignis auslöst. Die Syntax des „Kindselektors“ sieht wie im *Quellcode 3* zeigt aus. Am Anfang steht die Bedingung, die am Elternelement erfüllt sein muss. Beim Webshop wäre es die Bedingung, dass die Maus sich auf den Anmeldesymbol befinden soll. Diese Bedingung sieht in CSS geschrieben folgendermaßen aus: „.accountCell:hover“. Darauf folgt ein „>“-Zeichen. Dieses Zeichen legt fest, dass bei erfüllter Bedingung nicht die Style-Anweisung des Elternelements verändert werden soll, sondern die Style-Anweisung eines Kindelements. Nachfolgend wird das Kindelement angegeben, wo die Style-Anweisung geändert werden soll. Der letzte Teil der Style-Anweisung legt das neue Aussehen des Kindelements fest. Beim Webshop wird hier festgelegt, dass das Kindelement sichtbar werden soll. Ein Paar Zeilen vorher definiert eine Anweisung, dass das Kindelement unsichtbar sein soll. Bei gültiger Bedingung wird diese Anweisung einfach von der eben genannten Anweisung überschrieben.

```

1  /* Blendet den Loginbereich standardmäßig aus */
2  .popupRegistrierung, .popupWarenkorb{
3      ...
4      visibility:hidden;
5      ...
6 }
7 /*
8  * Zeigt den Loginbereich an,
9  * wenn die Maus sich auf dem Anmeldesymbol befindet
10 */
11 .accountCell:HOVER > .popupRegistrierung, .popupRegistrierung:hover{
12     visibility:visible;
13 }
```

Quellcode 6: Loginbereich: Umsetzung des Flyouts mit CSS 2.0

Kundenprofil ändern

Hat sich eine Kunde angemeldet, so bietet der Webshop ihm die Möglichkeit die bestehenden Login-Daten und die persönliche Anschrift zu ändern. Erreicht können die Formulare zur Änderung der Daten über ein Flyout. Wenn der Kunde angemeldet ist wird das Flyout unter den Anmeldesymbol zu einem Menü geändert (siehe *Abbildung 32*).

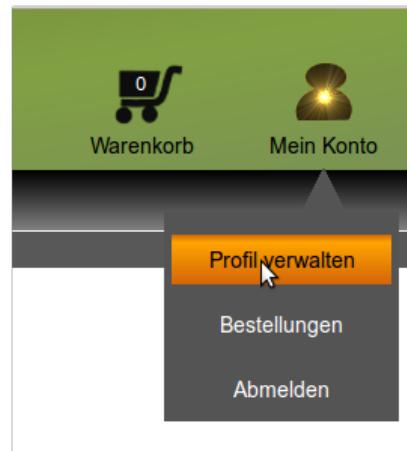


Abbildung 32: Flyout: Kunde angemeldet

Bei diesem Menü muss der Eintrag „Profil verwalten“ gewählt werden, um die bestehenden Login-Daten und die persönliche Anschrift ändern zu können. Nach der Anwahl des Eintrags bekommt der Kunde die in der *Abbildung 33* gezeigte Seite zu sehen.

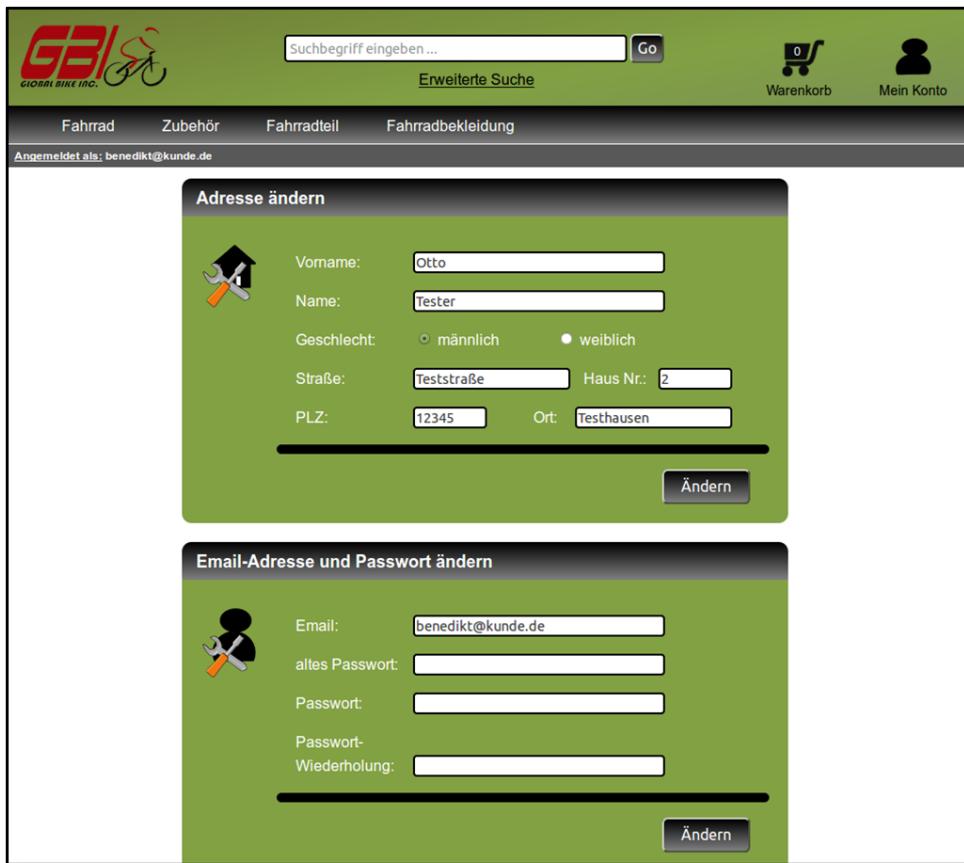


Abbildung 33: Formulare Kundenprofil ändern

Die Änderung des Kundenprofils wurde mit Hilfe von zwei Formularen umgesetzt. Das erste Formular ändert die Anschrift des Kunden und das andere die Logindaten. Der Vorteil dieser zwei Formulare ist, dass somit die Seite übersichtlicher gestaltet ist und meist der Kunde nur die Adresse ändert, wenn er umgezogen ist. Häufiger wird es wahrscheinlich vorkommen, dass der Kunde das Loginpasswort ändert. Durch die Trennung der Adresse und der Logindaten werden bei Änderung des Passworts nur das neue Passwort und die Login-Email-Adresse an den Server übertragen und neu in die Datenbank geschrieben. Somit werden so wenig wie möglich unnötige Daten zum Server übertragen.

Technisch umgesetzt wurde der Vorgang wie beim Registrierungsvorgang. Wenn JavaScript aktiviert ist, überprüft die JavaScript-Funktion „vorcheckEingabeEmailPasswort()“ oder „vorcheckEingabeAdresse()“ browserseitig die Eingabe. Die aufgerufene Funktion ist abhängig von welchen Formular der Submit-Button geklickt wurde. Diese Funktionen sind in der Datei „/Funktions/JS/“

vorcheck_profil_aendern.js“ vorzufinden. War die Überprüfung nicht erfolgreich, wird wieder eine Fehlermeldung auf den entsprechenden Formular ausgegeben. War die Überprüfung erfolgreich wird aus den bekannten Gründen diese nochmal auf dem Server durchgeführt. War die serverseitige Überprüfung erfolgreich, ruft die Funktionsdatei „/Funktions/PHP/-kunde_aendern.php.inc“ die entsprechende Änderungsmethode der Klasse „Kunde“ auf und übergibt der Methode als Parameter die Formulareingabe. Diese Methode ändert die ent-

sprechenden Werte in der Datenbank mit Hilfe einer Update-SQL-Anweisung. Wurde die Änderung erfolgreich ausgeführt, wird der Kunde über eine Informationsbox darüber informiert.

3.3.3 Arbeitspaket 7: Entwickeln einer Suchfunktion

Dieser Abschnitt beschreibt, wie die Suche beim Projekt umgesetzt wurde. Der Webshop hat eine „normale“ und eine „erweiterte“ Suche erhalten. Zudem wurde eine abstrakte Klasse entworfen, die Methoden für die Auflistung der Suchergebnisse bietet. Dieses Klasse wurde an die Klasse für die Suche vererbt und konnte ebenso von Herrn Schnürer für die Auflistung der über die Navigation gefundenen Produkte einer Kategorie verwendet werden.

Suchwortvervollständigung

Die Suchfunktion wurde beim Webshop mit einer Suchwortvervollständigung ausgerüstet. Diese Vervollständigungsfunktion ist mit Ajax programmiert worden. Mit Ajax kann der Browser über JavaScript, während eine Webseite geöffnet ist, Anfragen an den Server senden und die Antwort des Servers entgegen nehmen. Für den Webserver sind diese Anfragen wie ganz normale Seitenaufrufe. Will der Browser Parameter zur Anfrage hinzufügen werden dieses über „GET“ übermittelt. Dieses heißt, dass die Parameter an die Adresszeile angehängt werden.

Programmiert wurde die Suchwortvervollständigung, wie es in den folgenden Zeilen beschrieben wird. Beim Textfeld ruft das Attribut „onkeyup“ eine JavaScript-Funktion der Datei „/Funktions/JS/suche.js“ auf. Diese Funktion sendet mit AJAX eine Anfrage an den Server. Bei dieser Anfrage fragt der Browser den Server an, ob zu der Eingabe etwas in der Datenbank gefunden werden kann. Mit der Zeile „html2.onreadystatechange = autovervollstaendigung_normale_suche_result_erweiterte_suche;“ wird die Funktion festgelegt, die der JavaScript aufrufen soll, falls der Server antwortet. Diese Funktion nimmt den Rückgabewert des Servers entgegen und gibt diesen in einen DIV auf der Webseite aus. Der Rückgabewert ist mit Hyperlinks versehen. Dieses Hyperlinks sorgen dafür, dass bei Anwahl eines der Rückgabewerte das Suchtextfeld mit diesen Wert gefüllt wird und somit das Suchwort vervollständigt wurde. Die serverseitige Umsetzung der Suchwortvervollständigung findet in der Datei „/Funktions/PHP/ajax_autovervollstaendigung_suche.php“ statt. Diese Datei instanziert eine für die Suche entwickelte Klasse, wenn beim Aufruf der Datei ein Suchwort via GET-Parameter übergeben wurde. Die entwickelte Klasse trägt den Namen „Suche“. Nach der Instanzierung ruft die Datei eine Suchmethode der Klasse auf und gibt das Suchergebnis mit den Befehl „echo“ aus. Der Suchmethode wird als Parameter der Suchbegriff übergeben. Die aufgerufenen Methode ist abhängig von der Suchfunktion. Wurde die Anfrage von der „normalen“ Suchfunktion gesendet wird die Methode „normale_suche_autovervollstaendigung()“ aufgerufen. Ist der Absender der Anfrage die „erweiterte“ Suche, so wird die Methode „erweiterte_suche_autovervollstaendigung()“ aufgerufen. Die Ausgabe der Datei wird später von den JavaScript als den eben genannten Rückgabewert entgegengenommen.

Die eigentliche Suchfunktion

Beim Webshop wurden zwei Suchfunktionen implementiert. Eine „normale“ Suche im Header der Webseite und eine „erweiterte“ Suche. Bei der erweiterten Suche (siehe Abbildung 34)

kann der Kunde die Suche noch genauer spezifizieren. Der Kunde kann z.B. Angeben machen, um was für eine Bauart von Fahrrad es sich handeln soll (Mountainbike, Rennrad,...). Der Suchvorgang läuft bei beiden Suchfunktionen gleich ab. Wird der Submit-Button geklickt, wird zunächst erst über JavaScript überprüft, ob die Eingabe gültig ist. Wie schon beim letzten Arbeitspaket erläutert dient die browserseitig Überprüfung mit JavaScript nur zur Einsparung von unnötigen Datenverkehr zwischen Server und Browser. Aus den von letzten Arbeitspaket bekannten Grund, wird die Eingabeüberprüfung nochmal mit PHP durchgeführt. War die Eingabeüberprüfung mit PHP erfolgreich, wird die, für die Suchfunktion entsprechende Methode der Klasse „Suche“ aufgerufen. Die Suche wird innerhalb der Methoden rein in SQL umgesetzt. Bei SQL-Datenbanken ist die Suche praktisch schon fertig erhältlich. Der Programmierer muss nur einen entsprechende Abfrage formulieren und bekommt von der Datenbank die entsprechenden Rückgabewerte. Wird bei SQL der „like“-Operator verwendet, so unterstützt SQL auch Platzhalter für mehre Zeichen. Diese wurden bei den Suchfunktionen benutzt, um festzulegen, dass hinter und vor den Suchbegriff noch andere Zeichen stehen dürfen.

The screenshot shows a search interface titled "Erweiterte Suche". It features a magnifying glass icon on the left. The form contains the following fields:

- Produktbezeichnung:
- Preispanne: von: € bis: €
- Kategorie:
- Bauart:
- Bauvariante:
- Artikelnummer:
- Marke:

A "Suchen" button is located at the bottom right of the form.

Abbildung 34: Formular: Erweiterte Suche

Anzeige der Suchergebnisse

Bei der Anzeige der Suchergebnisse wurde versucht den Quelltext so zu programmieren, dass Herr Schnürer so viel, wie möglich für die Navigation übernehmen konnte. Es wurde somit versucht so viel, wie möglich in eine Klasse auszulagern. Dieses Klasse wurde so entwickelt, dass diese an die Navigations- und Suche-Klasse vererbt werden konnte. Die vererbten Methoden mussten also nur einmal programmiert werden. Der Klasse wurde der Namen „Artikelaufstellung“ gegeben. Sie ist in der Datei „/Classes/artikel.php.inc“ zu finden. Die Klasse enthält unter anderen eine Methode, die als Parameter den direkten Rückgabewert des Datenbankservers auf eine SQL-Anweisung entgegennimmt und aus den Rückgabewert eine

übersichtliche Liste der Produkte erstellt. Die Liste zeigt den Produktnamen, die vorhandenen Exemplare, die Produkt-Kategorie, den Preis und die Bewertung des Produktes an. Zudem ist ein Bild des Produktes zu sehen. Damit die Methode eine Leiste zur Seitenwahl am Ende der Webseite anzeigen kann, müssen der Methode noch weitere Parameter übergeben werden. Es wird in einen Parameter übergeben, welche Seitennummer aktuell aufgerufen ist. Ein anderer Parameter nimmt entgegen, wie viele Seiten insgesamt zum Suchergebnis gefunden wurden.

Die Methode kann nur den Datenbankrückgabewert verarbeiten, wenn dieser die festgelegten Feldnamen enthält. Da bei der Navigation und der Suche die SQL-Anweisung fast gleich ist, wurde der gleiche Teile in eine Methode ausgelagert. Diese Methode gibt somit den vorderen Teil der SQL-Anweisung als Rückgabe aus. Für die Suche und die Navigation muss in der entsprechenden Klasse also nur noch der Rückgabewert dieser Methode durch den „WHERE-Teil“ der Anfrage ergänzt werden.

Die Abbildung 35 zeigt wie ein Suchergebnis des Webshop aussieht. Beim gezeigten Suchergebnis wurde nach Fahrrädern im Preisbereich zwischen 250 und 400 € gesucht.



Abbildung 35: Suchergebnis

3.3.4 Zum Teil übernommenes Arbeitspaket: Warenkorb

Auf Grund von mehreren Fehlstunden des Gruppenmitglieds Raphael Gregarek war es beim Projektverlauf zu größeren zeitlichen Rückständen beim Arbeitspaket „Warenkorb“ gekommen. Da die Arbeitspakete von Benedikt Brüntrup schon weitgehend fertig waren, wurde somit die restlichen Aufgaben des Arbeitspaketes „Warenkorb“ von Herrn Brüntrup übernommen. Die Einarbeitungszeit in den bestehenden Quellcode war recht kurz, da dieser schon von Hilfestellungsarbeiten bekannt war. In den folgenden Zeilen wird beschrieben welche Funktionen des Arbeitspaketes Herr Brüntrup umgesetzt hat.

Eingabeüberprüfung

Wie bei den anderen Formularen wurde auch beim Warenkorb eine Eingabeüberprüfung programmiert. Diese wurde clientseitig in der Datei „/Funktions/JS/vorcheck_warenkorb.js“ und serverseitig in der Datei „/Funktions/PHP/warenkorb.php.inc“ umgesetzt. Die Überprüfung wurde, wie bei den anderen Formularen umgesetzt und wird deswegen nicht weiter in

dieser Ausarbeitung erläutert. Die Eingabeüberprüfung überprüft, ob überhaupt eine Menge angegeben wurde und ob diese im gültigen Bereich liegt. Es können somit keine negative Mengen und auch nicht mehr Produkte, als auf Lager sind, bestellt werden. Zudem wird als Wert nur eine ganze Zahl unterstützt. Die *Abbildung 36* zeigt die Fehlermeldung der Eingabeüberprüfung, wenn der Kunde mehr Artikel bestellen will, als auf Lager sind.



Abbildung 36: Eingabeüberprüfung Warenkorb

Warenkorbübersichtsseite

Bei der Warenkorbübersichtsseite handelt es sich um eine Seite, wo der Kunde den Warenkorbinhalt genauesten aufgelistet bekommt. Auf dieser Seite kann der Kunde die Bestellmenge nachträglich verändern oder das Produkt ganz aus dem Warenkorb entfernen. Zur Umsetzung dieser Seite wurden die von Herrn Gregarek schon entwickelten Klassen um Methoden erweitert. Herr Gregarek hatte bis zum Zeitpunkt der Erstellung der Warenkorbübersichtsseite schon ein Flyouts im Header der Webseite entwickelt, welches den Kunden übersichtlich die Artikel anzeigt, die sich im Warenkorb befinden. Das Flyout dient nur als Schnellübersicht und bietet keine Funktionen, um den Inhalt des Warenkorbs bearbeiten zu können. Die *Abbildung 37* die Warenkorbübersichtsseite.

Warenkorb					
Im Warenkorb befinden sich 1 Artikel:					
Bild	Name	Menge	Einzelpreis	Gesamtpreis	
<input type="checkbox"/>	Serious Rockaway 26"	1	279,99 €	279,99 €	
Alle Artikel markieren				Summe:	279,99 €
				Markierte Artikel löschen	Zur Kasse

Abbildung 37: Warenkorbübersichtsseite

Änderung der Bestellmenge

Die Änderung der Bestellmenge erfolgt durch einen Klick auf dem Hyperlink mit dem Stift-Symbole. Um die Webseite nicht ganz so altmodisch wirken zu lassen, aber trotzdem bei abgeschalteten JavaScript lauffähig zu haben, wurde sich entschlossen die Mengenänderung mit einem Dialog-Fenster zu lösen. Der Hyperlink lädt die aktuelle Webseite neu

und gibt als „GET-Parameter“ an, dass die Menge des Artikels mit der Artikelnummer x geändert werden soll. Die Datei „/Funktions/warenkorb.php.inc“ nimmt die Parameter entgegen und ruft ein für vorherige Projekte entwickeltes Dialog-Fenster auf. Das Dialog-Fenster schwebt durch die Style-Anweisung „position: absolute;“ über der Webseite. Beim Klicken des „OK“-Buttons wird die Seite wieder neu geladen und über „POST“ die neue Menge des Artikels an den Server übertragen. Wie dieses Dialog-Fenster aussieht kann der Abbildung 38 entnommen werden. Bei gültiger Eingabe ruft der Server die Methode „aende-re_bestellmenge_artikel_im_warenkorb(\$pid, \$menge)“ auf und ändert die Mengenangabe im Warenkorb.

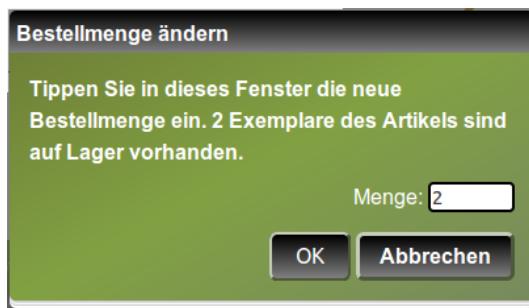


Abbildung 38: Dialog: Bestellmenge ändern

Löschen eines Artikels aus dem Warenkorb

Um ein Artikel aus dem Warenkorb zu löschen wurden zwei Methoden umgesetzt. Bei der ersten Methode handelt es sich um einen Hyperlink, der sich hinter jeder Artikelzeile auf der Warenkorbübersichtsseite befindet. Der Hyperlink wird auf der Seite durch ein rotes Kreuz dargestellt. Dieser Hyperlink lädt die Seite neu und übergibt als Parameter, dass der Artikel mit der Artikelnummer X gelöscht werden soll. Die Datei „/Funktions/warenkorb.php.inc“ nimmt wieder den Parameter entgegen und ruft ein Dialog-Fenster auf (siehe Abbildung 39). Beim Dialog-Fenster muss der Kunde bestätigen, dass wirklich der Kunde gelöscht werden soll. Der „OK-Button“ des Dialogs ist mit einem Hyperlink gelöst worden. Dieser Hyperlink lädt die Seite neu und übergibt als Parameter, dass der Artikel wirklich gelöscht werden soll. Die Datei „/Funktions/warenkorb.php.inc“ erkennt den gesetzten Parameter und ruft die Methode „entferne_artikel_aus_warenkorb(\$pid)“ der Klasse „Warenkorb“ auf. Diese Methode löscht den Artikel aus dem Warenkorb. Die Variabel „\$pid“ enthält bei dieser Methode die Artikelnummer des Artikels, der aus dem Warenkorb gelöscht werden soll.

Eine weitere Löschfunktion wurde mit Checkenboxen umgesetzt. Durch die Checkboxen kann der Kunde mehrere Artikel löschen. Der Kunde markiert mit den Checkboxen die Artikel, die gelöscht werden sollen. Durch einen Klick auf den Button „Markierte Artikel löschen“ wird die Markierung an den Server weitergeleitet. Die Weiterleitung findet über die Methode „POST“ statt. Bei den Button „Markierte Artikel löschen“ handelt es sich um einen Submit-Button. Die Checkboxen befinden sich in einem HTML-Formular. Dieses Formular überträgt bei einer Bestätigung des Submit-Buttons die Namen der markierten Checkboxen und die zugeordneten Artikelnummern an das PHP-Skript „/Funktions/warenkorb.php.inc“. Das Skript zeigt nach dem Erhalt der Daten auch ein Dialog mit einer Sicherheitsfrage an. Wird diese Frage bestätigt sorgt der Bestätigungs-Hyperlink dafür, dass die Methode „mehre-

re_artikel_aus_warenkorb_entfernen(\$arr_pids)" der Klasse „Warenkorb“ aufgerufen wird. Diese Methode löscht alle als Array übergebenen Artikel aus den Warenkorb. Der Parameter „\$arr_pids“ enthält somit ein Array mit Artikelnummern der Artikel, die aus dem Warenkorb entfernt werden sollen. Das Array wird aus dem vom Formular übertragenen Daten erstellt.



Abbildung 39: Dialog: Sicherheitsfrage Artikel löschen

3.3.5 Zum Teil übernommenes Arbeitspaket: Bestellvorgang

Da auch der Bestellvorgang etwas hinter den Zeitplan lag, wurde hier auch ein Teil von Herrn Brüntrup übernommen. Es wurde eine Routine programmiert, die nach dem Klicken des Buttons „Zur Kasse“ abgelaufen wird. In dieser Routine stellt der Kunde alle Einstellungen, die mit der Bestellung zu tun haben, ein. Hierzu gehört die Versandart, eine nochmalige Möglichkeit den Warenkorb zu ändern, die Zahlungsart und die Lieferadresse. Die letzte Seite der Bestellroutine zeigt nochmal alle Einstellungen übersichtlich auf einer Seite zusammen gefasst. Auf dieser Seite können alle Einstellungen nachträglich verändert werden. Die Übersichtseite entspricht somit den Gesetz des bürgerlichen Gesetzbuches §312g Absatz 1. Bei diesen Gesetz muss den Kunden die Möglichkeit gegeben werden nochmal vor dem Absenden der Bestellung alle Einstellungen bezüglich der Bestellung ändern zu können. Bei online Shops existiert auch eine „Ausgestaltungspflicht des Bestellbuttons“. Der Button zum Bestellabschluss muss aus den anderen Button hervorstechen und eindeutig beschriftet sein. Es muss also den Kunden klar werden, dass die Bestellung mit den Button abgeschlossen wird und diese kostenpflichtig ist. Bei diesem Webshop wurde der Button deswegen in einer anderen Farbe gefärbt und mit „Jetzt kostenpflichtig bestellen“ beschriftet. Der Kunde kann allerdings nur die Bestellung abschließen, wenn dieser die AGB des Webshop akzeptiert hat. Somit wird die AGB ebenso auf der Bestellübersicht angezeigt. Akzeptieren kann der Kunde die AGB indem dieser bei der Checkbox „Ich akzeptiere die AGB.“ einen Hacken setzt. Wie auf der Abbildung 40 zu sehen ist wurde bei der Übersichtsseite die Änderungsfunktionen über einen Hyperlink mit Stiftsymbol umgesetzt. Dieser Hyperlink leitet den Kunden zu der Seite der Bestellroutine zurück, wo die entsprechende Einstellung getätigter wurde.

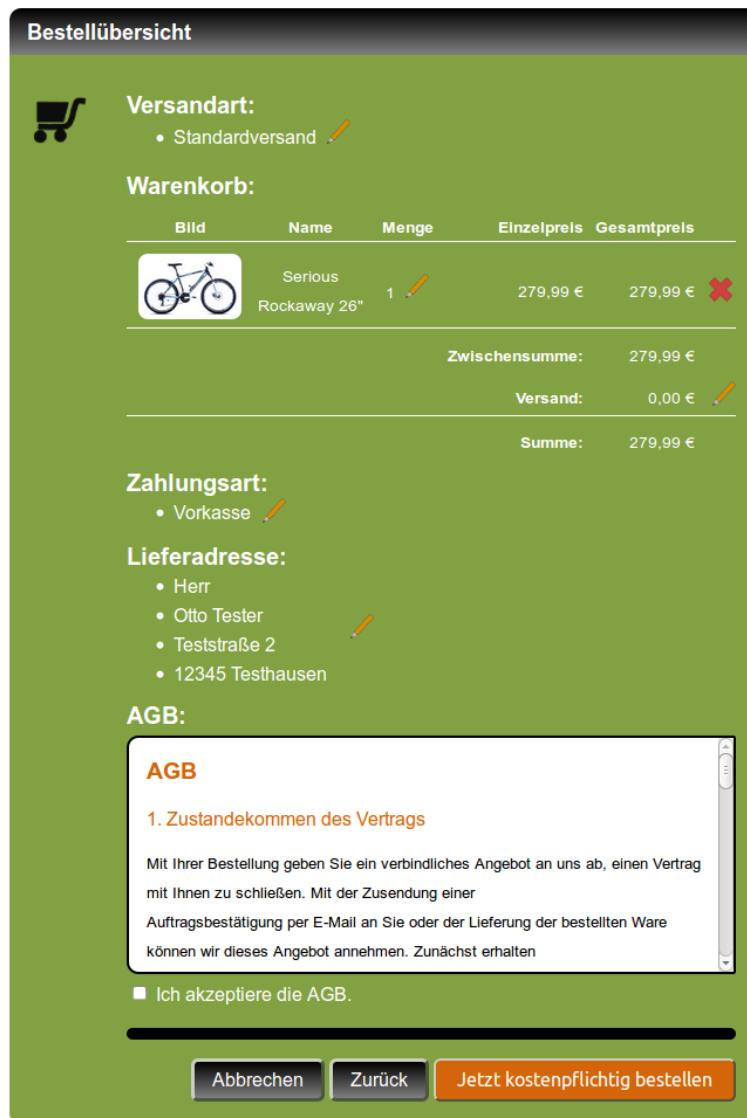


Abbildung 40: Bestellroutine: Bestellübersicht

Beim Klicken auf dem Bestellbuttons wird die Seite neugeladen. Die neugeladene Seite ruft nun die Methode „zwischengespeicherter_bestellungslisteneintrag_bestellen()“ auf, welche der Klasse „Bestellungen“ angehört. Diese Methode schließt den Bestellvorgang ab und speichert alle Einstellungen auf der Datenbank ab. Da es gesetzlich vorgeschrieben ist, dass unverzüglich nach dem Absenden der Bestellung diese bestätigt werden muss, versendet der Webshop somit sofort eine Bestätigungs-Email. Zum Absenden der Bestätigungs-Email wurde ebenfalls eine Methode in der Klasse „Bestellungen“ angelegt. Der Name dieser Methode lautet „sende_bestaeigungs_email()“. Als Content-Type der Email wurde HTML festgelegt. HTML bietet den Vorteil, dass die Email in den Design des Webshops gestaltet werden konnte und auch Tabellen, Farben und Bilder anzeigen kann. Über die eigentlichen HTML-Ansicht der Email wurde die Style-Anweisung der Email gepackt. Somit muss diese nicht erst vom Webserver geladen werden, wenn der Kunde die Email aufruft. Für

das Versenden der Email ist eine andere Klasse zuständig. Der Name dieser Klasse lautet „Email“. In dieser Klasse wurde eine Methode implementiert, die sofort schon den richtigen Header für eine Email mit den Content-Type „HTML“ schreibt und diese über die, in der „Config-Datei“ festgelegte Email-Adresse versendet. Der Name dieser Methode lautet „html_email_senden(\$empfaenger_adresse, \$betreff, \$html_code)“. Die Abbildung 41 zeigt einen Ausschnitt der Bestätigungs-Email.

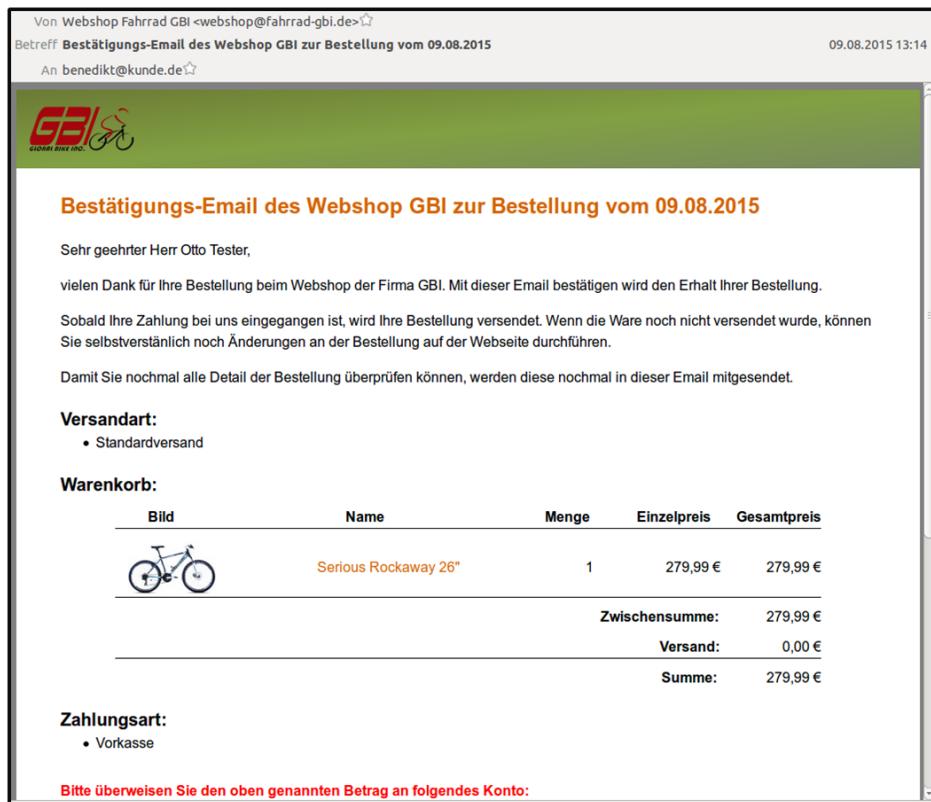


Abbildung 41: Bestätigungs-Email

Mit der Übersichtseite inbegriffen besteht die Bestellroutine aus fünf Seiten. Damit die Webseite möglich übersichtlich gestaltet ist, wird deswegen auf jeder Seite nur eine Einstellung vorgenommen. Bei der ersten Seite wird z.B. nur die Versandart festgelegt. Sind für die Einstellung Eingabefenster nötig, so werden diese, wie schon mehrfach erklärt client- und serverseitig überprüft. Es wurde sich dazu entschlossen immer nur eine Einstellung auf eine Seite zu packen, da bei bestehenden Anbietern, wie z.B. Amazon das „Vollpacken“ der Seiten der Bestellroutine als sehr unübersichtlich angesehen wurde.

4 Sicherheit

Zum Schutz des Webshop vor Angriffen wurden mehrere Sicherheitsmechanismen in die Seite eingebaut. Dieser Abschnitt erläutert welche Sicherheitsmechanismen verbaut wurden und warum diese verbaut worden sind.

4.1 „.htaccess“-Datei

Als eine mögliche Schwachstelle sah das Projektteam an, dass PHP-Dateien direkt über die Adressenleiste im Browser aufgerufen werden können. Es könnte zum Beispiel somit ein Benutzer durch direktes Aufrufen einer Datei, versuchen den Passwortschutz der Webseite zu umgehen. Dieser Benutzer wäre dann vielleicht in der Lage Produkte zu kaufen ohne registriert zu sein. Diese Schwachstelle wurde gelöst indem nur die Index-Datei direkt aufgerufen werden kann. Die Index-Datei erhielt somit als einzige Datei die Endung „.php“. Alle anderen PHP-Skripte bekamen die Dateiendung „.php.inc“. In einer sogenannten „.htaccess“-Datei wurde danach festgelegt, dass von Browser aus die „.php.inc“-Dateien nicht geöffnet werden können. Um die Skripte mit „.php.inc“-Endung ausführen zu können, werden diese zum vorgesehenen Zeitpunkt in die Index-Datei inkludiert. Die ganze Webseite wird somit komplett durch die Index-Datei abarbeitet.

Des Weiteren wurde mit der „.htaccess“-Datei festgelegt, dass der Browser nicht alle Dateien eines Verzeichnisses auflisten darf. Hiermit sollte sichergestellt werden, dass der unerwünschte Benutzer nicht die Ordnerstruktur der Webseite ermitteln kann.

4.2 Schutz vor: Cross-Site-Scripting (XSS) und SQL Injection

Als eine weitere Schwachstelle wurde Cross-Site-Scripting (XSS) und SQL Injection angesehen. Bei SQL-Injection versucht eine Angreifer Schadcode in die Datenbank einzuspielen. Hierzu nutzt der Angreifer den Aufbau einer SQL-Anweisung aus. Bei einer SQL-Anweisung werden die zu übergebenen Werte zwischen zwei Hochkomma geschrieben. Der Angreifer schreibt dann in das Textfeld, wovon die Werte in die Datenbank übernommen werden sollen, ein Hochkomma und ein Semikolon. Dadurch geht die SQL-Datenbank davon aus, dass die Datenbankanweisung schon an der Stelle zu Ende ist. In SQL können durch Semikolons mehrere Anweisungen hintereinander geschrieben werden. Somit wäre es möglich, dass der Angreifer nun zum Beispiel eine Anweisung in das Textfeld schreibt, dass die Datenbank gelöscht werden soll und alle Daten weg sind. Zum Schutz vor dieser Situation bietet das „MySQLi“-API die Methode „real_escape_string()“. Die gerade genannte Methode „escaped“ alle Hochkomma, sodass diese nicht mehr der SQL-Anweisung schaden können. Cross-Site-Scripting beschreibt einen ähnlichen Vorfall. Hier tippt der Angreifer einen JavaScript in die Textfelder ein. Dieses mal ist von den Angriff der Browser betroffen. Soll nun zu einem späteren Zeitpunkt die Texteingabe auf der Webseite ausgegeben werden, so wird im Browser der JavaScript ausgeführt. Zum Schutz vor einer solchen Situation bietet PHP die Funktion „htmlspecialchars“. Mit der Funktion „htmlspecialchars“ werden Steuerzeichen von HTML (&, ", <, >) durch die HTML-Ersatzschreibweise ersetzt. Die Ersatzschreibweise hat zu folge, dass der JavaScript nicht mehr ausgeführt wird. Der „Schadcode“ wird nun einfach als Text auf der Webseite ausgegeben.

5 Fazit

Nachfolgend soll noch einmal kurz darauf eingegangen werden, wie der Verlauf des Projektes zu beurteilen ist. Dabei wird sich auch zu der Motivation innerhalb der Gruppe und der Zusammenarbeit mit anderen Gruppen geäußert.

5.1 Motivation innerhalb der Gruppe

Die Motivation innerhalb der Gruppe war unterschiedlich. Während sich die Gruppenmitglieder Herr Schnürer und Herr Brüntrup stets bemühten die Arbeit an den Webshop voranzutreiben, lies die Motivation der beiden Gruppenmitglieder Herr Gregarek und Herr Dück etwas zu wünschen übrig. Bei letzteren war die Teilnahme am Projekt nicht immer vollständig vorhanden, was sich auch darin widerspiegelt, dass Herr Gregarek und Herr Dück nicht zu allen Treffen erschienen sind.

5.2 Zusammenarbeit mit anderen Gruppen

Die Zusammenarbeit mit anderen Gruppen gestaltete sich durchgehend positiv. Die wichtigsten Schnittstellen waren hierbei die Server-, SAP- sowie die APP-Gruppe. Zu jeder Gruppe herrschte ein reger Austausch, sodass entsprechende Schnittstellen leicht implementiert werden konnten.

5.3 Verlauf des Projektes

Abgesehen von kleinen Differenzen innerhalb der Gruppe hinsichtlich der Motivation, konnte das Projekt rechtzeitig vollendet werden. Durch der unterschiedlichen Arbeitsleistung bedingt, mussten jedoch einzelne Arbeitspakete neu verteilt oder mangels der nötigen Zeit gekürzt werden. Da die wichtigsten Funktionen, die der Webshop gemäß der Aufgabenerstellung erfüllen soll erfolgreich implementiert worden sind, lässt sich der Verlauf des Projektes abschließend durchaus positiv beurteilen.

6 Quellenverzeichnis

Vorlesungen

- Websdesign - Prof. Dr. rer. nat. Stefan Wolf
- Datenbanken - Prof. Dr.-Ing. Klaus Maas

Literatur

- ASMUTH, M. (2009): *Online-Projekte mit PHP und MySQL* - 3.Auflage - Bildungsverlag EINS
- REESE, G. & SCHULTEN, L. (2006): *MySQL - kurz & gut* - 2.Auflage - O'REILLY

Onlinequellen

- <http://php.net> (Stand: 29. August 2015)
- <http://selfhtml.org/> (Stand: 29. August 2015)
- <http://www.w3schools.com> (Stand: 29. August 2015)
- <http://www.css4you.de/wscss/css03.html> (Stand: 29.August 2015)