**Dokumentation**

**iBlog**

**Ann-Katrin Kottig**

**Anna Leinkenjost**

**Jan Raupach**

**Claudia Schäfers**

Hochschule Weserbergland

Studiengang Wirtschaftsinformatik

Studiengruppe: WI 44/14

Dozent: Till Glöggler

1. Inhaltsverzeichnis

[I Inhaltsverzeichnis I](#_Toc451179954)

[II Abbildungsverzeichnis I](#_Toc451179955)

[1 Vorgehen und Designentscheidungen 1](#_Toc451179956)

[2 Installationsanleitung 2](#_Toc451179957)

[3 Beschreibung der Datenbank 2](#_Toc451179958)

[4 Beschreibung der Anwendung 2](#_Toc451179959)

1. Abbildungsverzeichnis

**Es wurden keine Einträge für das Inhaltsverzeichnis gefunden.**

# Vorgehen und Designentscheidungen

Die Aufgabe dieser Projektarbeit bestand darin einen Mikro-Blog mithilfe von PHP zu programmieren. Der entstandene Blog trägt den Namen "iBlog".

Am Anfang des Projekts wurden zunächst alle erforderlichen Funktionalitäten, welche der Blog beinhalten soll, gesammelt. Diese haben wir von Anfang an in Muss- und Kann-Anforderungen unterteilt.

Zu den zwingend erforderlichen Funktionen gehörte, dass die zukünftige Anwendung mehrbenutzerfähig sein soll. Des Weiteren ist die Möglichkeit einen Blogeintrag zu erstellen und zu bearbeiten eine weitere Funktion, welche in die Anwendung integriert werden sollte. Ebenso ist eine Übersichtsseite zur Präsentation der Blogeinträge erforderlich.

Die Kann-Anforderungen setzten sich aus unterschiedlichen ergänzenden und neuen Funktionalitäten zusammen. Einige hiervon sollen in der neuen Anwendung bereits zu Beginn integriert sein, weitere sind für eine spätere Veröffentlichung geplant. Unteranderem gehört zu den Kann-Anforderungen das Senden einer Email als Anmeldungsbetätigung und bei Bedarf das Zusenden eines neuen Passworts, falls das alte vergessen wurde. Des Weiteren ist eine weitere Kann-Anforderung die Suche nach einem User, Thema oder Titel, sodass zu dem gesuchten Begriff die zutreffenden Beiträge angezeigt werden. Eine weitere Funktion, welche in die Anwendung integriert werden könnte, wäre die Möglichkeit einen Beitrag zu kommentieren. Ebenso zählt zu den Kann-Anforderungen, dass der Beitrag bewertet werden kann.

Im Anschluss an die Anforderungssammlung wurde ein Konzept zur Aufteilung der Funktionalitäten erstellt. Aus diesem wird ersichtlich, welche verschiedenen HTML-Seiten erforderlich sind um alle Funktionalitäten in den Blog zu integrieren. Dabei stellte sich heraus, dass es einen Login-Bereich geben muss, sowie einen Anmelde-Bereich zur Neuregistrierung. Um seine Anmeldung mit einem Bestätigungscode abzuschließen, wird auch eine Seite eingerichtet. Ebenfalls wird eine Seite für den Fall, dass der Benutzer sein Kennwort vergessen hat und eine Seite zur Änderung des Passworts eingerichtet. Des Weiteren müssen alle Blog-Einträge angezeigt werden, dies geschieht auf einer eigens dafür eingerichteten HTML-Seite. Auf dieser Übersichtsseite, wird den Nutzer zusätzlich ermöglicht nach Beiträgen, Themen oder auch Usern zu suchen. Um neue Beiträge erfassen zu können, sollte es einen eigenen Bereich für den User geben. Hier soll er die Möglichkeit haben alle seine Beiträge einzusehen und zu bearbeiten, sowie neue Beiträge zu erfassen.

Zu den geplanten HTML-Seiten erstellten wir Oberflächenentwürfe mit dem Tool Balsamiq Mockups (vergleiche Abbildungen x - y). Bei den Entwurf der Oberflächen, wurde stets darauf geachtet, dass jede Seite den gleichen Aufbau besitzt. Dadurch wird das Ziel verfolgt, dass der Nutzer den iBlog intuitiv bedienen kann und alles schnell findet.

Der Gesamtaufbau der Anwendung orientiert sich an dem Entwurfsmuster Modell-View-Controller. Hierzu wurde in verschiedenen php-Dateien, die jeweils eine einzelne Seite repräsentieren, alle benötigten HTML-Elemente hinterlegt. Diese orientieren sich an den zuvor erstellten Oberflächenentwürfen und dienen als View der Anwendung.

Es wurde zunächst im oberen Bereich der Seite eine Art Menüleiste implementiert. Diese dient dem Nutzer zur Navigation durch die Anwendung. Damit den Nutzer auch weiß auf welcher Seite er sich befindet, wurde unterhalb der Menüleiste ein Überschriftenbereich eingegliedert. Durch diesen Bereich sieht der Nutzer auf einem Blick, was ihm gerade auf der Seite angezeigt wird. Nachdem die Oberflächenentwürfe fertig gestellt waren, wurde anhand dieser Entwürfe das Front-End der Anwendung entwickelt. Hierbei wurde darauf geachtet, dass alle Schriften gut leserlich sind und ein guter Kontrast zwischen den Farben besteht. Bei der Gestaltung wurde kein CSS-Framework genutzt. Die Anwendung wurde ausschließlich vom Projektteam gestaltet. Diese Entscheidung fundiert auf der Einzigartigkeit und dem Corporate Design unseres iBlogs. Des Weiteren ist die Anwendung somit unabhängig und kann flexibel gestaltet werden.

Im nächsten Schritt wurde das Datenbankmodell erstellt. Hierbei wurden die Anforderungen an die Anwendung und die Verteilung der Funktionalitäten auf den einzelnen HTML-Seiten berücksichtigt. Es wurde eine Datenbank mit dem Namen project\_blog erstellt. Innerhalb dieser Datenbank liegen die Tabellen article, comment, ranking, theme, theme\_article, user und user\_likes\_article. Die Datenbank wurde in der dritten Normalform realisiert, um Redundanzen und Inkonsistenzen zu vermeiden. Außerdem wurde auf die Skalierbarkeit der Anwendung Wert gelegt. Beispiel dafür ist, dass das Datenbankmodell so ausgerichtet ist, dass einem Artikel mehrere Themen zugeordnet werden können. Zurzeit ist es im Rahmen der Anwendung lediglich möglich, einen Artikel in ein Thema zu einordnen

Im darauffolgenden Schritt wurde das Back-End entwickelt.

Während in den HTML-Seiten alle Oberflächenelemente enthalten sind, übernimmt die "index.php"-Datei die Aufgabe der Vermittlung zwischen der Oberfläche und dem Back-End. Über URL-Argumente wird der Kontrollfluss geregelt. Diese Argumente legen fest, welche Methoden aus dem Back-End aufgerufen werden.

Die Methoden wurden in den Klassen Template, Session, Mail, Article und Comment implementiert. Jeder dieser Klassen wurde eine Zuständigkeit zugewiesen. So übernimmt die Klasse Template das Rendern der Oberfläche. In der "Session.php" befinden sich alle benötigten Methoden mit Bezug zu der aktuellen Session und dem angemeldeten Benutzer. Des Weiteren beinhaltet die Klasse Get-Methoden für die Session-Variablen. In der Klasse Mail befinden sich alle Funktionen, über die Mails versandt werden, sowie Codes oder Passwörter generiert werden können. Über die Funktionen der Klasse Article und Comment können neue Artikel bzw. Kommentare erstellt, sowie alle Artikel und alle Kommentare aus der Datenbank abgefragt werden. Zusätzlich bietet die Article-Klasse Funktionen zum Suchen nach verschiedenen Artikeln über verschiedene Kriterien.

Da sich während der Bearbeitung --> Übereinstimmung von real und Entwürfe können abweichen, weil Anforderungen sich geändert haben?!

Im letzten Schritt fand die Qualitätssicherung statt. Hierzu wurden alle Funktionalitäten aus der Anforderungssammlung überprüft und die Sicherheit der Anwendung getestet. Bei fehlgeschlagenen Testfällen wurde der Code überprüft und gegebenenfalls verändert. Sobald ein Testfall fehl schlug, wurden alle Testfälle inkrementell ein weiteres Mal durchgeführt, um Seiteneffekte bei Veränderungen im Quellcode auszuschließen.

# Installationsanleitung

Innerhalb der Anwendung wird die Mail-Funktion von PHP verwendet, sodass Änderungen in den Konfigurationsdateien sendmail.ini und php.ini vorgenommen werden müssen

*sendmail.ini:*

|  |
| --- |
| ...  smtp\_server=smtp.gmail.com  ; smtp port (normally 25)  smtp\_port=587  ...  error\_logfile=error.log  ...  auth\_username= php.iBlog@gmail.com  auth\_password= iBlog123  ...  ; force the sender to always be the following email address  ; this will only affect the "MAIL FROM" command, it won't modify  ; the "From: " header of the message content  force\_sender= php.iBlog@gmail.com |

*php.ini:*

|  |
| --- |
| [mail function]  ; For Win32 only.  ; http://php.net/smtp  SMTP=smtp.gmail.com  ; http://php.net/smtp-port  smtp\_port=587  ; For Win32 only.  ; http://php.net/sendmail-from  sendmail\_from = php.iBlog@gmail.com  ; For Unix only. You may supply arguments as well (default: "sendmail -t -i").  ; http://php.net/sendmail-path  sendmail\_path = "\"C:\xampp\sendmail\sendmail.exe\" -t" |

Außerdem sollten die short\_open\_tags in der Datei php.ini per default auf "on" gesetzt werden.

Datenbank installation

# Beschreibung der Datenbank

# Beschreibung der Anwendung

A

**Eigenständigkeitserklärung**

„Wir versichern hiermit, dass wir diese Arbeit selbständig verfasst, keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht haben. Das gleiche gilt auch für eingefügte Zeichnungen, Kartenskizzen und Darstellungen.“

|  |  |
| --- | --- |
| Hameln, 21. Mai 2016 |  |
| Ort, Datum | Ann-Katrin Kottig |

|  |  |
| --- | --- |
| Hameln, 21. Mai 2016 |  |
| Ort, Datum | Anna Leinkenjost |

|  |  |
| --- | --- |
| Hameln, 21. Mai 2016 |  |
| Ort, Datum | Jan Raupach |

|  |  |
| --- | --- |
| Hameln, 21. Mai 2016 |  |
| Ort, Datum | Claudia Schäfers |