

Projektteam: Petar Barisic und Sascha Blank

Dokumentation Teil 2

Pics Are Love

# Pics Are Love

Auf Pics Are Love sollen Bilder hochgeladen, kategorisiert und kommentieren kann. Es sollen eine Liste von Kategorien bestehen in die die Bilder eingeordnet werden können. Auch sollen Benutzer die Bilder kommentieren können. Bilder können von dem Benutzer gelöscht werden der das Bild hochgeladen hat. Zufällige Bilder können angezeigt werden. Ausserdem soll man in der Lage sein die Bilder nach Kategorien zu filtern.

Inhalt

Pics Are Love 1

Technische Dokumentation 3

Code Style 3

Testauswertung 4

Installationsanleitung 4

Systemanforderungen 4

Benutzerhandbuch 5

Fazit 6

# Technische Dokumentation

## Code Style

Klassennamen sind gross geschrieben z.b. class Somthing  
Funktionen sind kleingeschrieben mit Camelcase z.b function getSomthing()  
Funktionsparameter haben den Präfix p\_. z.b $p\_somevalue  
Generell ist bei der Bennenung immer Camelcase zu verwenden.  
Kommentiert wird anhand des Javadoc Syntax.

Beispiel einer Klasse:

/\*\*

\* Irgendeine Klassen Beschreibung

\* **@author** Sascha Blank

\*/

**class** Somthing {

// Memberbrescheibung

**private** $value = 1;

/\*\*

\* Konstruktor

\* **@param** unknown\_type $p\_someValue, beschreibung der Variable

\*/

**function** \_\_construct( $p\_someValue) {

$this->value = $p\_someValue;

}

/\*\*

\* Getter Function fuer this->value

\* **@return** int value

\*/

**function** getValue(){

**return** $this->value;

}

/\*\*

\* Setter fuer $this->value

\* **@param** int $p\_value, Beschreibung der Variable

\*/

**function** setValue( $p\_value ) {

$this->value = $p\_value;

}

}

## Testauswertung

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| ST-01 | Ja |  |
| ST -02 | Ja |  |
| ST -03 | Ja |  |
| ST -04 | Ja |  |
| ST -05 | Ja |  |
| ST -06 | Ja |  |
| ST -07 | Ja |  |
| ST -08 | Ja |  |
| ST -09 | Ja |  |
| ST -10 | Ja |  |

Ein bis zwei Sätze, welche die Testresultate zusammenfassen.

## Installationsanleitung

### Systemanforderungen

* Apache 2.4
* PHP 7
* HTML 5
* CSS 3
* Alle gängigen Browser

### Installationsanleitung

Als erstes muss der Apache Server wie folgt konfiguriert werden.  
In der httpd.conf Datei des Apache Server müssen die Werte DocumentRoot und Directory  
gesetzt werden. In diesen Werten muss der root Pfad der Webseite hinterlegt werden.  
Das root Verzeichnis ist das Verzeichnis in dem die index.php der Webseite liegt.  
Diese index.php liegt im «code» Verzeichnis. Das «code» Verzeichnis stellt also das root Verzeichnis dar. Allenfalls muss noch die «Index.html» im Hostverzeichnis des Hosters gelöscht werden.  
Ausserdem muss noch das Secure Cookie aktiviert werden. Um dies zu bewerkstelligen sind 2 Schritte nötig:   
1.) LoadModule headers\_module modules/mod\_headers.so. Diese Zeile darf keine Raute als erstes Zeichen habe (Raute bedeutet das diese Zeile auskommentiert ist). Dies sollte aber schon Standartmässig der Fall sein.  
2.) Header edit Set-Cookie ^(.\*)$ $1;HttpOnly;Secure. Diese Zeile muss am Ende der httpd.conf Datei eingefügt werden.  
Der Server muss nachdem die Konfiguration abgeschlossen ist neu gestartet werden.

Der nächste Schritt ist die Konfiguration der Datenbank.  
Als erstes muss ein neuer Benutzer angelegt werden.  
Die folgenden Daten werden benötigt um dies zu tun:  
Host = localhost  
Username = bblanssql1  
passwort = Cpp4ever  
Folgende Rechte muss de r Datenbank Benutzer haben:  
- SELECT  
- INSERT  
- UPDATE  
-DELETE  
-FILE  
Sprich alle «Daten» Rechte im phpMyAdmin.  
Erstelle eine neue Datenbank mit den Namen picsarelove und lasse dann als   
letzten Schritt das picsarelove.sql Script ausführen um die Datenbank mit den entsprechendem Tabellen zu erstellen.  
Falls der Upload der Bilder nicht funktionieren sollte muss noch die Schreibberechtigung gesetzt werden.

# Benutzerhandbuch

* Menüführung
  + Registrieren 🡪 Öffnet eine neue Seite auf der sich ein Benutzer registrieren kann.
  + User Verwaltung 🡪 Hier kann der Benutzer sein Passwort neu setzten und von ihm hochgeladene Bilder wieder löschen.
  + Alle anzeigen 🡪 Zeigt alle Bilder, das neuste Bild wir zuerst angezeigt. Der Benutzer kann sich so die ganze Bildersammlung anzeigen lassen.
  + Zufälliges Bild 🡪 Zeigt ein zufälliges Bild an.
  + Kategorien 🡪 Wechselt in die ausgewählte Kategorie. Es werden nur Bilder der entsprechenden Kategorie angezeigt.

Die CRUD Elemente wurden wie folgt umgesetzt

* Das Create Element wird beim Registrieren, Erstellen von Bildern und dem Erstellen von Kommentaren benutzt.
* Das Read Element wird beim Anzeigen von Bildern und Kommentaren sowie beim Login verwendet.
* Das Update Element wird verwendet, wenn man sein Passwort neu setzt.
* Das Delete Element wird beim Löschen der eigenen Bilder benutzt.

# Fazit

Das Vorankommen war sehr speditiv und wir hatten den Grundpfeiler der Logik schon nach 2 Tagen fertig implementiert. Petar hatte zu Anfang Mühe das MVC Muster zu verstehen. Er war aber in der Lage diese Lücke nach 3 Tagen zu schliessen.

Grundsätzlich sind wir mit dem Ergebnis zufrieden. Einzig die Darstellung könnte ansprechender sein. Aber wir haben das von uns ausgesuchte Farbkonzept einigermassen gut umgesetzt.

Wir haben sehr viel über PHP und das MVC Muster gelernt. Auch haben wir uns mit der Sicherheit der Webseite auseinandersetzten müssen. Dies war ebenfalls sehr lehrreich.

Wir konnten zudem alle von uns konzipierten Funktionalitäten implementieren. Auch die Zeitplanung war gut durchdacht was sicher ein zusätzlicher Grund war weshalb wir pünktlich fertig geworden sind.