|  |
| --- |
| Modul 151: Projekt Memepage |
| Planen und Erarbeiten einer Web Applikation mit integrierter Datenbank |
|  |
| **Dueen Brunner, IN15-19C** |
| **21.12.2018** |

|  |
| --- |
| [Geben Sie hier das Exposee für das Dokument ein. Das Exposee ist meist eine Kurzbeschreibung des Dokumentinhalts. Geben Sie hier das Exposee für das Dokument ein. Das Exposee ist meist eine Kurzbeschreibung des Dokumentinhalts.] |

Inhalt

[1 Informieren 3](#_Toc533360863)

[1.1 Auftrag 3](#_Toc533360864)

[1.2 Projektvorgaben 3](#_Toc533360865)

[1.3 Technische Vorgaben 3](#_Toc533360866)

[2 Planen 4](#_Toc533360867)

[2.1 Grundidee Web-Applikation 4](#_Toc533360868)

[2.2 Werkzeuge 4](#_Toc533360869)

[2.2.1 Webserver / Datenbankserver 4](#_Toc533360870)

[2.2.2 IDE 4](#_Toc533360871)

[2.2.3 Code Verwaltung 4](#_Toc533360872)

[2.3 Pflichtenheft 5](#_Toc533360873)

[2.4 Zeitplan 6](#_Toc533360874)

[3 Entscheiden 7](#_Toc533360875)

[3.1 Template Engine 7](#_Toc533360876)

[3.2 Design 7](#_Toc533360877)

[3.3 JavaScript 7](#_Toc533360878)

[4 Realisierung 8](#_Toc533360879)

[4.1 Programmablaufplan Routing 8](#_Toc533360880)

[4.2 Klassendiagramm 9](#_Toc533360881)

[4.2.1 Model 9](#_Toc533360882)

[4.2.2 Controller 10](#_Toc533360883)

[4.2.3 View 10](#_Toc533360884)

[4.2.4 Repository Pattern 11](#_Toc533360885)

[4.2.5 Application 11](#_Toc533360886)

[4.3 Datenmodell 12](#_Toc533360887)

[4.4 Sicherheit 12](#_Toc533360888)

[5 Kontrolle 12](#_Toc533360889)

[6 Auswertung 12](#_Toc533360890)

[7 Fazit 12](#_Toc533360891)

[8 Quellenangabe 12](#_Toc533360892)

# Informieren

## Auftrag

Im Rahmen der LBV Modul 151-1 ist der Lehrling zu einer Projektarbeit verpflichtet, nämlich das Entwerfen und Realisieren einer Web Applikation mit integrierter Datenbank nach einem selbst definierten Pflichtenheft.

Terminiert und abgegeben, mitsamt Dokumentation und Quellcode, wird das Projekt spätestens am **23.Dezember 2018 um 23:55**.

## Projektvorgaben

Die formellen Vorgaben des Projekts wurden durch den Lehrer Alexander Schenkel erteilt und lauten wie folgt:

* Die Projektarbeit wird in Einzelarbeit umgesetzt.
* Sie definieren selber ein Pflichtenheft zu Ihrer Applikation. Dies kann z.B. ein BlogSystem, ein Web-Forum, eine Medien- resp. Bilderverwaltung oder ein RSS-FeedReader sein. Eigene Vorschläge sind willkommen, werden aber mit der Lehrperson abgesprochen.
* Wählen Sie die Applikation nicht zu umfangreich: Das Projekt muss zum Schluss des Semesters abgeschlossen sein. Die Applikation sollte ca. aus 5-10 Haupt-Modellen (z.B. Benutzer, Blog-Eintrag, Kategorien, Tags etc.) bestehen.
* Die Applikation muss mind. verschiedene Benutzer via Login-Mechanismus unterscheiden.
* Sie haben während des Unterrichts jeweils mind. 1 Lektion pro Woche Zeit, an Ihrem Projekt zu arbeiten.
* Sie dürfen alle Informationen verwenden: Unterrichtsunterlagen, Internet. Bei Unklarheiten oder grösseren Problemen wenden Sie sich an die Lehrperson.

## Technische Vorgaben

Neben den formellen Vorgaben des Projekts, sind auch bestimmte technische Vorgaben für die Web-Applikation gegeben:

* Als Server-Programmiersprache wird PHP 7 auf einem Apache Web Server eingesetzt.
* Für das Frontend wählen Sie HTML/CSS/JavaScript.
* Als Datenbank wird eine MySQL-Datenbank eingesetzt.
* Sie setzen kein PHP-Web-Framework ein: Sie entwerfen ein eigenes Web-MVCFramework. Die Grundlagen dazu werden in den ersten Unterrichtslektionen erarbeitet.
* Zusätzliche Bibliotheken oder Paket-Manager (z.B. php composer, Template-Engine) dürfen eingesetzt werden.
* Frontend-Bibliotheken / Frameworks (z.B. Bootstrap CSS Framework) sind erlaubt

# Planen

## Grundidee Web-Applikation

Für die Projektarbeit des Moduls erarbeite ich ein Web-Applikation, bei der sich Benutzer registrieren können und Medien, hauptsächlich Bilder, veröffentlichen können. Es wird ein öffentliches Dashboard zur Verfügung stehen, auf dem alle Benutzer die veröffentlichen Bilder, welche mit einem kurzen Titel und einer Beschreibung versehen werden, einsehen können. Zusätzlich kann ein einzelnes Bild aufgerufen werden, um eine Detailansicht zu öffnen, bei der es auch für die Benutzer möglich ist Kommentare zu hinterlassen.

## Werkzeuge

Für die Realisierung des Projekts werden einige Tools benutzt.

### Webserver / Datenbankserver

Bereits bei früheren Projekten wurde das Programm Xampp benutzt um einen Web- und Datenbankdienst lokal zur Verfügung zu stellen. Auch in diesem Projekt wird erneut auf Xampp zurückgegriffen.

Die Xampp Distribution kommt mit PHP Version 7.2.13 und PhpMyAdmin Version 4.7.0.

### IDE

Als Programmierumgebung wird Microsoft Visual Studio benutzt mit folgenden Plugins:

* PHP Debug
* PHP IntelliSense
* Phpfmt

### Code Verwaltung

Da auf mehreren Geräten codiert wird, muss eine Online Code Verwaltung eingerichtet werden. Für diesen Zweck wird GitKraken zur Hilfe genommen.

## Pflichtenheft

Teil des Projektes ist die Selbstbestimmung eines Pflichtenhefts. Ich habe mir für die gesamte Web-Applikation folgende Punkte zusammengefasst, welche es zu realisieren gibt.

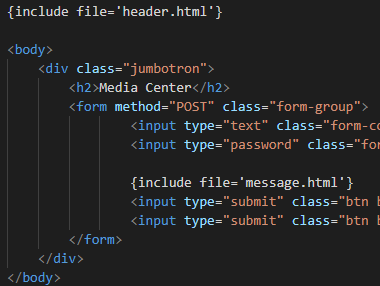
|  |  |  |
| --- | --- | --- |
| Nummer | Funktion | Beschreibung |
| 1 | Login Mechanismus | Ein Benutzer muss in der Lage sein einen persönlichen Account einrichten zu können. |
| 2 | Medien Upload | Ein eingeloggter Benutzer kann ein Bild, dass ein bestimmtes Dateiformat enthält, veröffentlichen. |
| 3 | Dashboard | Auf der Hauptseite wird ein Dashboard gezeigt, welches alle veröffentlichte Bilder in einer bestimmten Darstellungsweise anzeigt. |
| 4 | Persönliche Medien Ansicht | Ein Benutzer kann das Dashboard auf seine eigenen Medien beschränken. |
| 5 | Detailansicht | Es ist möglich ein Medium in einer Detailansicht zu öffnen |
| 6 | Kommentarfunktion | Auf einer Detailansicht können Benutzer Kommentare hinterlassen |

## Zeitplan

Siehe Zeitplan.xlsx

# Entscheiden

## Template Engine

Um die HTML Seiten einfach zusammensetzen zu können und einzelne Ausschnitte wiederverwendbar zu machen muss eine Template Engine eingesetzt werden. Ich habe mich in diesem Fall für Smarty entschieden, da es eine sehr simple Syntax besitzt.

## Design

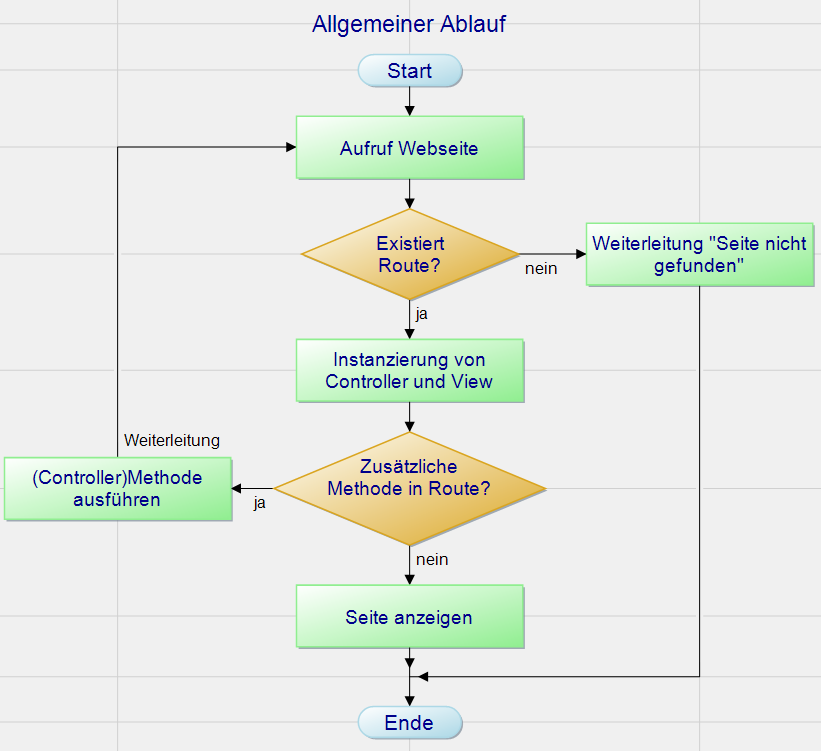
Da das Design der Web-Applikation eher im Hintergrund steht und nur soweit ausgebaut werden muss, damit das Benutzen möglich ist, wird hier auf normales Bootstrap zurückgegriffen.

## JavaScript

Auf den Einsatz von JavaScript wird in diesem Projekt komplett verzichtet, da es zum einen keine Anforderung ist, und zum anderen nur unnötige Komplexität reinbringt.

# Realisierung

## Programmablaufplan Routing

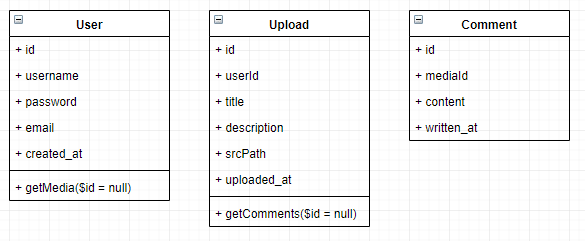
Hier wird ein grober Programmablauf für das Aufrufen der Webseite dargestellt. In der Router Klasse wird ein Dictionary gehandhabt, welches alle möglichen Routen mit der zuständigen Controller Klasse und einer optionalen Methode beinhaltet.



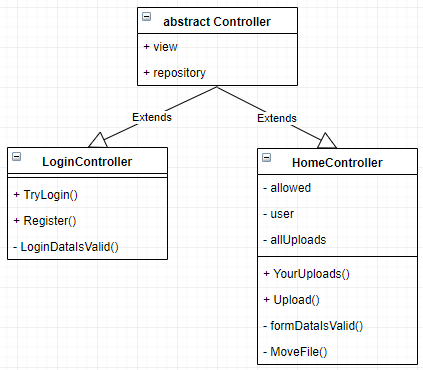
## Klassendiagramm

In diesem Abschnitt werden die eingesetzten Klassen in UML Form dargestellt und anschliessend deren Funktion erklärt.

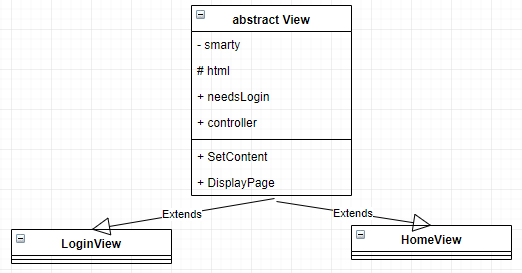
### Model

Diese drei Model Klassen stellen die Datenbank Entities und den Inhalt der Web-Applikation dar.

### Controller

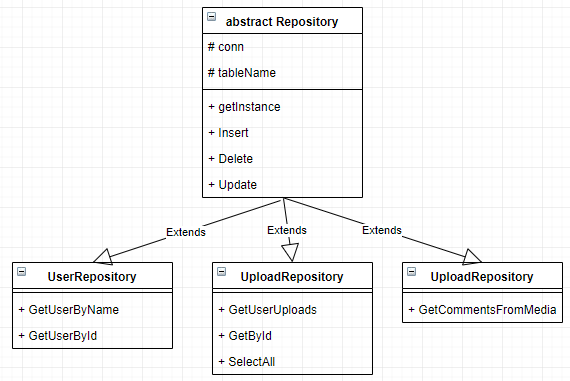
In der Superklasse Controller befinden sich zwei Properties View und Repository welche jeweils im Konstruktor des entsprechenden Controllers definiert werden. Die ableitenden Controller beinhalten spezifische Methoden, welche über den Router abgerufen werden.

### View

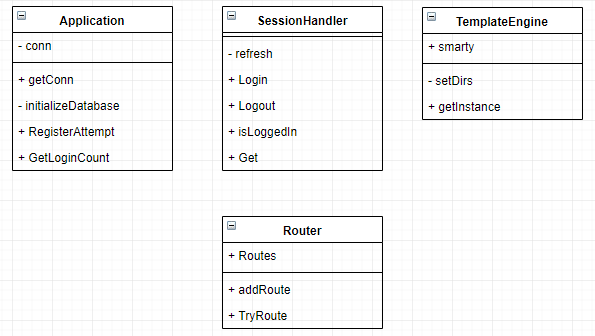
Die Superklasse View beinhaltet alle Information für das Front End der Applikation. Über die Smarty Variable wird, zusammen mit den Funktionen SetContent und DisplayPage, die gesamte Anzeige verwaltet. Die html Variable wird im Konstruktor der jeweiligen View definiert. Mit der Property needsLogin kann beim Routing überprüft werden, ob ein User zugreifen darf auf die Seite oder nicht.

### Repository Pattern

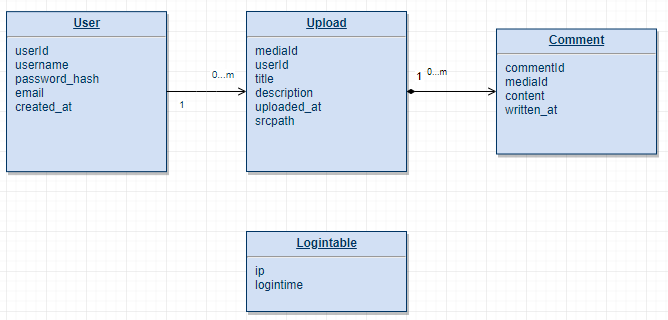
Das Repository Pattern übernimmt die ganze Arbeit für Datenbank Zugriffe, damit dies zentral an einem Ort statt findet. Die Superklasse Repository definiert hier die Funktionen Insert, delete und Update, da diese in allen Tables gleich funktionieren. Die Ableitenden Repositories definieren dann erweiterte Datenbank Abfragen.



### Application

Diese Klassen sind für den Ablauf der Medienseite zuständig.

## Datenmodell

Das Datenbank Modell besteht aus insgesamt vier Tables. Die Tabellen User, Upload und Comment speichern die Entities des Projekts ab. Die Tabelle Logintable speichert Loginversuche mit der IP Adresse des Sendegeräts und der Zugriffszeit.

## Sicherheit

# Kontrolle

# Auswertung

# Fazit

# Quellenangabe