

SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KIV/WEB

Webová aplikace programátorské konference

Patrik Harag

harag@students.zcu.cz

(A15B0034P)

14. prosince 2016

1 Použité technologie

Apache + PHP + MariaDB (WAMP)

Twig Šablonovací systém. Každá stránka využívá nějakou šablonu. Šablony tvoří hierarchii.

Bootstrap *Front-end* framework. Využíván je především pro vzhled, layout, ovládací prvky, formuláře, dialogy. . .

JavaScript + jQuery Použité pro větší interaktivitu webových stránek (např. počítadlo zbývajících znaků), nezbytné pro inicializaci některých ovládacích prvků (např. dialogy). *jQuery* je také závislostí dalších použitých technologií.

AJAX Použit pro kontrolu dostupnosti uživatelského při registraci.

starr Mini knihovna přidávající komponentu pro hodnocení hvězdičkami. Recenzenti pomocí této komponenty hodnotí publikace.

Font Awesome Umožňuje vkládání vektorových ikon do webu. Použito hlavně kvůli ikoně hvězdičky.

CKEditor Jedná se o *WYSIWYG* HTML editor. Administrátor může pomocí tohoto editoru upravovat obsahové stránky.

2 Popis adresářové struktury

- *lib* – obsahuje knihovny a další závislosti. Projekt nemá žádné externí závislosti, vše potřebné je zde.
- *model* – vrstva pracující s databází a obsahující doménovou logiku.
- *view*
 - *templates* – obsahuje šablony napsané v šablonovacím systému *Twig*.
 - *css* – obsahuje kaskádové styly.
 - *resources* – obsahuje obrázky. Případně by obsahoval i jiná potřebná data.
- *controller* – obsahuje kontrolery pro všechny stránky.

3 Architektura aplikace

Webová aplikace se striktně drží architektury MVC.

3.1 Model

Model pracuje s databází a obsahuje doménovou logiku. Ostatním částem aplikace poskytuje API, pomocí kterého mohou manipulovat s daty.

Pro každý typ dat *model* poskytuje neměnnou třídu:

- *Article*
- *User*
- *Publication*
- *Review*
- *Comment*

Ke každé takové třídě se váže třída obsahující statické metody pro práci s tímto typem dat:

- *Articles*
- *Users*
- *Publications*
- *Reviews*
- *Comments*

Například *Users::loadAllUsers()* načte všechny uživatele jako kolekci typů *User*. Třídy *User* a *Users* jsou pro jednoduchost a snadné použití uloženy v jedné třídě nazvané *users.php*. A podobně ostatní třídy...

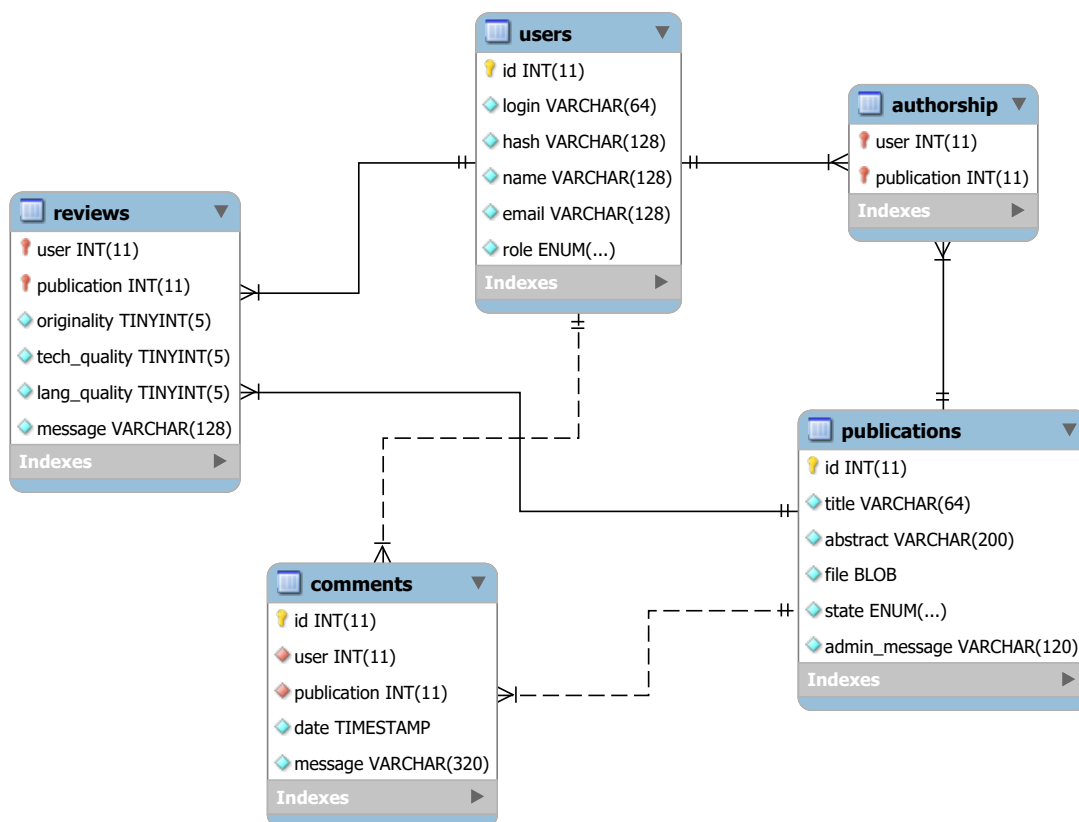
Model dále obsahuje třídu zajišťující spojení s databází a třídu pro kontrolu přihlášení uživatele.

3.2 View

View zahrnuje v podstatě jen šablony a *CSS*.

Základní šablonou je *layout.twig*, která definuje rozložení a vzhled stránky. Ostatní šablony od ní dědí a definují pouze obsah článku.

Šablonám jsou často předávány datové třídy z modelu. *Twig* totiž dokáže pracovat s třídami stejným způsobem jako s poli, kdy není nutné psát počáteční *get* a závorky, ale jen název příslušné *property*. Nevzniká tak žádná závislost.



Obrázek 1: ER diagram databázové vrstvy.

3.3 Controller

Zahrnuje hlavní kontroler v souboru *index.php* a kontrolery jednotlivých stránek, které jsou všechny třídy, dědící od *Page*.

Hlavní kontroler podle parametru *page* předaným metodou *GET* nebo *POST* vybere příslušnou třídu a vytvoří její instanci. Otestuje, zda má uživatel právo přístupu k této stránce, prostřednictvím metody *hasAccess(\$user)*, a poté ji zobrazí metodou *show()*. Pokud uživatel nemá právo přístupu, zobrazí jinou stránku.

Kontrolery jednotlivých stránek v zásadě jen (částečně) validují vstupy, předávají vstupy modelu a data získaná z modelu nechají zobrazit pomocí šablony nebo po předání dat modelu přesměrují na jinou stránku.

4 Závěr

Vytvořili jsme fungující webovou aplikaci programátorské konference. Aplikace je navržena spíše pro menší počet uživatelů. Pro větší počet uživatelů by bylo nutné upravit webové rozhraní tak, aby bylo možné rozumně pracovat i s většími daty (například filtrování, načítání jen části záznamů...).

Repozitář s projektem:

<https://github.com/Hartrik/KIV-WEB>