

**Webové stránky konferenčního systému**

*KIV/WEB – Semestrální práce*

student: Martin Folejtar

studijní číslo: A17B0206P

email: mfolejtar@gmail.com

datum: 14. 1. 2019

Obsah

[Obsah 2](#_Toc535342085)

[Zadání semestrální práce 3](#_Toc535342086)

[Popis použitých technologií 4](#_Toc535342087)

[Front-end (view) 4](#_Toc535342088)

[Back-end (controller, model) 4](#_Toc535342089)

[Popis adresářové struktury aplikace 5](#_Toc535342090)

[Popis architektury aplikace 7](#_Toc535342091)

[Digram tříd 8](#_Toc535342092)

Zadání semestrální práce

Jedná se o standartní zadání semestrální práce, která je uvedena na univerzitních stránkách courseware:

<https://courseware.zcu.cz/portal/studium/courseware/kiv/web/samostatna-prace/index.html>

Má práce se liší od zadání jen tím, že je na téma virtuální technologie a všechny potřebné body pro splnění semestrální práce obsahuje.

Popis použitých technologií

V mé semestrální práci jsem použil tyto technologie:

* HTML
* Kaskádové styly (CSS) – využití bootstrapu v3.7.7
* PHP skriptovací jazyk
* SQL jazyk

Celá semestrální práce je postavena na MVC architektuře, podle té se odvíjí, kde se jednotlivé technologie používají.

Front-end (view)

Pro realizaci front-endu jsem se rozhodl využít framework Bootstrap. Tam, kde dochází ke generování dat v závislosti na uživateli (př. vlastní příspěvky), tak se používá PHP jazyk. Dále, kde se očekává vstup uživatele neboli manipulace, tak se používají formuláře značkovacího jazyka HTML.

Back-end (controller, model)

Pomocí controlleru jsou data, popř. akce, které uživatel provedl částečně nebo úplně zpracována. Dále, pokud to situace vyžaduje, tak jsou data předána modelu, který komunikuje s databází.

Model data ukládá a získává z databáze, ke které je připojen.

Controller a model je napsán v PHP jazyce. V modelu navíc využíváme SQL jazyka pro komunikaci databáze.

Popis adresářové struktury aplikace

Zde je adresářová struktura:

KonferenceVR>src

* config
  + database.conf.php
  + session.conf.php
  + signpost.conf.php

Tento adresář je zaměřen na konfigurace neboli nastavení. Jsou zde také veškeré konstanty, které se využívají napříč aplikací.

* Libs
  + controller.class.php
  + database.class.php
  + model.class.php
  + session.class.php
  + signpost.class.php
  + user-current.class.php
  + view.class.php

V tomto adresáři jsou důležité třídy. Je zde rozcestník signpost.class.php, který se stará o zobrazování správných stránek. Dále jsou zde pomocné třídy, od kterých další třídy dědí.

* Controllers

Každá stránka má svůj kontrolér, který se o ní stará, díky dědičnosti, tak není nutné se opakovat v každém controlleru stránky.

* Models

Také zde se vyskytuje jednotlivě třída pro každou stránku. Třídy představují model v architektuře MVC. Díky tomu to rozdělení je velký přehled nad celou aplikací a nad stránkami komunikující s databází. A pokud bychom chtěli, aby byla využita funkce pro více stránek tak ji zahrneme do mateřské třídy model.class.php.

* View

> content

* + - * article.php
      * articleForRating.php
      * error.php
      * index.php
      * login.php
      * myarticle.php
      * publicArticles.php
      * register.php
      * serviceArticles.php
      * serviceUsers.php
  + footer.php
  + header.php
  + index.php

V tomto adresáři View se vyskytují všechny stránky, které se zobrazují. index.php je úvodní stránka.

* public
  + V tomto adresáři se vyskytují css styly bootstrapu a jeho potřebné soubory, jako je javascript a fonty písmen.
* uploads
  + Sem se ukládají nahrané pdf soubory k příspěvkům.

Popis architektury aplikace

Tak jak už bylo psáno výše, celá aplikace je postavena na MVC architektuře. Všechny třídy jsou intuitivně rozděleny do složek. Každá stránka ve webové aplikace má svůj controller a svůj model. Je to tak lepší pro přehlednost a pochopení celé funkčnosti.

Hlavní složky:

Libs

* + - Nachází se zde třídy controller, model, view od kterých jednotlivé specifické třídy dědí. Dále je zde rozcestník signpost.class.php, který spouští dané stránky.

View

* + - Zde se vyskytují soubory, kde každý soubor představuje jednu stránku v aplikaci.
    - Každá stránka je zobrazeno pod hlavičkou, kterou načítáme ze souboru header.php.

Controllers

* + - Zde se vyskytují soubory, které představují kontrolér každé samostatné stránky

Models

* + - Zde se vyskytují soubory, které představují tzv. model, který komunikuje s databází. Každá stránka má svůj model, pokud ho potřebuje.

Pro lepší znázornění architektury přikládám na další stránku diagram tříd.

Digram tříd

