

# KIV/WEB

## KONFEREČNÍ SYSTÉM

Martin Hamet  
A14B0205P  
hamet@students.zcu.cz

14.ledna 2019

# Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Analýza problému a návrh řešení</b>	<b>3</b>
2.1	MVC . . . . .	3
2.1.1	Model . . . . .	3
2.1.2	View . . . . .	3
2.1.3	Controller . . . . .	3
2.2	Adresářová struktura . . . . .	3
<b>3</b>	<b>Použité technologie</b>	<b>4</b>
3.1	MySQL . . . . .	4
3.2	PHP (Hypertext Preprocessor) . . . . .	4
3.3	HTML5 (HyperText Markup Language) . . . . .	4
3.4	JavaScript . . . . .	5
3.5	CSS (Cascading Style Sheets) . . . . .	5
3.6	W3.CSS . . . . .	5
<b>4</b>	<b>Implementace</b>	<b>5</b>
4.1	Controller . . . . .	5
4.2	Model . . . . .	6
4.3	View . . . . .	6
<b>5</b>	<b>Závěr</b>	<b>7</b>

# 1 Zadání

Vytvoříte kompletní webovou aplikaci konferenčního systému od začátku až do konce.

- Vaším úkolem bude vytvořit webové stránky konference. Téma konference si můžete zvolit libovolné.
- Uživatelé systému jsou autoři příspěvků (vkládají abstrakty článků a PDF dokumenty), recenzenti příspěvků (hodnotí příspěvky) a administrátoři (spravují uživatele, přiřazují příspěvky recenzentům k hodnocení a rozhodují o publikování příspěvků). Každý uživatel se do systému přihlašuje prostřednictvím vlastního uživatelského jména a hesla. Nepřihlášený uživatel vidí pouze publikované příspěvky.
- Nový uživatel se může do systému zaregistrovat, čímž získá status autora.
- Přihlášený autor vidí svoje příspěvky a stav, ve kterém se nacházejí (v recenzním řízení / přijat +hodnocení / odmítnut +hodnocení). Svě příspěvky může přidávat, editovat a volitelně i mazat. Rozhodnutí, zda autor může editovat či mazat publikované příspěvky je ponecháno na tvůrci systému.
- Přihlášený recenzent vidí příspěvky, které mu byly přiděleny k recenzi, a může je hodnotit (nutně alespoň 3 kritéria hodnocení). Pokud příspěvek nebyl dosud publikován, tak své hodnocení může změnit.
- Přihlášený administrátor spravuje uživatele (určuje jejich role a může uživatele zablokovat či smazat), přiřazuje neschválené příspěvky recenzentům k hodnocení (každý příspěvek bude recenzován minimálně třemi recenzenty) a na základě recenzí rozhoduje o publikování nebo odmítnutí příspěvku. Publikované příspěvky jsou automaticky zobrazovány ve veřejné části webu.
- Vypíše strukturu adresářové struktury včetně souborů.
- Databáze musí obsahovat alespoň 3 tabulky, které budou dostatečně naplněny daty tak, aby bylo možné předvést funkčnost aplikace.

## 2 Analýza problému a návrh řešení

### 2.1 MVC

Systém by měl dodržet architekturu MVC(Model View Controller). Bude tedy nutné oddělit data jejich zpracování a samotný kód stránky.

Vstupním bodem systému je tedy soubor `index.php`, který na základě parametrů volá příslušný controller. Controller podle požadavku získá příslušná data z modelu a předá je příslušnému view, které vytvoří vlastní stránku. Controller vrátí vytvořenou stránku do `index.php` a stránka se zobrazí uživateli.

#### 2.1.1 Model

Model zajišťuje pouze logiku práce s daty. Jedná se tedy o prostředek získání a zápisu dat do databáze.

#### 2.1.2 View

View zajišťuje zobrazení dat na stránce. Jedná se tedy o generátor vzhledu stránky, který ji na základě dat z controlleru vytvoří.

#### 2.1.3 Controller

Controller funguje jako prostředník mezi uživatelem a systémem. Controller tedy reaguje na požadavky uživatele. Pomocí modelu získává příslušná data, která předá příslušnému view.

### 2.2 Adresářová struktura

V kořenovém adresáři se bude nacházet pouze vstupní bod systému `index.php`, `.css` soubory a pomocná logovací třída `log.class.php`.

Dále bude adresářová struktura rozdělena logicky podle architektury MVC.

**Controller** Obsahuje všechny controllery. Každý controller bude potomkem abstraktní třídy `Controller.class.php` kvůli zajištění jednotného ovládání všech controllerů.

**View** Obsahuje všechny view, které se starají o generování výsledné stránky. Každý view je potomkem třídy `view-template.class.php`, která mimo jiné zajišťuje jednotnost každé stránky včetně navigace. Dále složka obsahuje soubory java scriptu `.js`, které jsou využity pro zobrazení některých prvků stránky. **Model** Obsahuje `data.class.php`, která zprostředkovává přístup k datům a zápis do databáze. Dále script pro vytvoření databáze a projektový soubor MySQL Workbench samotné databáze.

**Files** Složka obsahuje všechny `.pdf` soubory nahrané uživateli.

**Log** Obsahuje logovací soubor.

## 3 Použité technologie

### 3.1 MySQL

MySQL je systém řízení báze dat uplatňující relační databázový model. Komunikace s databází probíhá pomocí jazyka SQL. Protože se jedná o volně šiřitelný software je vhodnou volbou pro zajištění databáze pro vlastní webovou aplikaci. Databáze je použita pro uchovávání všech informací o uživateli a příspěvcích.

### 3.2 PHP (Hypertext Preprocessor)

Jedná se o skriptovací programovací jazyk který je určený především pro programování dynamických internetových stránek a webových aplikací. Kód PHP se vykonává na straně serveru a je použitý ve všech controllerech pro zajištění logiky, ve všech view pro generování stránky a pro přístup k databázi v modelu aplikace.

### 3.3 HTML5 (HyperText Markup Language)

Jedná se o verzi značkovacího jazyka pro tvorbu webových stránek. Je použitý ve všech view, kde je generovaný za pomoci PHP a tvoří tak výsledné stránky. Výsledný HTML kód stránky byl validován pomocí **HTML Validator v0.9.8.8**.

### 3.4 JavaScript

Je multiplatformní, objektově orientovaný, událostmi řízený skriptovací jazyk. Obvykle se využívá jako součást HTML jako pomoc pro ovládání různých interaktivních prvků. JavaScript byl použitý pro otevírání navigačního menu, menu pro vpřirázování recenzí, modálních oken a jejich obsahu a ve spojení s výběrem souboru pro upload.

### 3.5 CSS (Cascading Style Sheets)

Jedná se o jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v jazyce HTML. CSS umožňují oddělení vzhledu od vlastního kódu stránky a je tedy možné snadno měnit vzhled bez zásahu do kódu. V projektu je CSS použité především v W3.CSS. Dále jsou využité jako paleta `w3-my-theme.css` k obarvení komponent a k menším úpravám vlastností komponent v `styles.css`.

### 3.6 W3.CSS

Jako alternativa Bootstrap bylo použito W3.CSS. Jedná se o velmi snadno použitelný framework který zajišťuje responsivní design za použití kaskádových stylů CSS. W3.CSS byl použitý u velké většiny jednotlivých komponent ve view.

## 4 Implementace

### 4.1 Controller

`con-pages-list` Obsahuje seznam všech dostupných controlerů.

`controller.class` Abstraktní třída od které dědí všechny controllery. Poskytuje jednotné ovládací rozhraní.

`con-topics.class` Poskytuje správu vlastních a publikovaných příspěvků.

`con-reviews.class` Poskytuje správu vlastních recenzí.

`con-login.class` Poskytuje správu přihlašování a registrace uživatele.

`con-intro.class` Poskytuje úvodní stránku.

`con-admin.class` Poskytuje správu všech příspěvků, uživatelů a recenzí pro administrátory.

## 4.2 Model

`data.class` Poskytuje přístup k databázi.

## 4.3 View

`view-template.class` Základní template všech view. Poskytuje prostředky pro sestavení stránky v jednotném stylu a navigaci.

`view-admin_topics.class` Zobrazení všech příspěvků.

`view-admin-user-edit-form.class` Formulář pro editaci uživatele.

`view-admin-users.class` Zobrazení všech uživatelů.

`view-intro.class` Zobrazení úvodní stránky.

`view-login.class` Zobrazení přihlašovací stránky.

`view-reviews.class` Zobrazení všech recenzí.

`view-reviews-edit-form.class` Formulář pro editaci recenze.

`view-topics_own.class` Zobrazení vlastních příspěvků.

`view-topics-edit-form.class` Formulář pro editaci vlastních příspěvků.

`view-topics_pub.class` Zobrazení publikovaných příspěvků.

`scripts-sidebar.js` Poskytuje otevření a zavření navigačního menu.

`scripts-modal` Otevírání a zavírání modálních oken. V případě modálního okna pro čtení `.pdf` vytváří příslušný object pro zobrazení.

`scripts-choose-file.js` Aktualizuje výpis názvu vybraného souboru ve formuláři.

`scripts-accordion` Otevírá a zavírá roletkové okno pro vnořenou tabulku a zvýrazňuje příslušné tlačítko.

## 5 Závěr

Práce byla vytvořena s předpokladem rozšiřování o další stránky. Toho lze snadno docílit přidáváním `controlerrů` a příslušných `view`. Samotné stránky jsou responsivní vůči velikosti zobrazitelné plochy. Aby nedocházelo k problémovému čtení byla většina výpisů nahrazena tabulkami s posuvníky. Všechny generované stránky byly validovány pro HTML5 tak aby prošli bez varování. Pro server byl využit Wamp Server (Apache/2.4.35 (Win64) PHP/7.2.10) a pro databázi phpMyAdmin. Práce byla ve výsledku publikována na GitHub.