



# SEMESTRÁLNÍ PRÁCE

*WEBOVÉ APLIKACE*

*STANDARDNÍ ZADÁNÍ*

JAN ŠEDIVÝ

A14B0448P

JSEDIVY@STUDENTS.ZCU.CZ

11.12.2015

# OBSAH

Použité technologie .....	3
Adresářová struktura aplikace .....	3
controller .....	3
core.....	3
model.....	3
public .....	3
upload.....	3
view .....	3
Architektura aplikace .....	4
core.....	4
core.php .....	4
controller.php.....	4
controller .....	4
admin.php .....	4
article.php .....	4
error.php .....	4
home.php .....	4
login.php.....	4
register.php .....	4
review.php.....	5
user.php.....	5
model.....	5

## POUŽITÉ TECHNOLOGIE

V semestrální práci jsou použity všechny povinné technologie, HTML5, CSS3, PHP a MYSQL. Webová aplikace je tvořena v systémové architektuře MVC (Model-View-Controller). Dále jsem použil front-end framework Bootstrap verze 3.3.5 a dva Bootstrap pluginy. První plugin je Bootstrap File Input, který je použitý pro formulářové nahrávání PDF souborů. Druhý použitý plugin je Bootstrap Form Validator, použitý při ošetřování vstupů z formuláře při registraci nového uživatele. Oba pluginy využívají JavaScript. Databáze byla vytvořena v prostředí PhpMyAdmin, aplikace k databázi přistupuje pomocí rozhraní PDO, čímž je ošetřena ochrana proti SQL injection.

## ADRESÁŘOVÁ STRUKTURA APLIKACE

### CONTROLLER

Obsahuje ovladače, které získávají databázová data z modelu a zobrazují požadované pohledy.

### CORE

Adresář obsahující jádro aplikace a dále soubor config, ve kterém jsou definované konstanty používané pro ušetření času při psaní kódu.

### MODEL

Adresář se souborem model.php, ve kterém jsou řešeny veškeré dotazy na databázi.

### PUBLIC

V tomto adresáři je hlavní soubor index.php. Dále se zde nacházejí veškeré CSS soubory včetně Bootstrapu a jeho pluginů a také adresář img obsahující veškeré obrázky použité ve webové aplikaci.

### UPLOAD

Obsahuje autory nahrané PDF soubory.

### VIEW

Poslední adresář, který obsahuje podadresář template s hlavní šablonou a ostatní podadresáře s veškerými pohledy.

# ARCHITEKTURA APLIKACE

## CORE

### *CORE.PHP*

Hlavní část celé aplikace. Obsahuje funkci *parseUrl()*, která rozdělí URL adresu do jednotlivých prvků pole. Třída následně analyzuje jednotlivé prvky pole a zjišťuje, jaký ovladač v URL voláme, jakou metodu, a případně i parametry metody. Následně vytvoří instanci daného ovladače.

### *CONTROLLER.PHP*

Rodičovská třída, ze které dědí všechny ostatní ovladače. Vytváří spojení s databází pomocí rozhraní PDO a vytváří novou instanci modelu.

## CONTROLLER

### *ADMIN.PHP*

V funkci *index()* kontroluje, zda je přihlášen uživatel s administrátorským oprávněním a následně zobrazuje pohledy obsahující administraci. Dále obsahuje funkce *add\_reviewer()* pro přidělení recenzenta ke článku, *remove\_reviewer()* pro odstranění recenzenta a *approve\_article()* pro schválení článku.

### *ARTICLE.PHP*

Obsahuje funkce pro zobrazení pohledů *index()*, která vypisuje všechny schválené články, *article\_detail()* pro zobrazení jednoho kompletního článku, *add\_new()* a *edit\_article()* pro nahrávání článků. Dále funkce *newArticle()* pro uložení článku do databáze a nahrání PDF souboru na server, popřípadě funkce *updateArticle()* pro editaci článku. V této třídě je i funkce *download\_file()* pro stažení PDF souboru.

### *ERROR.PHP*

Jednoduchá třída s funkcí pro zobrazení chybového pohledu, kterou voláme, pokud uživatel zadá neplatnou URL adresu.

### *HOME.PHP*

Třída statických stránek, která obsahuje pouze funkce pro volání pohledů.

### *LOGIN.PHP*

Obsahuje třídu s funkcemi pro zobrazení přihlašovacího formuláře, případně chybových hlášek při přihlašování, a funkce pro přihlášení/odhlášení uživatele.

### *REGISTER.PHP*

Třída s funkcemi pro zobrazení pohledů s registračním formulářem a funkce *registerNew()* pro registraci nového uživatele.

#### *REVIEW.PHP*

Obsahuje funkce *index()* pro zobrazení pohledu s listem recenzí přihlášeného recenzenta, případně funkci *review\_article()* pro zobrazení recenze daného článku, kde je využita funkce *addReview()* pro přidání recenze do databáze.

#### *USER.PHP*

Další jednoduchá třída s funkcí pro zobrazení pohledu s detailem uživatele.

#### **MODEL**

Obsahuje jedinou třídu *model* ve které jsou veškeré SQL dotazy na databázi.