



Semestrální práce z KIV/WEB

Standartní zadání

David Košek
A14N0132P
kosek@students.zcu.cz
4. 12. 2015

1. Použité technologie

Frontend

Pro prezenční vrstvu jsem využil standartních technologií, jako jsou HTML, CSS, a JavaScript – Jquery, AJAX. Celou prezentační vrstvu jsem implementoval hlavně pomocí nejvíce populárního frameworku Bootstrap, který nabízí mnoho předpřipravených komponent, css stylů a také JS věcí. AJAX využívám v několika případech jako například při přihlášení a dohlášení uživatele, či registrace. Jquery společně s PHP zajišťují zbytek logiky na prezentační vrstvě. Aplikace obsahuje také galerii, ve které jsem využil známý framework lightbox. Pro zobrazování informačních zpráv (o přihlášení, o vzniklé chybě) využívám sympatický js framework alertify. Pro zobrazování různých ikon jsem použil styl font awesome.

Backend

Celý backend aplikace je implementován pomocí skriptovacího jazyka PHP. Databázový systém jsem si zvolil MySQL z důvodu snadné práce s touto databází. Pro komunikaci s databází využívám dibi. DIBI je knihovna pro komunikaci s databází. Podporuje mnoho databázových systémů i náš zmíněný MySQL. Jedná se o jednoduchou knihovnu pro elegantní zápis příkazů. DIBI lze svobodně používat i v komerčních projektech. Jejím cílem je maximálně ulehčit práci programátorům (snadný přístup k metodám, zjednodušit zápis SQL příkazu...), eliminovat výskyt chyb, přenositelnost mezi databázovými systémy a zachovat maximální jednoduchost. Třída dibi má jednu výhodu, kvůli které ji využívám, a to, že je vždy po ruce. Nemusím získávat instanci připojení a řešit sama SQL Injection.

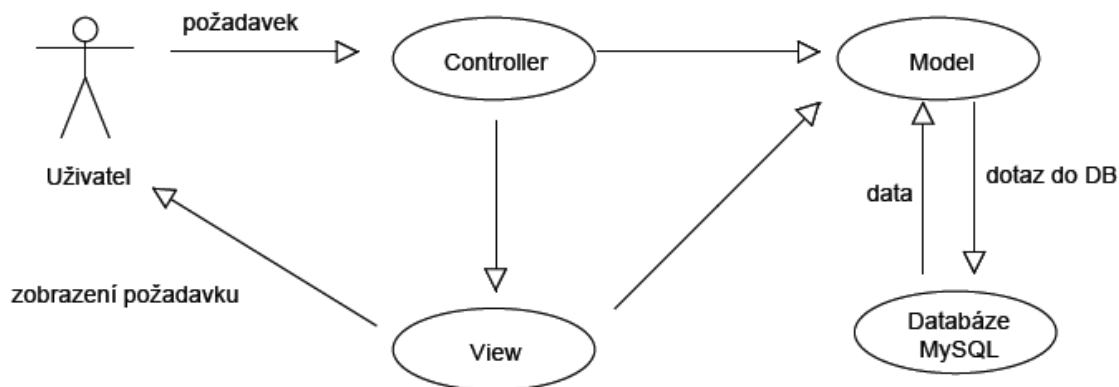
2. Adresářová struktura

Adresářovou strukturu jsem volil dle jednotlivých částí, které k sobě patří.

/class	– obsahuje třídy pro správu jednotlivých stránek
/css	– obsahuje všechny použité css style
/fonts	– obsahuje potřebné fonts kvůli využívání font-awesome
/gallery	– obrázky, zobrazené v galerii, z této složky se automaticky načítají všechny obrázky do galerie
/img	– potřebné obrázky použité v aplikaci
/js	– obsahuje použité skripty
/lib	– obsahuje dibi knihovnu pro komunikaci s databází
/model	– obsahuje modely pro jednotlivé třídy, pouze model přistupuje do databáze
/scripts	– obsahuje jednoduché skripty
/upload	– obsahuje nahrané soubory prostřednictvím aplikace
/view	– obsahuje jednotlivé view dané stránky
_config	– slouží pro připojení k DB
favicon.ico	– ikona viditelná v záložce stránky
index.php	– základní stránka, do které jsou includovány ostatní view
router.php	– zajišťuje nám, které třída bude volána, tudíž i které view

3. Architektura

Pro svoji aplikaci jsem zvolil třívrstvou architekturu MVC. Popisovat tuto architekturu by bylo, předpokládám zbytečné, ale zobrazíme si jí pomocí obrázku (obr. 3.1).



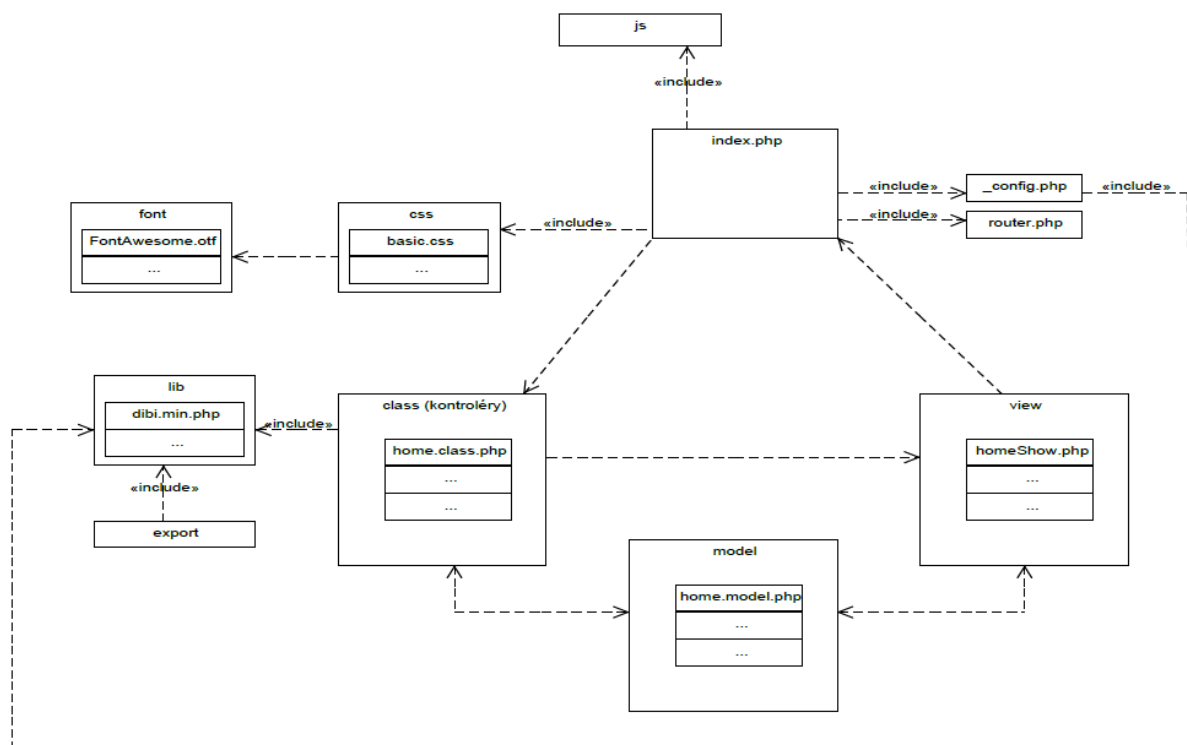
Obrázek 3.1: Schéma MVC architektury

Místo uživatele si lze představit prohlížeč. Z diagramu je jasné, co má která část za úkol. Jak celá aplikace funguje, se pokusím nastínit v implementační části.

4. Implementace

Při implementaci jsem využil objektově orientovaného programování. Všechny stránky vycházejí ze zmíněného *index.php*. Jedná se o konstrukci webu, do které je načítán obsah jednotlivých stránek pomocí PHP funkce *include*. Stránka je rozdělena na hlavičku a tělo. V hlavičce najdeme *meta-tagy*, pro nastavení kódování, klíčových slov, přilinkování css stylů atd. V těle stránky je menu, které je pro všechny možné stránky stejné. Poté se includeje příslušné view. Jednotlivé stránky obsahují vždy nadpis, který naznačuje jejich účel. Poté následuje informační text, případně tlačítka odkazující se na jiné stránky či formuláře.

Pro lepší pochopení a includování jednotlivých věcí jsem vytvořil obrázek (obr. 4.1), ze kterého by mělo být vše jasné.



Celou práci lze najít na githubu https://github.com/kosa13/kiv_web_redakcni_system.git.