

Tvorba redakčního systému pro sportovní portál

Diplomová práce

Vedoucí práce:

Ing. Jiří Lýsek, Ph.D.

Bc. Jaromír Hloch

Brno 2016

Zde bude zadání diplomové práce.

Poděkování

Na tomto místě chci poděkovat vedoucímu práce Ing. Jiřímu Lýskovi, Ph.D. za jeho odborné rady, cenné připomínky a zkušenosti, které mi pomohly při vypracování mé závěrečné práce. Chci poděkovat také rodině za morální podporu a trpělivost.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Tvorba redakčního systému pro sportovní portál** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 20. prosince 2016

Abstract

Hloch, J. Creation of Content Management System for the sports portal. Diploma thesis. Brno: Mendel University, 2016.

In this diploma thesis the scripting language PHP is described. The thesis compares selected PHP frameworks based on software architecture MVC. In practical part is analyzed, designed and implemented the Content Management System for sports portal using technologies and tools described in the theoretical part.

Keywords

Web applications, Content Management System, PHP, MVC, Nette, sports portal.

Abstrakt

Hloch, J. Tvorba redakčního systému pro sportovní portál. Diplomová práce. Brno: Mendelova univerzita v Brně, 2016.

V této diplomové práci je krátce popsán skriptovací jazyk PHP, následně se práce zabývá porovnáním vybraných PHP frameworků, které jsou postaveny na softwarové architektuře MVC. V praktické části je poté s využitím technologií a nástrojů popsáných v teoretické části analyzován, navržen a implementován redakční systém pro sportovní portál.

Klíčová slova

Webové aplikace, redakční systém, PHP, MVC, Nette, sportovní portál.

Obsah

1	Úvod a cíl práce	13
1.1	Úvod.....	13
1.2	Cíl práce.....	13
2	Popis vybraných webových technologií a frameworků	14
2.1	PHP.....	14
2.2	Popis vybraných PHP frameworků postavených na architektuře MVC.....	15
2.2.1	Softwarová architektura MVC.....	16
2.2.2	Laravel.....	17
2.2.3	Symfony.....	17
2.2.4	Nette.....	18
2.2.5	Codeigniter.....	18
2.2.6	Yii.....	19
2.2.7	Srovnání vybraných frameworků.....	20
2.3	MySQL.....	21
2.4	HTML5.....	21
2.5	CSS3.....	21
2.6	JavaScript, jQuery.....	21
3	Metodika	23
4	Návrh redakčního systému	24
4.1	Současný stav.....	24
4.2	Nefunkční požadavky systému.....	24
4.3	Funkční požadavky systému.....	25
4.3.1	Veřejná část.....	25
4.3.2	Administrační část.....	25
4.3.3	Scénáře k vybraným případům užití.....	29
4.4	Použité technologie a frameworky.....	31
4.5	Použité nástroje.....	31

4.5.1	XAMPP	31
4.5.2	Enterprise Architect.....	31
4.5.3	MySQL Workbench	31
4.5.4	NetBeans.....	31
4.5.5	Git, GitHub.....	32
5	Implementace redakčního systému	33
5.1	Struktura systému.....	33
5.1.1	Konfigurační soubory	33
5.1.2	Struktura databáze	34
5.1.3	Struktura modelové vrstvy.....	36
5.1.4	Presentery a šablony.....	41
5.1.5	Komponenty	41
5.1.6	Routování	42
5.1.7	Pomocné třídy	43
5.2	Veřejná část systému	43
5.2.1	Registrace uživatele	45
5.2.2	Zapomenuté heslo.....	47
5.2.3	Výpis článků	49
5.2.4	Detail článku.....	49
5.3	Administrační část systému	51
5.3.1	Správa článků.....	52
5.3.2	Správa fotografií	60
5.3.3	Správa uživatelů	63
6	Uživatelské testování	66
7	Budoucnost redakčního systému	67
8	Závěr	68
9	Literatura	69
A	Ukázky veřejné části	73
B	Ukázky administrační části	76

C Zdrojové kódy

Seznam obrázků

Obr. 1	Popularita PHP frameworků mezi vývojáři	15
Obr. 2	Porovnání jednotlivých PHP frameworků na serveru Github	16
Obr. 3	Princip návrhového vzoru MVC	17
Obr. 4	Use case diagram - 1. část	27
Obr. 5	Use Case diagram - 2. část	28
Obr. 6	Struktura složky /app redakčního systému	33
Obr. 7	ER diagram redakčního systému	36
Obr. 8	Struktura modelové vrstvy	37
Obr. 9	Drátěný model hlavní strany veřejné části portálu	45
Obr. 10	Registrační formulář	46
Obr. 11	Dialog pro zapomenuté heslo	48
Obr. 12	Formulářový dialog pro vytvoření nového hesla	48
Obr. 13	Podoba informativní hlášky v případě úspěšného vytvoření nového hesla.	49
Obr. 14	Seznam komentářů pod článkem z pohledu šéfredaktora	51
Obr. 15	Ukázka administračního rozhraní	52
Obr. 16	Základní obrazovka modulu Správa článků	53
Obr. 17	Pole pro vložení obsahu článku s využitím editoru TinyMCE	54
Obr. 18	Část formuláře pro publikování článku	55
Obr. 19	Automatický našeptávač pro zadávání témat	56
Obr. 20	Příklad výpisu nedostatků při publikování článku	57
Obr. 21	Začátek formuláře pro upravení článku.	57

Obr. 22	Dialog pro nastavení data vydání při úpravě článku.	57
Obr. 23	Modální okno pro výběr fotografie poté, co uživatel zvolil začínající znak tématu.	61
Obr. 24	Výchozí obrazovka modulu Správa uživatelů	64
Obr. 25	Část hlavní strany veřejné části portálu	73
Obr. 26	1. část obrazovky s detailem článku	74
Obr. 27	2. část obrazovky s detailem článku	75
Obr. 28	Ukázka modulu nástěnky před tím, než vloží uživatel první příspěvek.	76
Obr. 29	Ukázka modulu nástěnky, která právě obsahuje dva příspěvky.	76
Obr. 30	Upravit příspěvek na nástěnce lze pomocí modálního okna.	76
Obr. 31	Ukázka modulu Rezervace článků.	77
Obr. 32	Úprava rezervací je řešena pomocí modálního okna.	77
Obr. 33	Uživatel právě vygeneroval statistiku za leden 2016, kterou následně může odeslat na email vybranému příjemci.	77
Obr. 34	Modální okno pro vložení nové fotografie do systému.	78
Obr. 35	Modální okno pro upravení informací o fotografii v systému.	79

Seznam tabulek

Tab. 1	Srovnání frameworků popsaných v předchozích kapitolách	20
Tab. 2	Scénář - Use Case Přidat článek	29
Tab. 3	Scénář - Use Case Připojit fotografii ke článku	29
Tab. 4	Scénář - Use Case Vyhledat fotografii	30
Tab. 5	Scénář - Use Case Přidat komentář	30

1 Úvod a cíl práce

1.1 Úvod

V současné době existuje v oblasti webových aplikací velké množství řešení správy obsahu webových stránek či portálů uživateli, kteří nemají znalosti a zkušenosti ve vývoji webových aplikací a nedokážou tak editovat zdrojový kód. Takovému produktu se také říká redakční systém či software. Spektrum těchto systémů je široké, od těch robustních až po ty menší, od komerčních po volně dostupné.

Při výběru takového systému je tedy potřeba důkladně zvážit typ a rozsah obsahu, kterým bude výsledná webová stránka či portál disponovat. Pokud se např. jedná pouze o blog fyzické osoby, postačí většinou jednoduchý redakční systém nekomerčního typu. Nicméně v případě, že se jedná o komplexnější portál s obsahem, který může spadat do více různých kategorií a který tvoří více lidí nezávisle na sobě navzájem, je zapotřebí poohlédnout se po robustnějším systému.

Třetí možností je navrhnout a realizovat vlastní řešení redakčního systému, které bude přesně reflektovat požadavky takového portálu. S vývojem portálu se tak může upravovat vlastní řešení redakčního systému. Tato možnost je však zřídka realizována, neboť je ekonomicky velice nákladná. Pokud je ovšem tato nevýhoda odstraněna, je to vhodné řešení a investice do budoucna.

Vlastní redakční systém na míru, kterým se zabývá tato diplomová práce, je v tomto případě pro zákazníka ideální, a to z několika důvodů. Zákazník nebude mít na tento redakční systém žádné ekonomické náklady, navíc se také v posledních letech své nejrůznější webové projekty snaží transformovat na vlastní řešení, a tak nabídku vytvoření nového redakčního systému na míru velmi uvítal.

1.2 Cíl práce

Cílem diplomové práce je vytvořit redakční systém pro sportovní portál, který se primárně zabývá sportovními sázkařskými tipy. Před samotnou implementací je žádoucí nejprve provést rešerši dostupných nástrojů, na základě které je potřeba zvolit ty nejvhodnější. Je nutné provést také analýzu a návrh redakčního systému, po jeho úspěšné implementaci pak provést uživatelské testování.

2 Popis vybraných webových technologií a frameworků

V současné době existuje velké množství technologií, které jsou vhodné pro vývoj webových aplikací. V této kapitole jsou popsány nejen backendové, ale také frontendové webové technologie, které jsou využity ve vývoji redakčního systému. Mezi backendové technologie patří programovací jazyky, které jsou zpracovávány na straně serveru (Bittner, 2015). Naopak frontendové technologie jsou podle Bittnera (2015) ty, které nějakým způsobem zasahují do webové stránky na straně webového prohlížeče.

V této kapitole jsou popsány také nejpobulárnější PHP frameworky, které využívají softwarovou architekturu MVC. Z těchto frameworků je poté vybrán ten nejvhodnější a je spolu s popsanými webovými technologiemi využit k implementaci redakčního systému pro sportovní portál.

2.1 PHP

Vývojáři (Achour a kol., 2016) jazyka PHP (Hypertext Preprocessor) jej definují jako skriptovací jazyk, který může být vložen do HTML kódu, a proto se velmi hodí pro vývoj webových aplikací. Je zpracováván na straně serveru, takže nejprve zpracuje požadavek a poté pošle výsledek na stranu klienta - ve většině případů je výsledkem HTML kód, který je poslán prohlížeči (Achour a kol., 2016). Především z tohoto důvodu PHP v současné době využívá jako skriptovací jazyk na straně serveru 82,3% webů na celém internetu (Usage of server-side programming languages for websites, 2016).

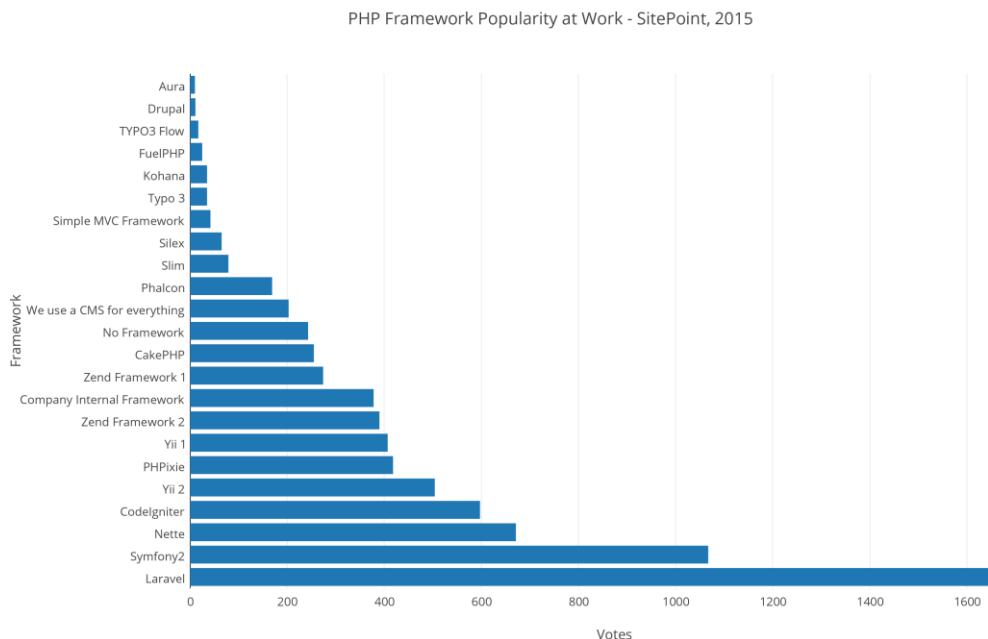
Server W3Schools (2016) ve svém příspěvku zmiňuje hlavní možnosti serverového jazyka PHP. Zde je výčet těch nejdůležitějších:

- dokáže generovat dynamický obsah webových stránek,
- umožňuje pracovat se soubory, které jsou uloženy na serveru,
- umí zpracovávat data z formulářů,
- nabízí univerzální rozhraní pro práci s databázemi,
- umožňuje šifrovat data.

Mezi výhody jazyka PHP řadí server W3Schools (2016) podporu širokého spektra databází, kompatibilitu s nejnámějšími servery (Apache, IIS atd.) nebo také schopnost běžet na nejrůznějších platformách (Windows, Linux, Mac OS X). Naopak Krill (2014) poukazuje na nevýhody jazyka PHP, konkrétně nekonzistentnost názvů funkcí jednotlivých knihoven, některé zastaralé programovací zvyklosti (globální proměnné, magic quotes atd.) a také absenci standardního ladícího nástroje.

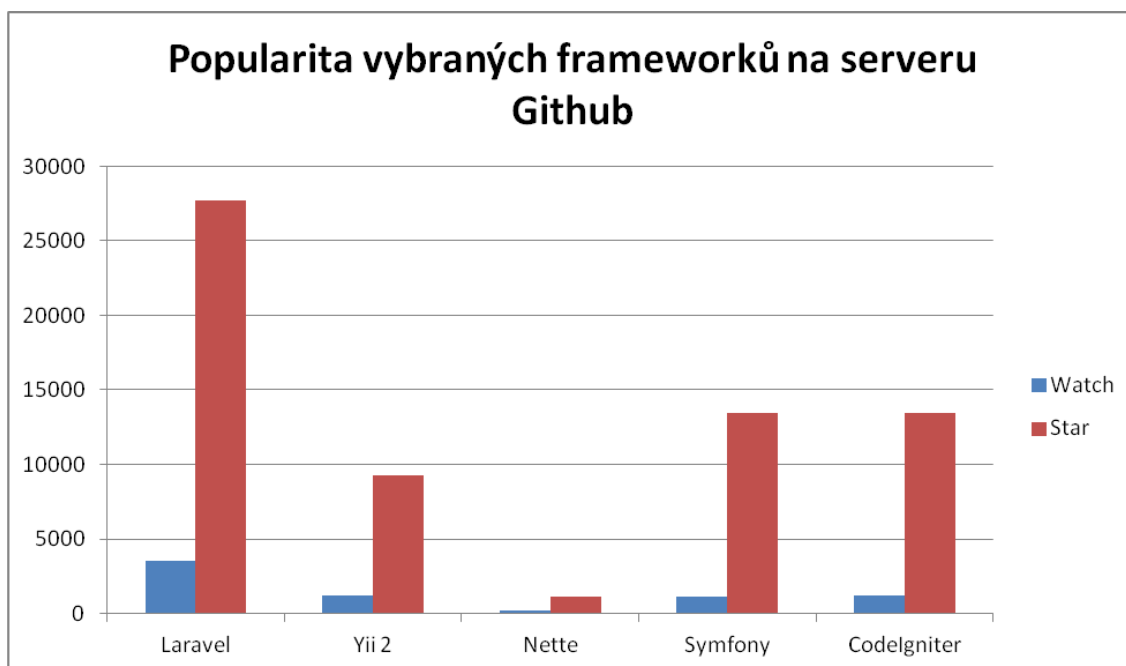
2.2 Popis vybraných PHP frameworků postavených na architektuře MVC

V této části diplomové práce je nejprve popsána softwarová architektura MVC, dále pak 5 nejpoužívanějších PHP frameworků pro rok 2015 podle serveru Sitepoint (The Best PHP Framework for 2015, 2015). Popularitu jednotlivých frameworků mezi vývojáři ukazuje obrázek 1.



Obr. 1 Popularita PHP frameworků mezi vývojáři

Nicméně na serveru Github je pro těchto 5 nejpoužívanějších PHP frameworků situace poněkud odlišná. Obrázek 2 ukazuje graf, na kterém je porovnáno 5 zmíněných frameworků podle počtu sledujících uživatelů a také podle počtu hvězd. Dominance frameworku Laravel zůstala v podstatě stejná, ovšem oproti hodnocení serveru Sitepoint výrazně ztrácí český framework Nette, který má oproti ostatním frameworkům pouhých 167 sledujících uživatelů a 1163 hvězd. To může poukazovat na fakt, že framework Nette ještě není dostatečně rozšířen mezi komunitou, která nemluví česky.



Obr. 2 Porovnání jednotlivých PHP frameworků na serveru Github

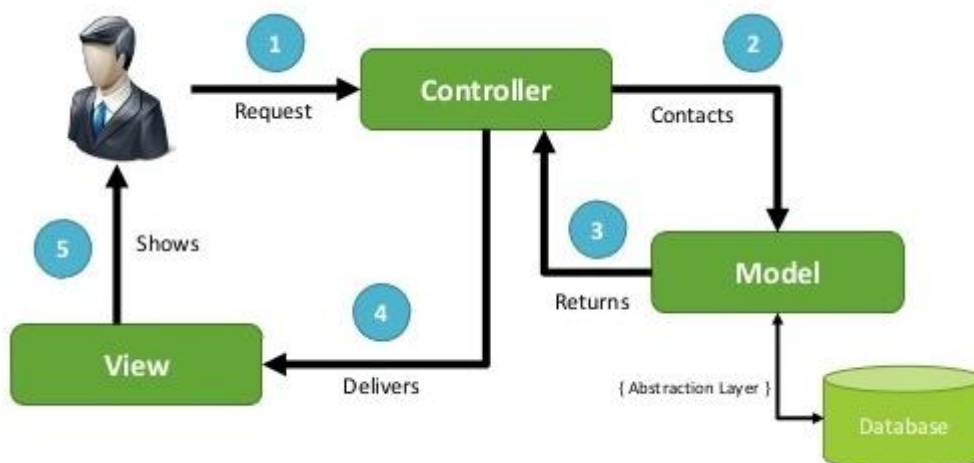
2.2.1 Softwarová architektura MVC

Softwarovou architekturu nebo také návrhový vzor MVC (Model - View - Controller) popsal poprvé norský profesor informatiky Trygve Reenskaug (Model-view-controller, 2016). Vrána (2012) zmiňuje, že MVC rozděluje aplikaci na tři na sobě nezávislé části a tím ulehčuje vývojáři (vývojářům) práci. Princip této architektury je dobře znázorněn na obrázku 3.

Uživatel aplikace učiní určitý požadavek, např. smazání položky. Nejprve je zavolán příslušný Controller, který podle typu požadavku požádá modelovou vrstvu o potřebná data (popř. potřebnou operaci nad daty). Model požadavek vykoná a pošle odpověď zpět Controlleru, který tuto odpověď buď přímo pošle View, nebo ji ještě před odesláním dle potřeby upraví. View pak poskytne uživateli výsledek, tj. ve většině případů vykreslí šablonu.

Výhody vzoru MVC (Vrána, 2012):

- odděluje aplikační logiku (Model) od prezentační vrstvy (View) tím, že je propojuje právě pomocí Controlleru,
- každá ze tří vrstev tohoto vzoru je samostatně upravitelná a testovatelná,
- dobře dělí práci v rámci vývojového týmu (databázový specialista, programátor, kodér).



Obr. 3 Princip návrhového vzoru MVC

2.2.2 Laravel

Jak již bylo zmíněno v kapitole 2.2, nejpoblárnějším PHP frameworkem je v současné době Laravel. Autorem tohoto frameworku je americký programátor Taylor Otwell, nicméně v současné době již na vývoj není zdaleka sám (Laravel, 2016). Laravel je vydáván pod MIT licenci (A PHP Framework For Web Artisans, 2016).

Tento framework může na rozdíl od svých konkurentů nabídnout vývojářům své vlastní vývojové prostředí s názvem Homestead (Laravel Homestead, 2016). Dále lze v elektronické dokumentaci nalézt nástroj pro verzování databáze, které je výhodné především pro práci v týmu (Database: Migrations, 2016).

Pro framework Laravel byl také speciálně vyvinut ORM systém s názvem Eloquent ORM, který je postaven na návrhovém vzoru Active Record (Eloquent: Getting Started, 2016).

Laravel disponuje vlastním šablonovacím systémem s názvem Blade (PHP Frameworks: The Best 10 for Modern Web Development, 2016). Zdroj dále uvádí, že systém Blade nijak výrazně nesnižuje výkon celého frameworku, který je sice podle Garbadeho (2016) poněkud nižší než u ostatních frameworků (Symfony či Yii), nicméně uživatel tuto skutečnost téměř nezaznamená.

Laravel má také širokou komunitu, která vytváří video-návody pro začínající uživatele frameworku (PHP Frameworks: The Best 10 for Modern Web Development, 2016).

2.2.3 Symfony

Tvůrci frameworku na svém oficiálním webu (What is symfony?, 2016) uvádí, že Symfony je framework, který je tvořen sadou znovupoužitelných komponent. Je možné jej získat pod MIT licenci (Symfony, 2016).

Původním zakladatelem tohoto frameworku je Fabien Potencier (Symfony, 2016), avšak na novějších verzích se podílí velký počet vývojářů (Symfony Contributors, 2016).

Kromě standardních nástrojů typického MVC frameworku upozorňuje Koutný (2013) na to, že Symfony podporuje tzv. aspektově orientované programování. Tato technika je podle něj v oblasti webových aplikací užitečným nástrojem, poněvadž řeší problém často se opakujícího problému (sdílení proměnných ve více šablonách) a do jisté míry nahrazuje dědičnost, která je pro tento typ problému méně vhodná.

Koutný (2013) dále vyzdvihuje kvalitní šablonovací systém Twig a také nezávislost frameworku na konkrétním databázovém ORM systému, resp. zabudovaný ORM systém Doctrine2 lze buď vynechat, vyměnit za jiný nebo pár příkazy v konfiguračním souboru nastavit a následně jej používat.

Symfony má však podle Koutného (2013) také své drobné nevýhody, např. méně kvalitní ladící nástroj, než má např. framework Nette (kapitola 2.2.4). Neobsahuje totiž všechny potřebné informace a také je méně přehledná.

2.2.4 Nette

Nette je český framework, jehož zakladatelem je David Grudl, nicméně v současné době se na jeho vývoji podílí několik vývojářů, kteří se sdružují v rámci organizace Nette Foundation (Nette Framework, 2016). Je vydán pod licencí BSD, je tedy volně dostupný.

Nette je komplexní a robustní framework, který je však sestaven z několika samostatných a znovupoužitelných komponent (Nette: Rychlý a pohodlný vývoj webových aplikací v PHP, 2016), podobně jako framework Symfony (kapitola 2.2.3). Zdroj dále uvádí, že tyto komponenty lze používat nezávisle na frameworku.

Nette framework na svých webových stránkách (Nette: Rychlý a pohodlný vývoj webových aplikací v PHP, 2016) uvádí, že disponuje podobně jako např. framework Laravel (kapitola 2.2.2) velice kvalitním šablonovacím systémem zvaným Latte, což potvrzuje ve svém příspěvku také český programátor Jakub Vrána (Unikátní vlastnosti Nette, 2009). Kromě šablonovacího systému nabízí také pokročilou práci s formuláři, které podporují i prvky z HTML verze 5.

Nette výrazně usnadňuje programátorovi implementaci technologie AJAX na straně serveru, obsahuje rovněž nástroj pro vytváření tzv. hezkých URL adres (routování). Velmi známou funkcionalitou v Nette frameworku je také tzv. předávání závislostí.

Jak již bylo zmíněno v kapitole 2.1, jazyk PHP postrádá kvalitnější ladící nástroj. Nette framework však nabízí balíček Tracy, který blíže popisuje autor Nette David Grudl (Laděnka, jak se vám líbí?, 2008). Tento nástroj poskytuje programátorovi komfort při vývoji webových aplikací.

2.2.5 Codeigniter

Framework Codeigniter vyvinula softwarová firma EllisLab v roce 2006, nicméně v současné době tento framework spravuje společnost British Columbia Institute of Technology (A Brief History of CodeIgniter, 2016). Je dostupný pod MIT licencí (CodeIgniter, 2016).

Anand (2014) ve svém článku 7 Best PHP Framework for Enterprise Application popisuje Codeigniter jako framework spíše menšího rozsahu, než jsou první tři frameworky (Laravel - kapitola 2.2.2, Symfony - kapitola 2.2.3 a Nette - kapitola 2.2.4). Codeigniter lze tedy podle Ananda (2014) označit jako micro-framework. Díky tomu je však vhodný pro začátečníky, protože je v porovnání s ostatními frameworky jednoduchý na naučení. Anand (2014) také upozorňuje v souvislosti s menším frameworkem na jeho vysoký výkon.

Anand (2014) doporučuje Codeigniter spíše pro projekty menšího rozsahu, ovšem zároveň dodává, že jádro frameworku je snadno rozšiřitelné o další prvky, se kterými lze poměrně snadno vyvinout i aplikaci většího rozsahu.

2.2.6 Yii

Framework Yii byl veřejnosti poprvé představen v roce 2008 a jeho autorem je čínský programátor Qiang Xue (Garbade, 2016). Je možné jej získat pod BSD licenci (Yii, 2016). V současné době má početnou aktivní komunitu a také dokonce disponuje tištěnou publikací (Yii2 Application Development Cookbook - Third Edition), která obsahuje řadu tipů a návodů, jak efektivně tento framework využívat (Yii2 Application Development Cookbook - Third Edition, 2016).

Garbade (2016) framework Yii charakterizuje jako moderní, bezpečný a velmi výkonný. Velmi výkonný proto, že je postaven na technice lazy loading¹ (Performance of Yii, 2016).

Framework také ve své dokumentaci (Yii PHP Framework Version 2, 2016) uvádí, že nabízí nadstandardní podporu techniky AJAX také díky tomu, že v sobě má explicitně zabudovanou knihovnu jQuery. Tvůrci Yii myslí také na databázovou vrstvu, která nabízí ORM systém založený na návrhovém vzoru Active Record.

Oproti ostatním frameworkům se Yii liší tím, že nabízí programátorům modul pro generování zdrojového kódu, např. generování tříd obsahující ORM entity nebo formuláře (Gii Extension for Yii 2, 2016).

Drobnou nevýhodou frameworku Yii se může zdát absence šablonovacího systému, ovšem to vidí Garbade (2016) jako výhodu, neboť tým vývojářů si může svobodně zvolit právě ten šablonovací systém, který mu nejvíce vyhovuje.

¹ technika, při které se data načítají až v době, kdy jsou skutečně potřebná

2.2.7 Srovnání vybraných frameworků

Tab. 1 Srovnání frameworků popsaných v předchozích kapitolách

Kritérium	Laravel	Symfony	Nette	Codeigniter	Yii
ORM	Eloquent	Doctrine	-	-	ano (Active Record)
Podporované typy databází	PostgreSQL MySQL SQLite MSSQL	PostgreSQL MySQL Oracle Memcache MongoDB MicrosoftBI DynamoDB GraphDB NoSQL CouchDB GemFire Membase	PostgreSQL MySQL SQLite Oracle MSSQL	MySQL Oracle PostgreSQL MSSQL SQLite CUBRID Firebird	SQLite MySQL PostgreSQL MSSQL Oracle
Šablonovací systém	Blade	Twig	Latte	-	-
Routování	ano	ano	ano	ano	ano
SQL injection ochrana	ano	ano	ano	ano	ano
XSS ochrana	ano	ano	ano	ano	ano
CSRF ochrana	ano	ano	ano	ne	ano
Dokumentace	anglická	anglická	česká/anglická	anglická	anglická
Licence	MIT	MIT	BSD	MIT	BSD
Aktivní komunita	ano	ano	ano	ano	ano
CLI	ano	ano	ano	ano	ano
Instalace pomocí composeru	ano	ano	ano	ne	ano
Migrace DB	ano	ano (v rámci Doctrine)	ne	ano	ano

Zdroj: Nette: Rychlý a pohodlný vývoj webových aplikací v PHP (2016); Yii PHP Framework Version 2 (2016); What is symfony? (2016); CodeIgniter User Guide (2016); Laravel (2016)

2.3 MySQL

Databázový systém MySQL je postaven na relačním principu a v současné době je druhým nejpopulárnějším databázovým systémem na světě (DB-Engines Ranking, 2016). Podle organizace WSchools (PHP MySQL Database, 2016) je MySQL spolu s jazykem PHP nejčastější kombinací jazyka na straně serveru a databázového systému, důvodem je především snadná dostupnost a také univerzálnost (obě technologie jsou multiplatformní). Další výhodou MySQL je rychlost a také používání standardního jazyka SQL (PHP MySQL Database, 2016).

MySQL využívá mnoho známých internetových projektů, např. Facebook, YouTube, Joomla, Github, LinkedIn a další (MySQL Customers, 2016).

2.4 HTML5

HTML5 je standardní hypertextový značkovací jazyk, který se využívá ke tvorbě webových stránek. Jednotlivé prvky kódu HTML tedy tvoří stromovou strukturu webové stránky (What is HTML?, 2016). HTML se však nestará o grafický vzhled webových stránek, ale definuje pouze jejich strukturu (HTML Introduction, 2016).

Castro (2012) dále uvádí, že HTML5 přišlo kromě nových formulářových prvků také se zcela novými značkami (tagy), s jejichž pomocí může tvůrce webových stránek ještě lépe definovat jejich strukturu. Tyto značky také pomohou např. zrakově postiženým, kteří jsou díky nim schopni rozumět obsahu webové stránky.

2.5 CSS3

CSS3 je jazyk, pomocí kterého lze formátovat HTML strukturu a určovat tak grafický vzhled webových stránek (Castro, 2012). Organizace W3Schools (CSS Introduction, 2016) zmiňuje, že použitím CSS3 se HTML kód stává přehlednějším a lépe upravitelným. Zároveň také dodává, že použitím CSS3 lze definovat odlišný vzhled webové stránky na odlišných zařízeních - tiskárně a obrazovce.

CSS3 se v neposlední řadě používá také pro realizaci responzivního designu. K tomuto účelu se však neřídka využívá i CSS frameworků, např. Bootstrapu verze 3, který je v současné době nejvyužívanější na světě (Bootstrap 3 Tutorial, 2016).

2.6 JavaScript, jQuery

JavaScript je skriptovací jazyk primárně určený k použití na frontendovém rozhraní (zmíněno v kapitole 2), nicméně v současné době se stále více využívá i na straně serveru (Propst, 2014). Propst (2014) i Zakas (2009) definují tento jazyk jako interpretovaný, dynamický a objektově-orientovaný, který ovšem umožňuje vývojarům přiřazovat objektům vlastnosti a metody až za běhu, tj. dynamicky.

JavaScript je v případě použití na frontendovém rozhraní interpretován prohlížečem, tzn., že jeho výsledné chování může být napříč prohlížeči poměrně odlišné (Zakas, 2009).

I proto je v současné době výhodnější použít některou z JS knihoven, která výše zmíněný problém řeší. Zřejmě nejznámější JS knihovnou je v současnosti jQuery, kterou podle serveru W3Techs (Usage of JavaScript libraries for websites, 2016) používá téměř 72% webů.

Možnosti jQuery jsou poměrně široké. jQuery obsahuje standardní nástroje pro realizaci událostí, manipulaci s HTML dokumentem a komunikaci se serverem (AJAX). Obsahuje také prostředky pro vývoj komplexního uživatelského rozhraní (jQuery UI) a pro mobilní zařízení (jQuery Mobile). Nabízí také testování JS kódu (jQuery Introduction, 2016).

Pokud je ovšem potřeba realizovat komplexnější frontendovou (RIA) aplikaci, knihovna jQuery pro tento typ není vhodná a je lepší zvolit komplexnější JS frameworky, např. AngularJS (AngularJS: HTML enhanced for web apps!, 2016).

3 Metodika

Při tvorbě redakčního systému bylo postupováno skrze následující etapy vývoje webových aplikací, při čemž v průběhu tohoto vývoje se stalo, že tyto jednotlivé fáze se cyklicky opakovaly.

Analýza. Nejdříve bylo nutné analyzovat stav stávajícího redakčního systému, zjistit jeho přednosti a nedostatky. Bylo žádoucí rozhodnout, které části systému jsou pro jeho uživatele klíčové a které naopak do nového systému nezahrnout. Tato analýza probíhala ve spolupráci s nejkritičtějšími uživateli redakčního systému a také s majiteli sportovního portálu.

Návrh. Na základě analýzy byly definovány funkční i nefunkční požadavky na redakční systém. Funkční požadavky byly zpracovány ve formě případů užití i se scénáři. Nefunkční požadavky do jisté míry reflektovaly omezení možností a technologií na straně serveru. Výstupem této etapy tedy bylo nejen definování výše uvedených požadavků, ale také zvolení nástrojů, technologií a frameworků, které byly použity pro implementaci redakčního systému. Návrhem se podrobněji zabývá následující kapitola 4.

Implementace. Na základě definovaných požadavků a zvolených technologií, nástrojů a frameworků byl systém implementován. Při vlastní implementaci byl kladen důraz na přehlednost, znovupoužitelnost a čistotu kódu. Byla také snaha dodržovat obecné principy objektového programování s důrazem na snadnou rozšiřitelnost redakčního systému. Implementace probíhala po malých částech (ve většině případů po jednotlivých modulech), po dokončení každé této části bylo provedeno uživatelské testování autorem systému. Implementaci se věnuje kapitola 5.

Testování. Po dokončení základní verze redakčního systému bylo provedeno uživatelské testování mezi autorem systému a také jeho uživateli (redaktory). Nalezené chyby byly buď opraveny, nebo na ně bylo reagováno jiným, vhodnějším způsobem. Testování je podrobněji rozebráno v kapitole 6.

4 Návrh redakčního systému

4.1 Současný stav

Doposud je sportovní portál postaven na volně dostupném redakčním systému WordPress, je tedy napsán v jazyce PHP a využívá databázový systém MySQL. Celá aplikace běží na webovém serveru Apache pod operačním systémem Linux (LAMP).

Nastavení WordPressu obsahuje spoustu redundantních modulů, naopak některé potřebné chybí. Proto byli uživatelé redakčního systému nuceni využívat některé externí webové služby, např. Facebook či Google Documents. Vznikla proto poptávka po určitém komplexním systému, který by obsahoval všechny potřebné funkce. Vznikla tedy diskuze, jaký postup bude zvolen při tvorbě nového redakčního systému.

Jelikož provozovatel sportovního portálu vlastní také další webové projekty podobného charakteru, které postupem času přecházejí z volně dostupných redakčních systémů na vlastní řešení, i v tomto případě požadoval, aby bylo vyvinuto vlastní řešení, ve kterém bude vývoj pokračovat i po ukončení práce v rámci diplomové práce.

4.2 Nefunkční požadavky systému

- **Multiplatformnost.** Je žádoucí, aby systém byl schopen běžet na různých operačních systémech (Windows, Linux, Mac OS) a také se choval konzistentně správně v různých prohlížečích (Mozilla Firefox, Internet Explorer, Chrome, Opera, Safari a další).
- **Responzivita.** Veřejná část systému musí být plně responzivní, administrační část by měla, i když se nepředpokládá, že by uživatel používal (alespoň některé) funkce v administraci např. na mobilním telefonu.
- **Bezpečnost.** Systém musí být odolný vůči typickým webovým útokům, např. SQL injection, XSS či CSRF. Zároveň musí být osobní data uživatelů chráněna tak, aby i při jejich krádeži třetí stranou byla co nejméně zneužitelná.
- **Autentizace a autorizace uživatelů.** Každému uživateli bude umožněn přístup do určité sekce či provedení vybrané akce na základě jeho role. Administrační část bude přístupná pouze po úspěšném přihlášení.
- **Uživatelská přívětivost.** Uživatel by měl nejen intuitivně ovládat administrační rozhraní, ale také pohodlně využívat veřejnou část systému, např. číst článek (velké písmo, přehledné metainformace o článku), jednoduše se registrovat (nevyžadovat např. nahrání profilové fotografie, která je v tomto systému irelevantní) atd.

4.3 Funkční požadavky systému

Redakční systém je rozdělen na dvě části, a sice na veřejnou a administrační.

4.3.1 Veřejná část

Ve veřejné části bude obsah, který bude dostupný nepřihlášeným uživatelům, až na některé vybrané případy užití, viz následující seznam.

- Zobrazení skupiny článků podle zařazení do kategorií, od určitého autora a také tagu. Na hlavní straně veřejné části je třeba zobrazovat články nezávisle na předchozích kritériích.
- Nepřihlášeným uživatelům nabídnout registraci do systému, případně umožnit resetovat jejich zapomenuté heslo.
- Zobrazení detailu článku včetně komentářů, které pod daný článek spadají.
- Přihlášeným uživatelům umožnit komentovat jednotlivé články. Tento uživatel může své vlastní komentáře také mazat.
- Správa komentářů. Tato funkčnost je umožněna pouze šéfredaktorovi. Ten může označit z nejrůznějších důvodů komentář libovolného uživatele jako nevhodný.

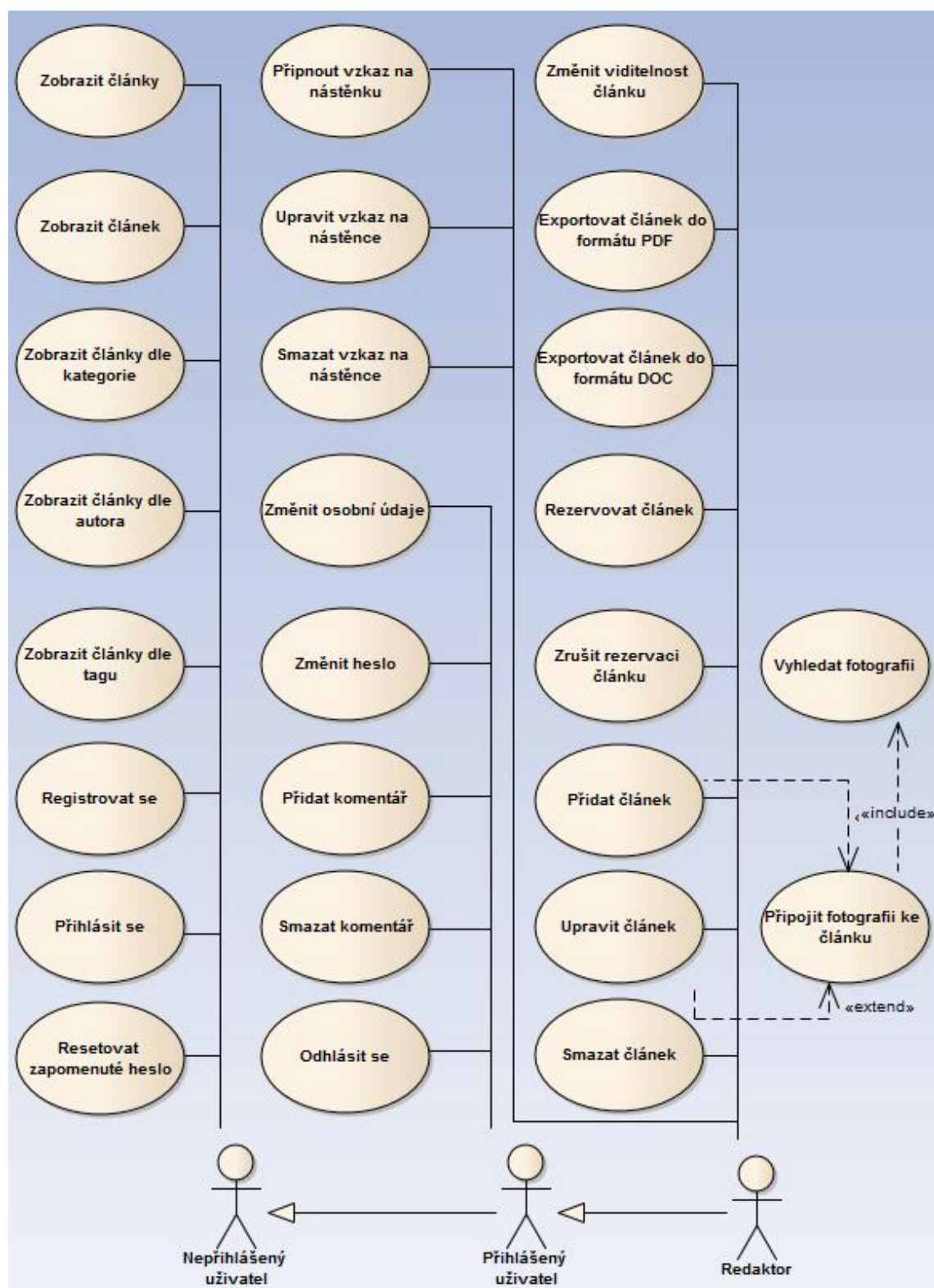
4.3.2 Administrační část

Jak bylo uvedeno v kapitole 4.2, tato část bude přístupná pouze přihlášeným uživatelům. Jednotlivé moduly v této části pak budou přístupné pouze určitým rolím, které k nim budou mít právo přistoupit.

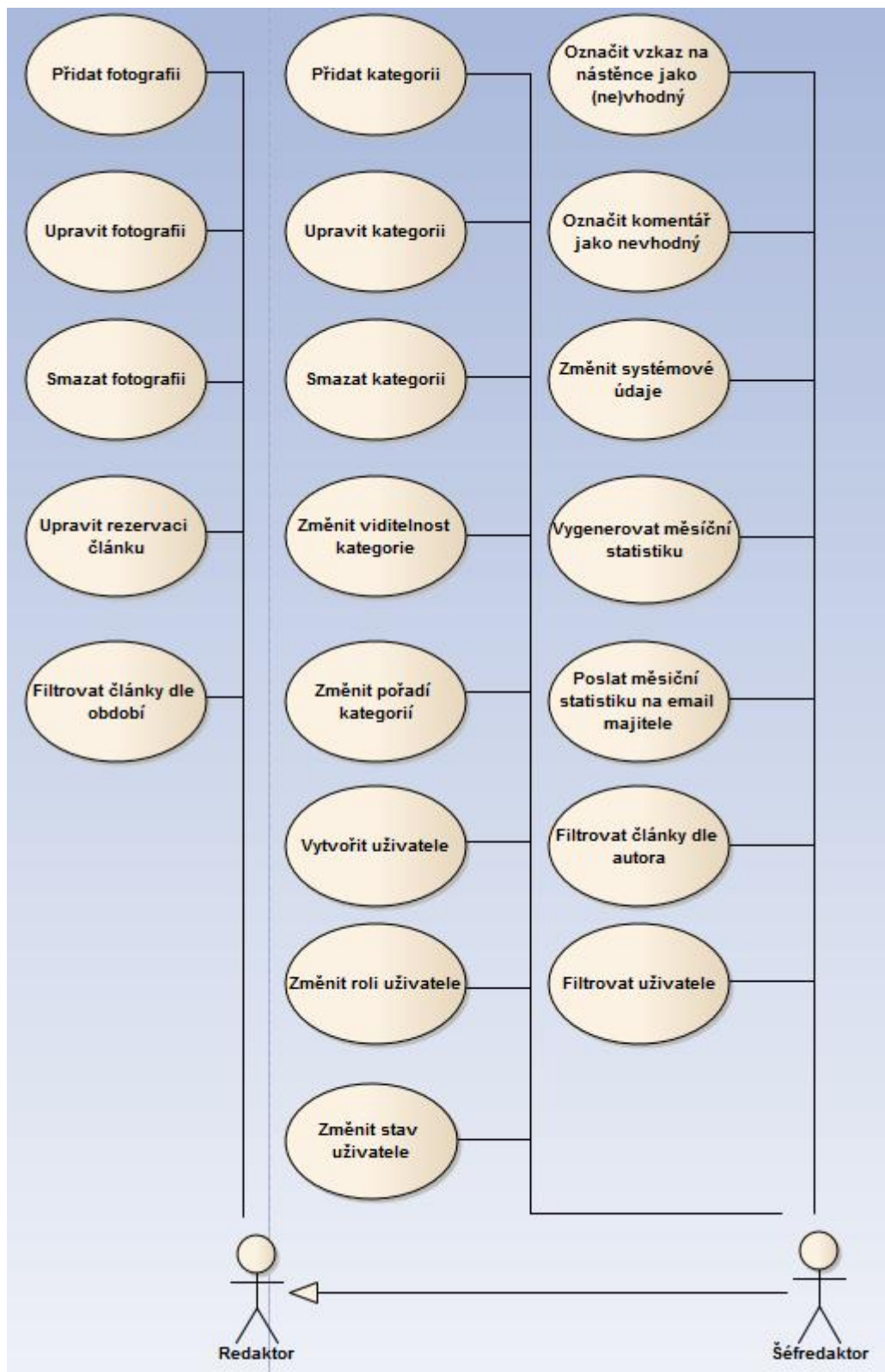
- Správa článků. Klíčový modul celého redakčního systému. Umožňuje uživatelům tvořit si přehled v publikovaných člancích a také s těmito články manipulovat, tj. přidávat, upravovat, mazat, měnit jejich viditelnost ve veřejné části, exportovat do formátů DOC, PDF. Redaktor může takto nakládat pouze s vlastními články, šéfredaktor se všemi. Šéfredaktor navíc může filtrovat články nejen dle měsíců, ale také dle autorů.
- Správa fotografií. Jelikož ke každému článku musí být přiřazena právě jedna fotografie, tento modul úzce souvisí s předchozím, resp. je jeho submodulem. Tento submodul umožňuje redaktorům přidávat do systému fotografie, upravovat, mazat a připojovat je k jednotlivým článkům. Šéfredaktor může manipulovat (úprava, mazání) se všemi fotografiemi, redaktor pouze s těmi, které do systému nahrál.
- Správa uživatelů. Do tohoto modulu má přístup pouze šéfredaktor, který může vytvářet uživatele, měnit jejich roli a také je blokovat či odblokovat. Jelikož bude uživatelů časem nemalý počet, je k dispozici filtr, který nabízí omezení výpisu uživatelů podle rolí, stavu (blokování/neblokování) a aktivitě (aktivní/neaktivní).

- Správa kategorií. Do tohoto modulu má opět přístup pouze šéfredaktor. Zde může vytvářet, upravovat a mazat kategorie. Může také měnit viditelnost a pořadí těchto kategorií.
- Generování statistik. V tomto modulu si může šéfredaktor zobrazit statistiku (počet článků) všech redaktorů za vybraný měsíc. Tato data pak může odeslat majiteli portálu na email (ten je editovatelný).
- Změna systémových údajů. Tento modul je spíše pro budoucí rozšíření. Prozatím je zde pouze možnost změnit email majitele portálu.
- Změna osobních údajů.
- Změna hesla za předpokladu, že uživatel zná heslo aktuální.
- Nástěnka. Slouží ke komunikaci mezi redaktory a šéfredaktorem. Redaktor může vkládat, upravovat a mazat své příspěvky. Šéfredaktor může navíc označit příspěvky libovolného redaktora za nevhodné, či tuto akci vzít později zpět.
- Rezervace článků. Jelikož je portál zaměřen na publikování recenzí na jednotlivá sportovní utkání, měli by mít redaktoři a šéfredaktor přehled v publikování článků. Každý redaktor tedy může vkládat, upravovat a mazat své rezervace na jednotlivé zápasy. Šéfredaktor může totéž provádět se všemi učiněnými rezervacemi.

Celý systém podrobněji popisuje Use Case diagram, který se nachází na obrázcích 4 a 5.



Obr. 4 Use case diagram - 1. část



Obr. 5 Use Case diagram - 2. část

4.3.3 Scénáře k vybraným případům užití

Tab. 2 Scénář - Use Case Přidat článek

Název	Přidat článek
Aktéři	Redaktor, šéfredaktor
Vstupní podmínky	Aktér je přihlášen a má přístup k modulu Správa článků.
Výstupní podmínky	Do systému je přidán nový článek.
Scénář	<ol style="list-style-type: none"> 1. Aktér požádá systém o formulář pro přidání článku aktivací tlačítka Přidat nový článek. 2. Systém tento formulář vygeneruje. 3. Aktér zadá nadpis a obsah článku 4. Extend by Připojit fotografii ke článku. 5. Aktér vybere příslušné kategorie, témata ke článku a také nastaví datum vydání. 6. Aktér odešle formulář aktivací tlačítka Publikovat článek. 7. Systém zkontroluje údaje zadané aktérem a uloží článek. Pokud nejsou údaje validní, tak alternativní scénář Chybné vstupní údaje.
Alternativní scénář: Chybné vstupní údaje	7a1. Systém aktérovi vypíše konkrétní nedostatky a vyzve jej k jejich opravě.

Tab. 3 Scénář - Use Case Připojit fotografii ke článku

Název	Připojit fotografii ke článku
Aktéři	Redaktor, šéfredaktor
Vstupní podmínky	Aktér je přihlášen a nachází se ve formuláři pro přidání článku.
Výstupní podmínky	Ke článku je přiřazena fotografie.
Scénář	<ol style="list-style-type: none"> 1. Aktér pomocí aktivace tlačítka Vyber fotku požádá systém o modální okno Správa fotografií. 2. Systém toto modální okno aktérovi zobrazí. 3. Include Vyhledat fotografii. 4. Aktér z nalezených fotografií aktivuje jednu, jejíž detail se zobrazí v pravém panelu modálního okna. 5. Aktér aktivuje tlačítko Připojit ke článku. 6. Systém připojí příslušnou fotografii ke článku a zavře modální okno.

Tab. 4 Scénář - Use Case Vyhledat fotografii

Název	Vyhledat fotografii
Aktéři	Redaktor, šéfredaktor
Vstupní podmínky	Aktér je přihlášen a nachází se v modálním okně pro Správu fotografií.
Výstupní podmínky	Seznam nalezených fotografií.
Scénář	<ol style="list-style-type: none"> 1. Aktér aktivuje příslušný znak kliknutím na něj. 2. Systém vyhledá všechny fotografie, které obsahují téma, které začíná na požadovaný znak. 3. Systém nalezené fotografie zobrazí uživateli v hlavním levém panelu modálního okna. V případě, že žádnou fotografii nenalezne, tak alternativní scénář Nenalezeny žádné fotografie.
Alternativní scénář: Nenalezeny žádné fotografie	3a1. Systém oznámí aktérovi, že nebyly nalezeny žádné fotografie a vyzve jej, aby zadal jiný znak.

Tab. 5 Scénář - Use Case Přidat komentář

Název	Přidat komentář
Aktéři	Přihlášený uživatel, redaktor, šéfredaktor
Vstupní podmínky	Aktér musí být přihlášen a musí se nacházet na příslušném článku
Výstupní podmínky	Ke článku je přidán nový komentář
Scénář	<ol style="list-style-type: none"> 1. Aktér vyplní text komentáře. Pokud aktér předtím přidal dva a více komentářů během jedné minuty, tak alternativní scénář Vyplnění antisпамu. 2. Aktér odešle komentář pomocí tlačítka Přidat komentář. 3. Systém zkontroluje správnost zadaných údajů a přidá komentář ke článku. Pokud nejsou údaje správně zadány, tak alternativní scénář Chybné údaje.
Alternativní scénář: Vyplnění antisпамu	1a1. Aktér je vyzván k tomu, aby před odesláním komentáře vyplnil antisпамovou otázku.
Alternativní scénář: Chybné údaje	3a1. Systém upozorní aktéra na chybně zadané údaje a vyzve ho, aby je opravil.

4.4 Použité technologie a frameworky

Na základě funkčních, nefunkčních požadavků a dostupných technologií na produkčním serveru popsaných v kapitole 3.1 byly vybrány následující technologie k realizaci redakčního systému. Většina z nich už byla popsána v této práci dříve, proto je v jejich případě uveden pouze název a odkaz na příslušnou kapitolu, ve které je popsán.

- Skriptovací jazyk PHP, popsaný v kapitole 2.1,
- PHP MVC framework Nette, popsaný v kapitole 2.2.4,
- databázový systém MySQL, popsaný v kapitole 2.3,
- značkový jazyk HTML5, popsaný v kapitole 2.4,
- kaskádové styly CSS3, popsané v kapitole 2.5,
- CSS3 & JS framework Bootstrap, zmíněný rovněž v kapitole 2.5,
- JavaScript spolu s knihovnou jQuery, oba popsány v kapitole 2.6.

4.5 Použité nástroje

Zde zmíněné nástroje slouží pouze k usnadnění vývoje aplikace, nejsou tedy nutnou podmínkou k realizaci výsledného redakčního systému.

4.5.1 XAMPP

Je to virtuální webový server, který mj. obsahuje server Apache a také interpret pro skriptovací jazyk PHP a databázový systém MySQL. Umožňuje tedy běh webové aplikace v lokálním prostředí počítače, aniž by bylo potřeba přesouvat vyvíjenou aplikaci na vzdálený server. Je určen zejména pro vývojové a testovací potřeby.

4.5.2 Enterprise Architect

Je to nástroj (desktopová aplikace), který pomáhá při návrhu nejen webových aplikací. Umožňuje definovat diagramy dle UML jazyka, jako např. ER diagram, Use Case diagram, diagram tříd apod.

4.5.3 MySQL Workbench

Je to nástroj pro návrh a správu MySQL databází. MySQL Workbench umožňuje mj. definovat strukturu databáze z vytvořeného ER diagramu, či naopak z existující struktury databáze vygenerovat ER diagram.

4.5.4 NetBeans

NetBeans je volně dostupné vývojové prostředí, které vývojáři usnadňuje práci. Obsahuje mj. také podporu pro PHP, HTML a CSS, takže má vývojář svůj zdrojový

kód barevně vyznačený a tudíž i přehledný. Pro NetBeans existuje také volně dostupný doplněk pro framework Nette spolu se šablonovacím systémem Latte.

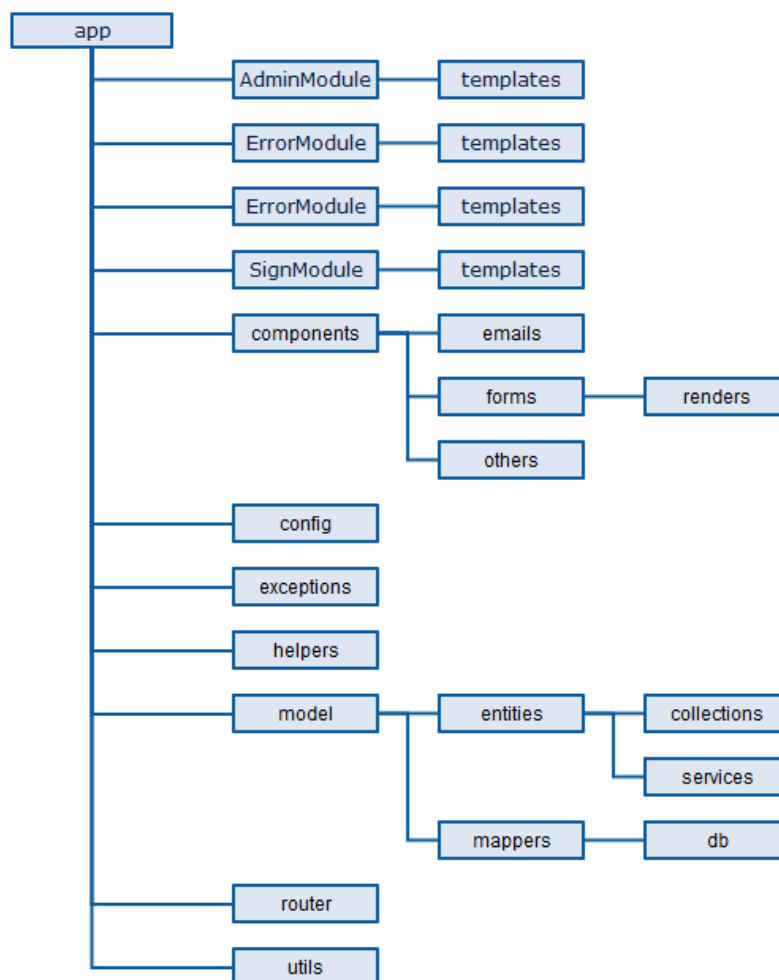
4.5.5 Git, GitHub

Git je distribuovaný verzovací systém, který nabízí archivaci historie vývoje softwarových aplikací. Umožňuje tedy např. v případě potřeby vrátit vyvíjenou aplikaci do dřívější verze. GitHub je webová služba, která uživatelům nabízí publikovat obsah verzované aplikace na serveru github.com.

5 Implementace redakčního systému

5.1 Struktura systému

Jelikož lze tento redakční systém označit za středně velkou aplikaci, je žádoucí zdrojové kódy rozdělit do logických částí. Framework Nette (popsaný v kapitole 2.2.4) proto rozděluje základní strukturu každého projektu do několika adresářů. Redakční systém však potřebuje oproti tomuto standardu několik rozšíření, a tak vznikla konečná adresářová struktura, která je uvedena na obrázku 6.



Obr. 6 Struktura složky /app redakčního systému

5.1.1 Konfigurační soubory

Jak je patrné z obrázku 6, konfigurační soubory jsou umístěny ve složce /`config`. První z nich, soubor `config.local.neon`, obsahuje v případě tohoto redakčního systému pouze připojení k databázi, jehož podoba je následující:

```
database:
  dsn: 'mysql:host=127.0.0.1;dbname=*****'
  user: '*****'
  password: '*****'
  debugger: true
  explain: true
  options:
    lazy: yes
```

Druhý konfigurační soubor `config.neon` obsahuje různá nastavení, a sice časového pásma, způsobu mapování presenterů, expirace sessions, routeru, autorizátoru a také seznamu servisních tříd.

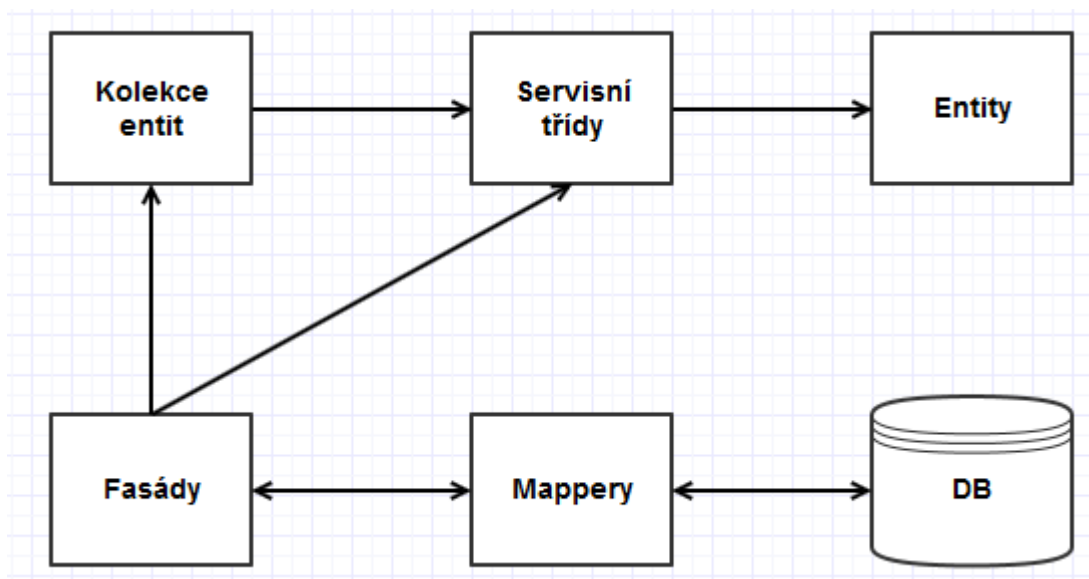
5.1.2 Struktura databáze

Jak již bylo zmíněno v kapitole 2.3, použitá databáze MySQL je založena na relacích mezi jednotlivými tabulkami. Kompletní ER diagram je umístěn na obrázku 7. V ER diagramu jsou uvedeny tabulky s následujícími vzájemnými vztahy.

1. Každý uživatel musí být právě v jedné roli, každá role se může vyskytovat u 0 až N uživatelů.
2. Každý uživatel (šéfredaktor) může vygenerovat 0 až N statistik, každá statistika musí být vygenerována právě jedním uživatelem (šéfredaktorem).
3. Každý uživatel může přidat na nástěnku 0 až N příspěvků, každý příspěvek musí být přidán právě jedním uživatelem.
4. Každý uživatel může přidat do systému 0 až N fotografií, každá fotografie musí být do systému nahrána právě jedním uživatelem.
5. Každý uživatel (redaktor, šéfredaktor) může přidat do systému 0 až N článků, každý článek musí být přidán právě jedním uživatelem (redaktor, šéfredaktor).
6. Každý uživatel může ke článku přidat 0 až N komentářů, každý komentář musí být ke článku přidán právě jedním uživatelem.
7. Každý uživatel může do systému přidat 0 až N rezervací, každá rezervace musí být do systému přidána právě jedním uživatelem.
8. Každá rezervace musí obsahovat právě jeden domácí a jeden hostující tým, každý tým může být v 0 až N domácích týmech a 0 až N hostujících týmech rezervace.
9. Každá role může mít právě jednoho rodiče (roli), každá role může být jako rodič v 0 až N rolích.
10. Každá role může disponovat 0 až N právy, každé právo je uděleno právě jedné roli.
11. Každý zdroj může být předmětem 0 až N práv, každé právo obsahuje právě jeden zdroj.

12. Každý článek může obsahovat 0 až N komentářů, každý komentář musí patřit právě k jednomu článku.
13. Každý článek musí patřit do 1 až N kategorií, každá kategorie může obsahovat 0 až N článků.
14. Každý článek musí obsahovat 1 až N tagů, každý tag může patřit k 0 až N článkům.
15. Každý článek musí obsahovat právě jednu fotografii, každá fotografie může patřit k 0 až N článkům.
16. Každá fotografie musí obsahovat 1 až N tagů, každý tag může patřit k 0 až N fotografiím.

avšak sama je navržena objektově. Struktura modelové vrstvy a vztahy mezi jednotlivými částmi tohoto modelu jsou uvedeny na obrázku 8.



Obr. 8 Struktura modelové vrstvy

Entity

Entitní třídy v modelové vrstvě představují téměř ve všech případech objektovou reprezentaci jednotlivých databázových tabulek. Atributy jednotlivých entit jsou pak zpravidla sloupce v daných databázových tabulkách. Každá entita (třída) obsahuje kromě tzv. getterů a setterů i metody potřebné pro změnu stavu entit (manipulace s atributy). Na ukázkou je uveden následující zdrojový kód, který představuje databázovou entitu `tag`.

```
class Tag extends Entity {

    const MAX_NAME_LENGTH = 50;
    const MAX_URL_LENGTH = 255;

    protected $id;
    protected $name;
    protected $url;

    public function setId($id) {
        $this->setNumber("id", $id);
    }

    public function setName($name) {
        $this->setString("name", $name, 1, self::MAX_NAME_LENGTH);
    }

    public function setUrl() {
        $url = Strings::webalize($this->name);
```

```
        $this->setString("url", $url, 1, self::MAX_URL_LENGTH);
    }

    public function getId() {
        return $this->id;
    }

    public function getName() {
        return $this->name;
    }

    public function getUrl() {
        return $this->url;
    }
}
```

Entita `tag` obsahuje tři atributy - `id`, `name` a `url`. Aby bylo možné předejít potenciálním neshodám s databázovou tabulkou `tag`, obsahuje tato třída také dvě konstanty, které určují maximální délku řetězce atributu `name`, resp. `url`. Hodnoty těchto konstant jsou totiž také maximální povolené hodnoty délky řetězců, které lze vložit do sloupce `name`, resp. `url` v databázové tabulce `tag`. Zbývá ještě doplnit, že třída disponuje také již zmíněnými gettery i settery pro všechny atributy. Jelikož každá entita dědí od abstraktní třídy `Entity`, využívají v nastavovacích metodách metody třídy `Entity` pro kontrolu datových typů (celé číslo, textový řetězec, datum, pravdivostní hodnota `true/false`). Následující zdrojový kód obsahuje jednu z metod abstraktní třídy `Entity` pro nastavení textového řetězce.

```
protected function setString($property, $value, $minLength, $maxLength
= null) {
    if(isset($maxLength)) {
        if(!StringUtil::isLengthBetween($value, $minLength,
            $maxLength)) {
            throw new InvalidArgumentException;
        }
    }
    else {
        if(StringUtil::isUnderMinLength($value, $minLength)) {
            throw new InvalidArgumentException;
        }
    }
    $this->$property = $value;
}
```

Servisní třídy

Servisní třídy poskytují podporu pro entitní třídy, tj. mají za úkol vytvářet jednotlivé entity k nejrůznějším potřebám, např. při vytváření, úpravě, načítání entity z databáze a jiných příležitostech. Každá entita tedy má svoji servisní třídu, tato třída se vždy jmenuje `NázevEntityService`. Všechny metody těchto tříd jsou statické, tzn., že tyto třídy fungují do jisté míry jako návrhový vzor `Factory`. Jako příklad lze uvést metodu pro načtení uživatele z databáze (třída `UserService`).

```

public static function loadFromDb(DbEntry $user) {
    $instance = new User();
    $instance->setId($user->id);
    $instance->setEmail($user->email);
    $instance->setFirstName($user->first_name);
    $instance->setLastName($user->last_name);
    if($user->registered_at) {
        $instance->setRegisteredAt($user->registered_at);
    }
    $instance->setActive($user->active);
    $instance->setBlocked($user->blocked);
    $instance->setHash($user->hash);
    if($user->token) {
        $instance->setToken($user->token);
    }
    if($user->validity) {
        $instance->setValidity($user->validity);
    }
    $instance->setRole($user->role);
    $instance->setUrl($user->url);
    return $instance;
}

```

Kolekce Entit

Pokud je v systému potřeba pracovat s více než jednou entitou, jsou tyto entity uloženy v tzv. kolekci entit. Tyto třídy se pak jmenují `NázevEntityCollection`. Jelikož chování těchto tříd je napříč všemi entitami stejné (liší se pouze v ukládaném typu entity), mají společnou funkcionalitu, která je umístěna v abstraktní třídě `Collection`. Konkrétní potomci poté už nemají žádné atributy ani metody, stačí tyto třídy pouze definovat. Klíčová je tedy implementace metod v již zmíněné abstraktní třídě `Collection`, která disponuje dvěma hlavními veřejnými metodami, které jsou uvedeny ve zdrojovém kódu níže.

```

public static function loadFromDb(DbEntries $entries) {
    $entity = self::ENTITIES_NAMESPACE . self::getCollectionType() .
        "Service";
    $collectionType = get_called_class();
    $collection = new $collectionType;
    foreach($entries as $entry) {
        $collection->addItem($entity::loadFromDb($entry));
    }
    return $collection;
}

public function addItem($item) {
    $collectionType = self::getCollectionType();
    $itemType = self::getItemType($item);
    if($itemType !== $collectionType) {
        throw new InvalidArgumentException("Item must be instance of " .
            $collectionType);
    }
    $this->items[] = $item;
}

```

První z uvedených metod zajistí načtení více databázových záznamů jako entity do příslušné kolekce (třídy). Tu pozná podle toho, který potomek ji volá. Druhá metoda přidá danou entitu do příslušné kolekce. Kontroluje přitom, zda typ entity odpovídá typu dané kolekce.

Mappery

Tento typ tříd slouží k ukládání a načítání dat z úložiště. Jelikož systém ukládá data pouze do databáze, obsahuje pouze databázové mappery. Každý databázový mapper musí implementovat rozhraní daného typu obecného mapperu. Ukázka takového rozhraní je v následujícím zdrojovém kódu, konkrétně se jedná o rozhraní `ICommentMapper`.

```
interface ICommentMapper {  
  
    public function insert(Comment $comment, $articleId);  
  
    public function update(Comment $comment);  
  
    public function delete($id);  
  
    public function findByArticleId($id);  
  
    public function findOneById($id);  
  
    public function getCommentCountForLastMinute(User $user);  
  
}
```

Z výše uvedeného textu vyplývá, že toto rozhraní musí implementovat třída `CommentDbMapper`. Každý takový databázový mapper navíc dědí databázové spojení z abstraktní třídy `AbstractDbMapper`. Databázové mappery pracují ve svých metodách s vrstvou `Nette\Database\Table`, která umožňuje odstínění od klasických SQL dotazů. Tyto dotazy jsou tedy nezávislé na použitém typu databáze. Příklad použití této vrstvy je v následující metodě pro vložení komentáře ke článku do databáze, která je součástí již zmiňované třídy `CommentDbMapper`.

```
public function insert(Comment $comment, $articleId) {  
    return $this->db->table(self::TABLE_NAME)->insert(array(  
        self::COLUMN_CONTENT => $comment->content,  
        self::COLUMN_PUBLISHED_AT => $comment->publishedAt,  
        self::COLUMN_IMPROPER => $comment->improper,  
        self::COLUMN_USER => $comment->user->id,  
        self::COLUMN_ARTICLE => $articleId  
    ));  
}
```

Fasády

Fasádní třídy jsou ve struktuře modelové vrstvy na nejvyšší úrovni, jejich úkolem je tedy komunikovat s ostatními vrstvami systému. Jsou tedy volány většinou

v presenterech či komponentách. Každá fasáda má ve své implementaci vždy instanci mapperu, který využívá (konkrétně vždy databázový). Výměna typu mapperu ve fasádě je tedy velice jednoduchá, stačí změnit vyžadovaný typ instance mapperu v konstruktoru dané fasády. Jako příklad lze uvést část implementace třídy `ArticleFacade`.

```
private $mapper;

public function __construct(ArticleDbMapper $mapper) {
    $this->mapper = $mapper;
}

public function save(Article $article) {
    if(!$article->id) {
        return $this->mapper->insert($article);
    }
    else {
        return $this->mapper->update($article);
    }
}
```

Výše uvedený kód ukazuje nejprve konstruktor třídy `ArticleFacade`, který si vyžádá instanci třídy `ArticleDbMapper` a uloží ji do své privátní proměnné. Vybraná metoda pro uložení článku pak podle toho, zda má entita `Article` nastavený atribut `id`, volá buď metodu mapperu pro uložení či upravení záznamu v databázi.

5.1.4 Presentery a šablony

Presentery mají za úkol zpracovat požadavek uživatele a podle jednotlivých případů užití mu poslat odpověď (kapitola 2.2.1). Jak bylo uvedeno na obrázku 6 v kapitole 5.1, vrstva presenterů a k nim souvisejících šablon je logicky členěna do tzv. modulů. Veřejná (`FrontModule`) i administrační (`AdminModule`) část využívá dva vedlejší moduly, a sice `SignModule`, ve kterém je umístěn jediný presenter s metodou pro odhlášení uživatele, a `ErrorModule`, který zajišťuje zpracování a logování klientských a serverových chyb.

Šablony jsou zpravidla úzce spojeny s daným presenterem, proto jsou umístěny v podadresáři `templates/název presenteru`. Každý z presenterů může mít několik šablon, nebo také nemusí mít žádnou. Adresář `templates` také obsahuje šablonu s názvem `@layout.latte`, která je společná pro všechny ostatní šablony jednotlivých presenterů. Šablona může fungovat též jako komponenta, tudíž může být využita ve více dalších šablonách.

5.1.5 Komponenty

Komponentové třídy jsou umístěny ve složce `components` a jsou logicky rozděleny do 3 podadresářů - komponenty pro posílání emailů, formulářové a také ostatní

komponenty. V adresáři pro posílání emailů je umístěna jediná třída s názvem `MailSender`, která obsahuje jedinou metodu `send($from, $to, $subject, $template)`. Vedle této třídy se nachází v adresáři také několik latte šablon, které představují těla jednotlivých emailových zpráv.

Formulářové komponenty naopak představují konkrétní formuláře v systému. Jednou vytvořený formulář tak může být využit na více místech. Takováto třída zpravidla obsahuje kromě konstruktoru, který si vynutí všechny potřebné závislosti, také metodu `create()`. Ta se postará o vytvoření instance formuláře a zároveň jej vrátí jako návratovou hodnotu. Nepovinně může komponenta obsahovat také metodu, která se zavolá po úspěšném odeslání formuláře a která zpracuje odeslaná data.

V adresáři `others` se nachází dvě třídy - `Paginator` a `Antispam`. `Paginator` slouží pro stránkování libovolných položek, `Antispam` slouží pro generování dat pro antispamovou ochranu.

5.1.6 Routování

Routování slouží pro obousměrný překlad URL adres a akcí presenterů. O routování se stará třída `RouterFactory`, která disponuje statickou metodou `createRouter(ArticleFacade $articleFacade, CategoryFacade $categoryFacade, TagFacade $tagFacade, UserFacade $authorFacade)`. Ta vrátí pole všech definovaných routovacích pravidel. Routovací pravidla lze definovat více způsoby. Pokud je výsledná URL adresa statická (tj. bez parametru), stačí do příslušného routeru přidat jednoduché pravidlo.

```
$router = new RouteList;

$router[] = $signRouter = new RouteList("Sign");
$signRouter[] = new Route("odhlaseni", "Sign:out");
...
return $router;
```

Výše uvedený kus kódu nejprve vytvoří instanci třídy `RouteList`, poté do něj přidá pravidlo, které říká, že při zavolání presenteru s názvem `Sign` a akcí `out` má aplikace přesměrovat na adresu <http://domena.cz/odhlaseni> a naopak. V případě složitějšího routování je potřeba použít funkci. Např. URL adresy ve tvaru <http://domena.cz/clanek/id-nazev-clanku> bylo dosaženo následujícím fragmentem kódu.

```
$frontRouter[] = new Route("clanek/<id>", array(
    "presenter" => "Article",
    "action" => "default",
    "id" => array(
        Route::FILTER_IN => function($url) use ($articleFacade){
            if(is_numeric($url)){
                return $url;
            } else {
```

```
        $article = $articleFacade->findIdByUrl($url);
        if($article){
            return $article->id;
        } else {
            return NULL;
        }
    }
},
Route::FILTER_OUT => function($id) use ($articleFacade){
    if(!is_numeric($id)){
        return $id;
    } else {
        $article = $articleFacade->findUrlById($id);
        if($article){
            return $article->url;
        } else {
            return NULL;
        }
    }
},
),
));
```

5.1.7 Pomocné třídy

Pomocné třídy jsou umístěny v adresáři `utils`. Všechny tyto třídy obsahují výhradně statické metody a atributy. Každá z těchto pomocných tříd obsahuje metody pro práci s různými datovými typy:

- `DateTimeUtils`. Tato třída disponuje metodami pro práci s datem a časem, např. vytvoření datového typu `DateTime` z textového řetězce na základě formátu, vrácení aktuálního data, českých dnů v týdnu atd.
- `StringUtils`. Tato třída vlastní metody pro kontrolu délky, prázdnosti řetězce atd.
- `NumberUtils`. Obsahuje jedinou metodu pro kontrolu kladného celého čísla.

Nutno zmínit, že výše zmíněné třídy, resp. jejich metody jsou do jisté míry pouze mírnou modifikací metod či funkcí frameworku Nette či přímo jazyka PHP tak, aby výsledek reflektoval potřeby redakčního systému.

5.2 Veřejná část systému

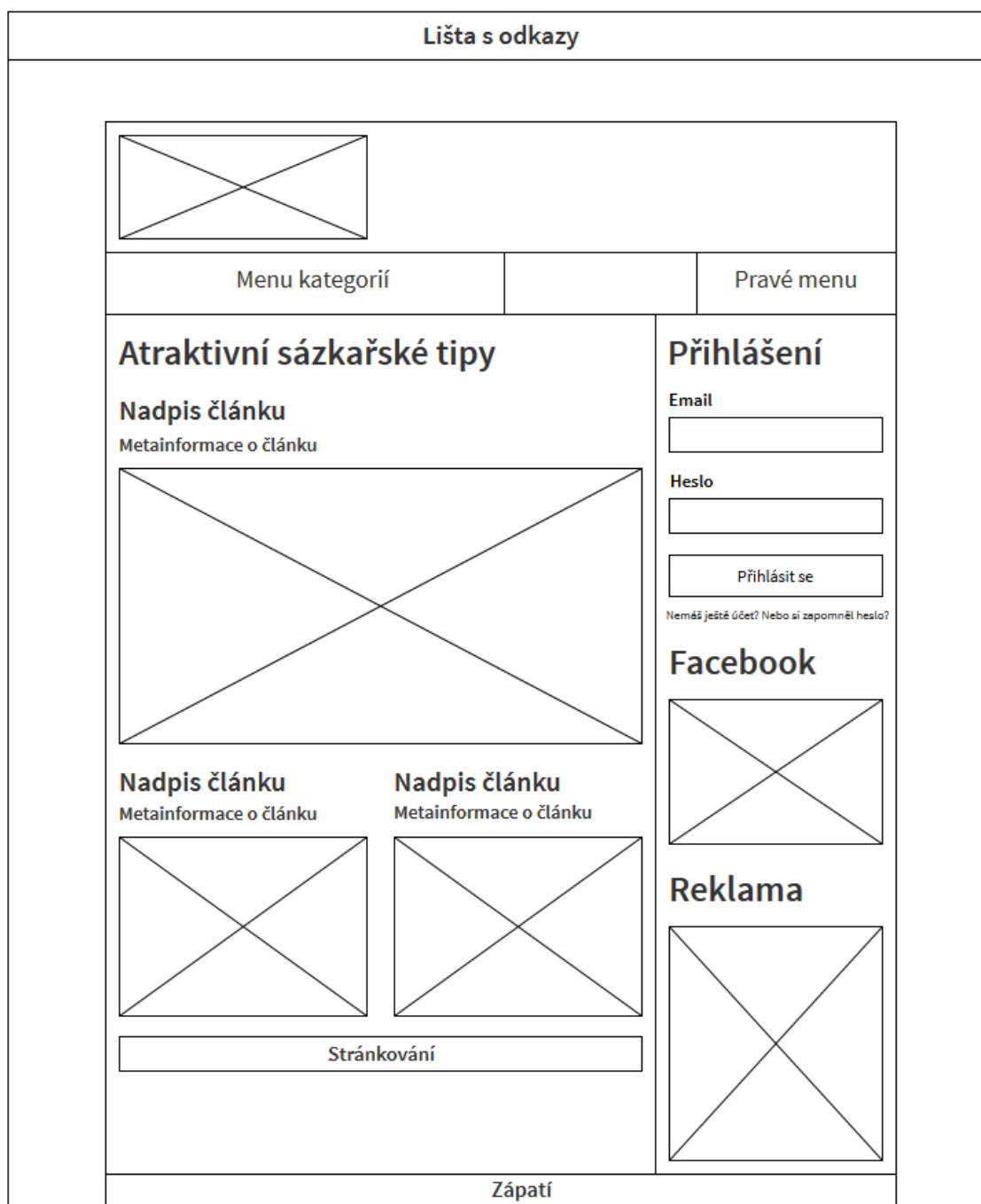
Sportovní portál, který se zabývá sázkařskými tipy pro své uživatele, obsahuje samozřejmě kromě samotného administračního rozhraní také veřejnou část. Ta byla z části představena v kapitole 4.3.1.

Jak bývá u podobných portálů či jednodušších webových stránek zvykem, hlavní strana spolu se všemi dalšími podstranami má určité prvky společné a liší se pouze obsahem samotné stránky. Těmito prvky jsou konkrétně:

- Horní lišta, která obsahuje odkazy na partnerské portály.

- Záhloví, ve kterém se v jeho levé části vyskytuje logo portálu. V pravé části je plánováno umístění reklamy.
- Horizontální hlavní nabídka (menu). V jeho levé části se nachází výpis kategorií definovaných v administrační části. Naopak v pravé části je odkaz na kontakt společnosti, která portál provozuje, a všeobecné podmínky, které stanovuje.
- Pravý panel. V jeho horní části se nachází přihlašovací dialog spolu s odkazem na registraci a vygenerování nového hesla v případě, že uživatel není přihlášen. V opačném případě je místo těchto prvků zobrazeno jméno, příjmení a role přihlášeného uživatele společně s odkazem do administrační části systému a tlačítkem pro odhlášení. Hned pod těmito prvky je zobrazen výstřižek z facebookové stránky Sportwin. Dále pak následují reklamy spolu s bannery spřátelených portálů podobného zaměření.
- Zápatí, které obsahuje autorské podmínky.

Ukázku hlavní strany veřejné části portálu je možné vidět v příloze A na konci této práce. Na obrázku 9 je uveden její drátěný model.



Obr. 9 Drátěný model hlavní strany veřejné části portálu

5.2.1 Registrace uživatele

Jak bylo uvedeno v předchozí kapitole, pokud je uživatel nepřihlášený (návštěvník), nabídne mu portál odkaz umístěný pod přihlašovacím dialogem s názvem

„Nemáš ještě účet“², který směřuje na registrační formulář. Ten je vidět na obrázku 10. Kromě jiného obsahuje celkem 5 povinných polí, tj. email, který bude sloužit nejen ke komunikaci uživatele s portálem, ale také jako přihlašovací údaj, jméno, příjmení a své heslo, které musí uživatel zadat pro kontrolu dvakrát. Heslo musí obsahovat alespoň 6 znaků.

Registrace

Jaké budeš mít výhody, pokud se registruješ?

- můžeš komentovat články

Email

Příjmení

Jméno

Heslo (min. 6 znaků)

Heslo znovu

Registrovat se

Obr. 10 Registrační formulář

Poté, co uživatel splní všechna validační pravidla definovaná v metodě `create()` komponenty `RegistrationForm`, je zavolána metoda, která zajistí zpracování registračního formuláře. Ta nese název `registrationFormSucceeded(Form $form, $values)` a je umístěna v presenteru `RegistrationPresenter`. Tato metoda nejprve požádá třídu `UserService` o vytvoření instance třídy `User` z položek formuláře, poté zavolá prostřednictvím třídy `UserFacade` metodu `save(User $user)`, která nového uživatele uloží, resp. zavolá příslušný databázový mapper, který vloží nový záznam do tabulky `user`.

² Předpokládá se, že cílová skupina sportovního portálu budou mladí muži - převážně sázkaři, tudíž byl zvolen jako komunikační styl portálu s uživatelem tykání.

V případě, že email vkládaného uživatele v databázi není, je zavolána metoda `send($from, $to, $subject, $template)` třídy `MailSender` (popsána v kapitole 5.1.5). Ta odešle aktivační email novému uživateli s odkazem, pomocí kterého je možné vytvořený účet aktivovat. Odkaz je platný 12 hodin od doby odeslání emailu. V případě, že email vkládaného uživatele v databázi už je, uživatelský účet se nevytvoří a uživateli se zobrazí upozorňující hláška. Kompletní zdrojový kód metody `registrationFormSucceeded` je umístěn níže.

```
public function registrationFormSucceeded(Form $form, $values) {
    try {
        $user = UserService::create($values);
        $this->userId = $this->userFacade->save($user);
        $this->mailSender->send("registrace@sportwin.cz",
                               $values->email,
                               "Dokončení registrace",
                               $this->createRegistrationTemplate($user));
        $this->flashMessage("Na Vaši emailovou adresu byly zaslány pokyny
                             k dokončení registrace.", "alert-success");
        $this->redirect("this");
    } catch (DuplicateEmailException $ex) {
        \Tracy\Debugger::log($ex);
        $form->addError("Zadaný email už používá jiný uživatel, zvol si
                        prosím jiný.");
    }
}
```

Pokud uživatel klikne na aktivační email, zavolá se metoda `actionActivateUser()`, která sice nepřijímá žádný parametr, ovšem pracuje s persistentními parametry `token` a `email` uvnitř presenteru `RegistrationPresenter`. Podle těchto parametrů je nalezen v databázi daný uživatel, který je načten do instance třídy `User`. Nad touto instancí třídy `User` je následně zavolána metoda `activate()`, která nastaví příslušné atributy uživatele tak, aby byl aktivní. Poté je instance tohoto uživatele uložena, resp. upravena v databázi. Následně je pomocí již zmíněné třídy `MailSender` odeslán email uživateli, že byl proces registrace úspěšně dokončen.

5.2.2 Zapomenuté heslo

Požádat o změnu stávajícího hesla bez jeho znalosti je možné přes případ užití s názvem „Zapomenuté heslo“, který je přístupný přes odkaz v pravém panelu portálu za předpokladu, že není uživatel přihlášen. Dialog s požadavkem na změnu hesla je zobrazen na obrázku 11.

Zapomenuté heslo

Nech si na svůj registrační email poslat instrukce, pomocí kterých si vytvoříš nové heslo bez znalosti toho stávajícího!

Email

Zažádat o změnu hesla

Obr. 11 Dialog pro zapomenuté heslo

Po správném vyplnění emailové adresy systém zavolá metodu `forgotPasswordFormSucceeded(Form $form, $values)`, která zpracuje data z formuláře pro zadání emailu. Tato metoda nejprve prostřednictvím třídy `UserFacade` vyhledá podle zadaného emailu uživatele a načte ho do instance třídy `User`. Poté testuje, zda je uživatel aktivní, či nikoliv. Pokud ano, tak zavolá metodu třídy `User` s názvem `updateToken()`, která uživateli nastaví atribut `token` pro vytvoření nového hesla. Následně je zavolána metoda `save(User $user)` třídy `UserFacade`, která uloží změny daného uživatele. Poté je pomocí třídy `Mailer` odeslán uživateli email, ve kterém se nachází odkaz na formulář, pomocí kterého si může uživatel vytvořit nové heslo bez znalosti hesla stávajícího. Tento formulářový dialog je uveden na obrázku 12.

Tvorba nového hesla

Zadej nové heslo, které budeš používat pro přihlašování.

Heslo (min. 6 znaků)

Heslo znovu

Vytvořit heslo


Obr. 12 Formulářový dialog pro vytvoření nového hesla

Pravidla pro vytvoření nového hesla jsou samozřejmě stejná jako v případě zadání hesla v rámci registrace uživatele (předchozí kapitola 5.2.1). Pokud je formulář vyplněn správně (heslo je dvakrát zadáno stejně a obsahuje alespoň 6 znaků), třída `ForgotPasswordPresenter` zavolá metodu `newPassword-`

`FormSucceeded(Form $form, $values)`. Ta v první řadě najde pomocí metody `findOneByValidToken($email, $token)` třídy `UserFacade` uživatele podle persistentních parametrů `token` a `email`. Pokud je uživatel nalezen, je načten do instance třídy `User`. Pokud je tento uživatel aktivní, zavolá instance třídy `User` metodu `changePassword($password)`. Zdrojový kód této metody je uveden níže.

```
public function changePassword($password) {
    $this->hash = Passwords::hash($password);
    $this->token = null;
    $this->validity = null;
}
```

Následuje zavolání metody `save(User $user)` třídy `UserFacade`, která se postará o uložení změn atributů instance třídy `User`. Uživatel je o výsledku této operace informován, čímž proces vytvoření nového hesla končí. Příklad takové informativní hlášky je uveden na obrázku 13.



Tvoje heslo bylo úspěšně změněno. Nyní se můžeš přihlásit. ×

Obr. 13 Podoba informativní hlášky v případě úspěšného vytvoření nového hesla.

5.2.3 Výpis článků

Výpis článků je ve veřejné části portálu realizován na více místech. Na hlavní straně je výpis článků nezávisle na autorovi, kategorii a tématu (tagu). Jediné omezení je ve viditelnosti článků (označené jako neviditelné) a datu vydání (naplánované na budoucí datum). Pokud uživatel přejde na některou z kategorií, omezí se výpis článků pouze na danou kategorii. Stejně je to v případě kliknutí na autora libovolného článku a také tématu (tagu), který je umístěn v detailu článku v sekci Související témata.

Jelikož může být počet článků (a téměř ve všech případech i je) velmi vysoký, bylo k těmto výpisům přidáno stránkování v podobě tlačítka „Další články“³, které zajistí připojení dalších článků ke stávajícím. Toto stránkování je řešeno pomocí technologie AJAX. O stránkovací matematiku se stará třída `Paginator`, která dědí od třídy `\Nette\Utils\Paginator` a která mírně modifikuje její chování.

5.2.4 Detail článku

O zobrazení detailu článku se stará třída `ArticlePresenter`, konkrétně tedy její veřejná metoda `renderDefault($id)`, jejíž zdrojový kód je uveden níže.

³ V tomto kontextu jsou to články, ovšem komponentu stránkování je možné využít u libovolného typu položek.

```
public function renderDefault($id) {
    try {
        $article = $this->articleFacade->findOneById($id);
        $article->incrementDisplayed();
        $this->articleFacade->incrementDisplayed($article);
        $this->template->article = $article;
    } catch (EntityNotFoundException $ex) {
        \Tracy\Debugger::log($ex);
        $this->flashMessage("Požadovaný článek nebyl nalezen.",
            "alert-danger");
        $this->redirect("Homepage:default");
    }
}
```



Metoda `renderDefault` nejprve zavolá metodu `findOneById($id)` třídy `ArticleFacade`, která vrátí instanci třídy `Article`. Nicméně pokud článek není nalezen, vyhodí výjimku `EntityNotFoundException`, která je zpracována níže v kódu. Ještě předtím, než je tato instance předána do šablony `default`, zvýší metoda `incrementDisplayed()` třídy `Article` počet zobrazení daného článku a následně je instance třídy `Article` uložena pomocí stejnojmenné metody třídy `ArticleFacade`.

Na stránce s detailem článku je tedy nejprve jeho název, poté metainformace (autor, datum, zařazení do kategorií a také počet komentářů článku). Poté následuje fotografie, která mj. obsahuje svůj popis a zdroj. Pod fotografií je zobrazen text článku, pod textem jsou umístěna související témata (tagy) jako odkazy (výpis článků dle tagu, předchozí kapitola 5.2.3) a nakonec také seznam komentářů, který je demonstrován na obrázku 14.



Komentáře

 **Hloch Jaromír** před 7 dny 



Druhý komentář

 **Hloch Jaromír** před 9 minutami 

ME 2016 vyhraje zřejmě Francie, Ronaldo totiž není zdravotně úplně v pořádku.

 **Lánský Bořivoj** před 2 minutami 

Tento komentář byl označen šéfredaktorem jako nevhodný.

 **Lánský Bořivoj** před minutou 

Ne, vyhraje Portugalsko v prodloužení!!!

Mám totiž vsazené!

Napiš svůj názor!

Přidat komentář

Obr. 14 Seznam komentářů pod článkem z pohledu šéfredaktora

V případě, že je uživatel přihlášen, se zobrazí formulář pro vložení komentáře. Toto vložení je řešeno pomocí technologie AJAX, tudíž uživatel po vložení komentáře ihned vidí výsledek a nemusí rolovat stránkou znovu na seznam komentářů. Na obrázku 14 je vidět seznam komentářů z pohledu šéfredaktora (v tomto případě se jedná o uživatele jménem Jaromír Hloch). Ten má oproti standardním možnostem (smazat svůj komentář) také právo označit komentář libovolného uživatele jako nevhodný, či tuto akci vzít zpět.

5.3 Administrační část systému

Administrační část je zpřístupněna pouze registrovaným (přihlášeným) uživatelům. Administrace je založena na jednotlivých modulech, ke kterým jsou jasně definována přístupová pravidla (dynamický ACL), tzn., že jednotlivý modul je dané uživatelské roli zpřístupněn buď zcela, nebo jen částečně. Ukázky administračního rozhraní demonstruje jak obrázek 15, tak obrázky v příloze B.

Administrace – SportWin Jaromír Hloch (šéfredaktor) [Odhlásit se](#)

Správa článků

[+ Přidat nový článek](#)

Filtr

Leden 2016 Všechni autoři [Filtrovat](#)

Leden 2016 – všichni autoři

Za toto období bylo na portálu publikováno 7 článků.

Titulek	Publikován	Naposledy upraven	Zobrazen	Autor	Kategorie	Zveřejněn	Akce	Export
Liverpool má nakročeno na pohárové přičky ...	31. 01. 2016 09:45	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Předehrávka 20. kola Eredivisie slibuje ...	25. 01. 2016 11:44	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Juventus by chtěla do čela Serie A. Proti ...	24. 01. 2016 10:26	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Šlágr kola se o víkend chystá na ...	23. 01. 2016 15:44	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Atraktivní středeční čtvrtfinále! Celta ...	19. 01. 2016 09:52	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Swansea chce pryč od sestupu. Musí však ...	17. 01. 2016 09:30	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄
Vložené 21. kolo Premier League ...	10. 01. 2016 08:47	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	↩ ✖ 🔄 📄 📄	📄 📄

Copyright © 2015 - 2016 Redakční systém vytvořil Jaromír Hloch

Obr. 15 Ukázka administračního rozhraní

Cílem grafické podoby administrace byla jednoduchost a přehlednost. V záhlaví se v jeho levé části nachází pouze název části portálu, který zároveň funguje jako odkaz na hlavní stranu administrace. V pravé části je umístěna ikonka s celým uživatelským jménem a také jeho rolí. Nakonec je tam také tlačítko pro odhlášení uživatele, po kterém je uživatel odhlášen a přesměrován na hlavní stránku veřejné části portálu.

V levém panelu stránky se nachází hlavní nabídka, kde každá její položka představuje zpravidla jeden modul administrace. Pokud se uživatel právě nachází v určitém modulu, je položka představující daný modul zvýrazněna (viz obrázek 15). Před názvem každého modulu je navíc umístěna intuitivní ikonka, která symbolizuje funkci jednotlivých modulů.

V dolní části stránky je standardně umístěno zápatí, kde jsou umístěny informace o autorských právech a také kontakt na autora systému. V následujících podkapitolách budou popsány pouze vybrané klíčové moduly, které nejvýrazněji ovlivňují obsah celého portálu.

5.3.1 Správa článků

Tento modul je zřejmě nejdůležitější v celém redakčním systému. Umožňuje totiž uživateli provádět nejrůznější operace s články, které jsou těžištěm sportovního portálu. Přístup k tomuto modulu má uživatel, který je v roli redaktora či šéfredaktora. S tím rozdílem, že redaktor může operovat pouze se svými články, zatímco šéfredaktor může operovat se všemi články v systému. Pokud uživatel vstoupí do modulu „Správa článků“, zobrazí se mu následující dialog, který je uveden na obrázku 16.

Správa článků

[+ Přidat nový článek](#)

Filtr

Leden 2016 Všichni autoři [Filtrovat](#)

Leden 2016 – všichni autoři

Za toto období bylo na portálu publikováno 7 článků.

Titulek	Publikován	Naposledy upraven	Zobrazen	Autor	Kategorie	Zveřejněn	Akce	Export
Liverpool má nakročeno na pohárové příčky ...	31. 01. 2016 09:45	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Předehrávka 20. kola Eredivisie slibuje ...	25. 01. 2016 11:44	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Juventus by chtěl do čela Serie A. Proti ...	24. 01. 2016 10:26	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Šlágr kola se o víkend chystá na ...	23. 01. 2016 15:44	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Atraktivní středeční čtvrtfinále! Celta ...	19. 01. 2016 09:52	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Swansea chce pryč od sestupu. Musí však ...	17. 01. 2016 09:30	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	
Vložené 21. kolo Premier League ...	10. 01. 2016 08:47	Zatím neupraveno	0x	Jaromír Hloch	Fotbal	✓	✎ ✖ 🔄 📄 📄	

Obr. 16 Základní obrazovka modulu Správa článků

Pod nadpisem je tlačítko pro vložení nového článků, níže se nachází filtr pro výpis článků. V případě role šéfredaktora obsahuje kromě základního filtru pro období (měsíc a rok) také filtr pro jednotlivé autory článků. Informace o tom, které články (na základě aktuálních hodnot filtru) jsou právě vypsané a kolik jich je, se nachází hned pod filtrem. Nakonec následuje tabulka s výpisem samotných článků.

Články jsou ve výchozím zobrazení seřazeny sestupně dle data publikování. Uživatel však může články v tabulce řadit pomocí šipek v záhlaví podle libovolného sloupce, a to jak vzestupně, tak i sestupně. K této funkcionalitě byla použita javascriptová utilita TableSorter 2.0 (MIT licence). Ta pracuje výhradně na straně klienta, při řazení tedy není žádným způsobem zatížena serverová strana. Při testování této utility však vznikl drobný problém, a sice absence řazení podle data v českém formátu (d. m. Y H:i). Proto bylo žádoucí přidat do zdrojového kódu této utility funkci, která bude schopna parsovat české datum a určit tak jeho pořadí. Zdrojový kód této funkce je přiložen níže.

```
ts.addParser({
  id: "czechDate",
  is: function(s) {
    return false;
  },
  format: function(s, table) {
    s = s.replace(/\\-/g, ".");
    s = s.replace(/(\\d{1,2}) [\\.\\-] [ ] (\\d{1,2}) [\\.\\-] [ ] (\\d{4})/,
      "$3/$2/$1");
    return $.tablesorter.formatFloat(new Date(s).getTime());
  },
  type: "numeric"
});
```

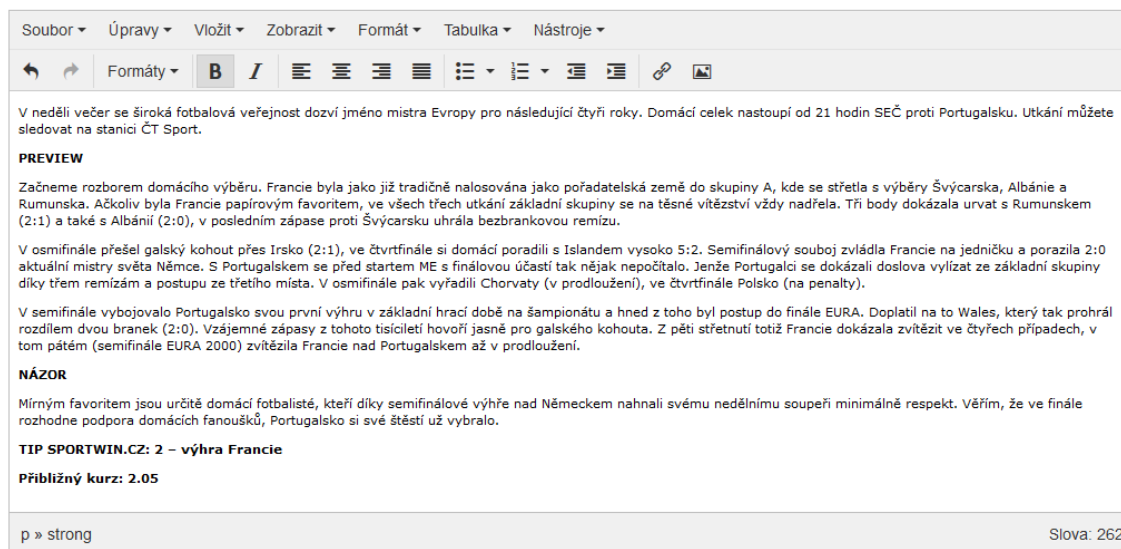
Podobná funkce musela být doimplementována také pro další netradiční formáty dat, např. pro počet zobrazení článku. TableSorter totiž tuto hodnotu (např. 145x) chápal jako textový řetězec a výsledné seřazení tedy bylo chybné.

Jak je patrné z obrázku 16, uživatel má kromě nejrůznějších informací o článku také možnost provádět s jednotlivými články nejrůznější operace. Mezi tyto operace patří úprava, smazání, změna viditelnosti a export článku do formátů DOC, resp. PDF.

Přidání článku

Po kliknutí na tlačítko Přidat nový článek v modulu Správa článků se uživateli zobrazí formulář pro publikování. Uživatel nejprve vyplní nadpis, který je realizován formou textového políčka. Následuje položka pro vložení obsahu článku, který může být formátován autorem článku. Proto bylo klasické textové políčko (textarea) nahrazeno WYSIWYG HTML editorem TinyMCE, který lze dokonce lokalizovat do českého jazyka. Ukázka rozhraní pro vložení obsahu článku je vidět na obrázku 17.

Obsah



The screenshot shows the TinyMCE editor interface. At the top, there is a menu bar with options: Soubor, Úpravy, Vložit, Zobrazit, Formát, Tabulka, and Nástroje. Below the menu is a toolbar with various icons for text formatting (bold, italic, underline, strikethrough, text color, background color), list creation, link insertion, and image insertion. The main content area displays a preview of a football article. The text in the preview is as follows:

V neděli večer se široká fotbalová veřejnost dozví jméno mistra Evropy pro následující čtyři roky. Domácí celek nastoupí od 21 hodin SEČ proti Portugalsku. Utkání můžete sledovat na stanici ČT Sport.

PREVIEW

Začneme rozborem domácího výběru. Francie byla jako již tradičně nalosována jako pořadatelská země do skupiny A, kde se střetla s výběry Švýcarska, Albánie a Rumunska. Ačkoliv byla Francie papírovým favoritem, ve všech třech utkání základní skupiny se na těsné vítězství vždy nadřela. Tři body dokázala urvat s Rumunskem (2:1) a také s Albánií (2:0), v posledním zápase proti Švýcarsku uhrála bezbrankovou remízu.

V osmifinále přešel galský kohout přes Irsko (2:1), ve čtvrtfinále si domácí poradili s Islandem vysoko 5:2. Semifinálový soubor zvládla Francie na jedničku a porazila 2:0 aktuální mistry světa Němce. S Portugalskem se před startem ME s finálovou účastí tak nějak nepočítalo. Jenže Portugalci se dokázali doslova vylízt ze základní skupiny díky třem remízám a postupu ze třetího místa. V osmifinále pak vyřadili Chorvaty (v prodloužení), ve čtvrtfinále Polsko (na penalty).

V semifinále vybojovalo Portugalsko svou první výhru v základní hrací době na šampionátu a hned z toho byl postup do finále EURA. Dopltil na to Wales, který tak prohrál rozdíllem dvou branek (2:0). Vzájemné zápasy z tohoto tisíciletí hovoří jasně pro galského kohouta. Z pěti střetnutí totiž Francie dokázala zvítězit ve čtyřech případech, v tom pátém (semifinále EURA 2000) zvítězila Francie nad Portugalskem až v prodloužení.

NÁZOR

Mírným favoritem jsou určitě domácí fotbalisté, kteří díky semifinálové výhře nad Německem nahnali svému nedělnímu soupeři minimálně respekt. Věřím, že ve finále rozhodne podpora domácích fanoušků, Portugalsko si své štěstí už vybralo.

TIP SPORTWIN.CZ: 2 – výhra Francie

Přibližný kurz: 2.05

At the bottom left of the editor, there is a link "p » strong" and at the bottom right, it says "Slova: 262".

Obr. 17 Pole pro vložení obsahu článku s využitím editoru TinyMCE

Poté ještě zbývá uživateli označit kategorie, do kterých daný článek patří, zvolit témata (tagy), ke kterým se článek vztahuje, nastavit datum vydání a také připojit fotografii ke článku⁴. Tato část formuláře je vidět na obrázku 18.

⁴ Výběr fotografie ke článku je podrobně vysvětlen v kapitole 5.3.2 (Správa fotografií).

Fotografie

Vyber fotku

Kategorie

Fotbal Hokej Basketbal Tenis Ostatní

Související témata

arsenal as říím premier league

Datum vydání

Publikovat ihned
 Naplánovat na

. v v : hodin

Obr. 18 Část formuláře pro publikování článku

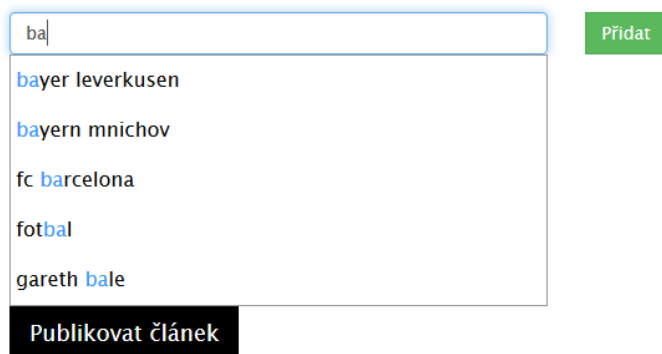
Po kliknutí na tlačítko „Vyber fotku“ se otevře modální okno pro Správu fotografií (je popsáno dále v kapitole 5.3.2). Zařazení článku do kategorií je prosté, probíhá pouze jejich označením (políčka typu checkbox). Poté následuje definice seznamu témat, se kterými publikovaný článek souvisí. K tomu slouží textové políčko spolu s tlačítkem „Přidat“.

Pokud uživatel přidá určité téma, ihned se zobrazí v seznamu pod zmíněným textovým políčkem pro vložení. Nicméně o skutečnosti, že uživatel vložil do seznamu nové téma, musí být informován také formulář pro vložení článku. Takže je nutné dané téma kromě vložení do DOM struktury HTML dokumentu vložit také do skrytého políčka typu pole s názvem `tagList[id]`. Každé téma v seznamu také disponuje možností odstranit sebe samo ze seznamu (červené tlačítko s bílým křížkem u každého tématu na obrázku 18). Pokud uživatel zvolí danou možnost, téma se odebere nejen z viditelného seznamu, ale také ze skrytého pole `tagList[id]`.

Jak bylo uvedeno v kapitole 5.1.2, databáze obsahuje tabulku `tag`, která představuje právě témata (nejen) pro články. Této skutečnosti je využito při automatickém našeptávači, který je součástí textového pole pro zadávání tématu. Ukázka automatického našeptávače je na obrázku 19. Tento našeptávač nabídne všechna témata, která obsahují aktuální podřetězec v textovém políčku pro zadávání témat. Tento podřetězec navíc v nalezených tématech barevně odliší (viz obrázek 19).

Našeptávač je javascriptový plugin, který je založen na jQuery a je vydáván pod MIT licenci. Jeho autorem je Tomáš Kirda.

Související témata



The screenshot shows a search input field with the text 'ba|'. Below the input field, a list of suggestions is displayed: 'bayer leverkusen', 'bayern mnichov', 'fc barcelona', 'fotbal', and 'gareth bale'. To the right of the input field is a green button labeled 'Přidat'. Below the list of suggestions is a black button labeled 'Publikovat článek'.

Obr. 19 Automatický našeptávač pro zadávání témat

Nakonec ještě zbývá nastavit datum vydání. Uživatel může nastavit toto datum buď na aktuální (možnost Publikovat ihned), nebo ho může nastavit ručně, pomocí aktivace možnosti „Naplánovat na“ a následném vyplnění formuláře pro nastavení data.

Jakmile uživatel formulář odešle, systém zkontroluje, zda jsou všechna kritéria pro publikování článku splněna. Konkrétně jsou to tato validační pravidla.

- Nadpis a obsah článku musí být vyplněn.
- Ke článku musí být připojena fotografie.
- Článek musí být zařazen alespoň do 1 kategorie.
- Ke článku musí být definováno alespoň 1 téma.
- Článek musí mít nastavené datum vydání. To obsahuje dvě podmínky. Musí být označena jedna z možností (Publikovat ihned nebo Naplánovat na) a v případě volby druhé varianty musí být zadáno validní datum a čas.

V případě, že jsou tato validační pravidla splněna, systém článek uloží a uživateli se spolu s informativní hláškou zobrazí formulář pro editaci právě vydaného článku. V opačném případě systém uživateli zobrazí seznam nesplněných podmínek, avšak zanechá dosud vyplněné hodnoty ve formuláři. Příklad takového upozornění lze vidět na obrázku 20.

Přidat nový článek

- Vyber prosím fotografii ke článku.
- Zadej prosím alespoň jeden tag ke článku.

Nadpis

Obr. 20 Příklad výpisu nedostatků při publikování článku

Úprava článku

Redakční systém využívá pro upravení článku stejný formulář jako pro jeho vložení s tím, že do něj načte všechny výchozí hodnoty. Nicméně i přesto jsou mezi těmito dvěma případy užití drobné odlišnosti. Ve formuláři pro úpravu je umístěn pod políčkem pro vložení nadpisu odkaz na daný článek, což je možné vidět na obrázku 21.

Upravit stávající článek

Nadpis

URL článku: <http://localhost/sportwin/clanek/9-liverpool-ma-nakroceno-na-poharove-pricky-musi-vsak-porazit-vedouci-leicester>

Obr. 21 Začátek formuláře pro upravení článku.

Mírně odlišný je také dialog pro nastavení data vydání. Uživatel, který daný článek upravuje, totiž nemusí chtít datum vydání vůbec měnit, a tak je žádoucí uživateli tuto možnost nabídnout (na rozdíl od přidání nového článku) a také ho informovat o tom, kdy přesně byl článek vydán. Vše ukazuje obrázek 22.

Datum vydání

Vydáno: 31. 1. 2016 09:45

- Ponechat stávající
 Naplánovat na

. 2016 v : hodin

Upravit článek

Obr. 22 Dialog pro nastavení data vydání při úpravě článku.

Smazání článku

Uživatel má možnost z nejrůznějších důvodů článek odstranit tak, že již nebude dostupný nejen ostatním uživatelům ve veřejné části portálu, ale ani uživateli samotnému v administraci. Jelikož se jedná o operaci, která může jistým způsobem učinit nemalé škody portálu, systém se uživatele po kliknutí na tlačítko pro smazání článku pro jistotu zeptá, zdali uživatel skutečně chce daný článek odstranit. Celý

proces odstranění zajišťuje metoda `actionDelete($articleId)`, která je součástí třídy `ArticlePresenter` v modulu `AdminModule`. Zde je uveden její kompletní zdrojový kód.

```
public function actionDelete($articleId) {
    try {
        $article = $this->articleFacade->findOneById($articleId);
        $this->articleFacade->delete($article);
        $this->flashMessage("Článek s názvem \"{$article->title}\" byl
                            úspěšně smazán.", "alert-success");
    } catch (EntityNotFoundException $ex) {
        \Tracy\Debugger::log($ex);
        $this->flashMessage("Požadovaný článek nebyl nalezen.",
                            "alert-danger");
    } finally {
        $this->redirect("Articles:default");
    }
}
```

Patrně nejzajímavější příkaz z celé metody `actionDelete($articleId)` je volání metody `delete(Article $article)` třídy `ArticleFacade`. Ta totiž zavolá stejnojmennou metodu svého mapperu (třída `ArticleDbMapper`), jejíž zdrojový kód vypadá následovně.

```
public function delete(Article $article) {
    $this->db->table(self::TABLE_NAME)
        ->where(self::COLUMN_ID, $article->id)
        ->update(array(
            self::COLUMN_DELETED => true
        ));
}
```

Ze zdrojového kódu je patrné, že metoda `delete(Article $article)` pouze nastaví atribut daného článku na stav „smazán“, tzn., že jej fyzicky z databáze nevymaže. Toto řešení bylo zvoleno proto, že by přece jen bylo nutné článek obnovit zpět, či v budoucnu by bylo žádoucí v administraci implementovat funkci pro obnovu smazaných článků.

Změna viditelnosti článku

Z technického hlediska funguje případ užití s názvem Změna viditelnosti článku na stejném principu jako předchozí případ, tedy Smazání článku. Ovšem s tím rozdílem, že při tomto případě užití se změna viditelnosti projeví pouze ve veřejné části portálu, v administraci bude daný článek vidět nezávisle na tomto stavu viditelnosti. Pokud uživatel u příslušného článku klikne na tlačítko představující změnu viditelnosti, zavolá se metoda `actionChangeVisible($articleId)`, která nejdříve prostřednictvím metody `findOneById($articleId)` třídy `ArticleFacade` nalezne daný článek a načte jej do instance třídy `Article`.

Na této instanci je následně zavolána metoda `changeVisible()`, která zne-
guje současnou hodnotu atributu `visible`. Nakonec je zavolána metoda `chan-
geVisible(Article $article)`, která zavolá stejnojmennou metodu pří-
slušného databázového mapperu, který se postará o změnu atributu `visible` da-
ného článku. Původně byla pro uložení změny atributu `visible` použita metoda
`save(Article $article)` třídy `ArticleFacade`, nicméně z hlediska opti-
malizace systému byla zvolena výše popsaná varianta. Metoda `save(Article
$article)` totiž ukládá celou entitu `Article`, která je rozsáhlá a má přesah do
více tabulek v databázi.

Export článku

Jak bylo zmíněno v kapitole 4.3.2, uživatel má možnost jednotlivé články expor-
tovat do dvou formátů, a sice do DOC a PDF. Pro tento účel bylo nutné vytvořit novou
HTML šablonu, do které se vloží atributy daného článku a následně se převede do
výsledného formátu, viz dále. Tato šablona se jmenuje `default.latte` a je umís-
těna ve složce `app/AdminModule/templates/export/article`.

Pro převod této HTML šablony do formátu PDF byla použita PHP knihovna
s názvem `mPDF`, jejíž zdrojové kódy jsou v projektu umístěny ve složce `ven-
dor/mpdf`. Její použití je dobře vidět v metodě `actionPdf($articleId)` třídy
`ArticlePresenter`, která má právě na starost převod článku do PDF formátu.
Zdrojový kód této metody je níže.

```
public function actionPdf($articleId) {  
    try {  
        $article = $this->articleFacade->findOneById($articleId);  
        $latte = new \Latte\Engine();  
        $latte->setTempDirectory("../temp/cache");  
        $content["article"] = $article;  
        $content["basePath"] = $this->getHttpRequest()  
            ->getUrl()  
            ->getBasePath();  
        $result = $latte->renderToString(__DIR__ .  
            "/templates/export/article/default.latte", $content);  
        $mpdf = new \mPDF();  
        $mpdf->WriteHTML($result);  
        $fileName = \Nette\Utils\Strings::webalize($article->url);  
        $mpdf->Output("$fileName.pdf", "D");  
        exit();  
    } catch (EntityNotFoundException $ex) {  
        \Tracy\Debugger::log($ex);  
        $this->flashMessage("Požadovaný článek nebyl nalezen.",  
            "alert-danger");  
        $this->redirect("Articles:default");  
    }  
}
```

Kód této metody nejdříve pomocí již několikrát zmiňované metody `findO-
neById($articleId)` třídy `ArticleFacade` načte do instance třídy `Ar-`

`ticle` požadovaný článek. Následně je vytvořen objekt třídy `Engine`, tj. šablona pro export článku. Tomuto objektu je následně nastaven adresář pro vyrovnávací paměť, dále je vytvořeno pole s názvem `content`, do kterého je uložena instance třídy `Article` a také URL adresa portálu, na kterém je metoda zrovna spuštěna.

Tato adresa je v šabloně potřebná pro správné určení absolutní cesty k obrázku, který je součástí každého článku. Poté je do proměnné `$result` uložen textový řetězec, který představuje kód šablony s již dosazenými proměnnými, resp. proměnnou s názvem `content`. Nakonec je vytvořen objekt třídy `mPDF`, do kterého je následně pomocí metody `writeHTML($result)` vložena vytvořená šablona. Výsledný PDF soubor je pak poslán do prohlížeče.

Export do formátu DOC probíhá obdobně s tím rozdílem, že výsledná HTML šablona se nezpracovává pomocí knihovny do výsledného formátu, ale je jí pouze změněna přípona výsledného souboru na formát DOC a také některé parametry hlavičky skriptu pomocí PHP funkce `header()`. Vše je lépe vidět ve zdrojovém kódu níže. Tento kód je součástí metody `actionDoc($articleId)` třídy `ArticlePresenter`.

```
$result = $latte->renderToString(__DIR__ .
    "/templates/export/article/default.latte", $content);
$fileName = \Nette\Utils\Strings::webalize($article->url);
header("Pragma: public");
header("Expires: 0");
header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
header("Content-Type: application/force-download");
header("Content-Type: application/octet-stream");
header("Content-Type: application/download");
header("Content-Disposition: attachment; filename=$fileName.doc");
header("Content-Transfer-Encoding: binary ");
echo $result;
```

5.3.2 Správa fotografií

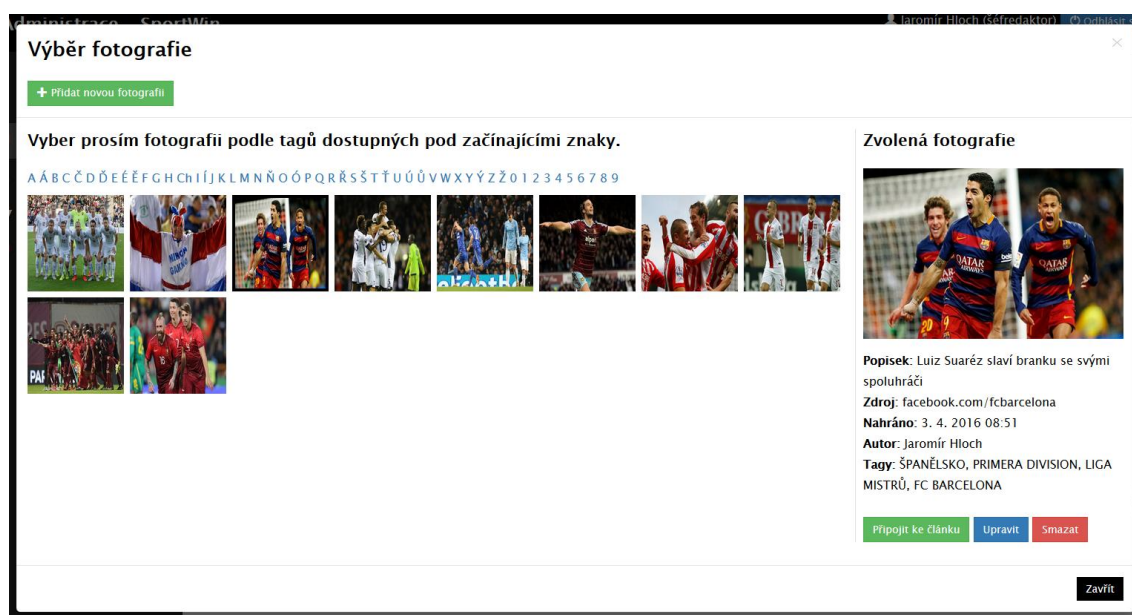
Správa fotografií je úzce spjatá s modulem Správa článků, který byl popsán v předchozí kapitole. Dalo by se tedy říci, že je jeho submodule. Tento modul může uživatel spustit ze dvou případů užití, a sice z přidání a z editování článku, kliknutím na tlačítko „Vyber fotku“ v sekci Výběr fotografie. Tento modul je kompletně řešen pomocí modálních oken, tj. veškeré jeho akce či případy užití jsou řešeny pomocí technologie AJAX.

Jelikož fotografie jsou v tomto systému používány k tomu, aby byly přiřazovány k jednotlivým článkům, cílem tohoto modulu je, aby uživatel pohodlně vybral nejvhodnější fotografii ke článku, který se právě upravuje či přidává. Pokud tedy uživatel aktivuje tlačítko „Vyber fotku“, zobrazí se mu výchozí modální okno s názvem Výběr fotografie.

Ihned pod názvem je umístěno tlačítko „Přidat novou fotografii“, které uživatel využije v případě, že v systému nenalezl vhodnou fotografii pro svůj článek. Pokud však uživatel zamýšlí připojit ke článku fotografii, která se již v systému na-

chází, slouží k tomu jednoduchý vyhledávač. Ten je umístěn nahoře v hlavním levém panelu modálního okna. Jeho princip je následující.

Každá fotografie má k sobě podobně jako článek přiřazeno několik (minimálně však jedno) témat (tagů). Uživatel má tedy možnost hledat fotografie podle témat. A to tak, že nahoře klikne na určitý znak (písmena od A do Z společně s číslicemi 0 až 9). Tím se zavolá metoda `handleLoadPhotosByStartCharTag($startChar)`, která vrátí všechny fotografie, které obsahují téma, které začíná na požadované písmeno. Tím se uživateli zobrazí výsledek, který je vidět na obrázku 23.



Obr. 23 Modální okno pro výběr fotografie poté, co uživatel zvolil začínající znak tématu.

Pokud uživatel aktivuje libovolnou fotografii z výsledku (seznam fotografií vlevo), zobrazí se detail této fotografie v pravém panelu modálního okna. Pravý panel nese podnázev „Zvolená fotografie“, pod ním je zobrazena aktuální vybraná fotografie ve zvětšené velikosti. Pod ní jsou uvedeny informace, které s fotografií úzce souvisí. Jedná se o popis, zdroj, datum nahrání do systému, autora a také související témata. Nakonec jsou ve spodní části pravého panelu umístěna tlačítka pro nejrůznější akce dle toho, jaká má právě přihlášený uživatel práva.

Připojení fotografie ke článku

Jak bylo uvedeno výše v této kapitole, pokud uživatel vybere určitou fotografii z levého seznamu, má možnost ji poté pomocí tlačítka „Připojit ke článku“ přiřadit do formuláře pro přidání/editaci článku. Celou tuto akci zajišťuje metoda `handleJoinPhotoToArticle($photoId)`, která je uvedena ve zdrojovém kódu níže.

```
public function handleJoinPhotoToArticle($photoId) {
```

```
if($this->isAjax()) {
    try {
        $this->template->chosenPhoto = $this->photoFacade
                                     ->findOneById($photoId);

        $this->template->photoId = $photoId;
    } catch (EntityNotFoundException $ex) {
        \Tracy\Debugger::log($ex);
        $this->template->photoId = 0;
    } finally {
        $this->redrawControl("articleWrapper");
        $this->redrawControl("photoField");
    }
} else {
    $this->redirect("this");
}
}
```

V metodě je nejdříve zkontrolováno, zda se jedná o AJAX požadavek. Pokud ano, tak se zavolá metoda `findOneById($id)` třídy `PhotoFacade`, která vrátí instanci třídy `Photo`. Ta je hned uložena do proměnné `chosenPhoto` aktuální šablony. Do šablony je nutné uložit také atribut `id` dané fotografie (kvůli tomu, aby formulář pro přidání/editaci článku věděl, která fotografie byla ke článku přiřazena).

Pokud požadovaná fotografie neexistuje, metoda `findOneById($id)` vyhodí výjimku `EntityNotFoundException`. V tom případě se jako `id` fotografie nastaví 0, která značí, že fotografie nebyla vybrána. Nakonec se musí v šabloně překreslit tzv. snippet s názvem `photoField`, aby byla změna viditelná také uživateli v rozhraní.

Ještě předtím, než se tento snippet překreslí, dojde k automatickému zavření modálního okna, tedy submodulu Správa fotografií. Ve formuláři nyní už není v sekci Výběr fotografie tlačítko „Vyber fotku“, nýbrž miniatura právě vybrané fotografie, která bude přiřazena ke článku.

Tato miniatura zároveň slouží jako odkaz, který po své aktivaci znovu otevře modální okno se Správou fotografií, aby uživatel případně mohl aktuálně zvolenou fotografii změnit.

Přidání nové fotografie

Po kliknutí na tlačítko „Přidat novou fotografii“ ve Správě fotografií se uživateli zobrazí další modální okno, tentokrát s formulářem pro vložení nové fotografie. Tento dialog je vidět na obrázku 34 v příloze B. Nejprve je uživatel vyzván k tomu, aby nahrál ze svého zařízení fotografii.

Následují vstupní textová pole pro zadání popisu a zdroje. Nakonec musí uživatel k nové fotografii přiřadit alespoň jedno téma. Pravidla pro přiřazování témat k fotografiím jsou stejná jako v případě článků. Po odeslání formuláře systém kontroluje všechna validační pravidla, a pokud jsou splněna, tak uloží fotografii a informuje o tom uživatele. V opačném případě uživatele upozorní na nedostatky.

Úprava fotografie

K této operaci se vztahují určitá přístupová pravidla. Upravit fotografii může buď šéfredaktor (ten má právo upravovat všechny fotografie), anebo redaktor, který je zároveň autorem dané fotografie. Jak bylo uvedeno na začátku této kapitoly, tlačítko „Upravit“ se nachází pod informacemi dané fotografie v pravém panelu hlavního modálního okna.

Po aktivaci tohoto tlačítka se zobrazí nové modální okno s formulářem pro přidání nové fotografie, který je ovšem přizpůsoben pro úpravu. Vše je vidět na obrázku 35 v příloze B. Tzn., že je zde vidět miniatura aktuálně upravované fotografie (tu nelze změnit), dále jsou do textových polí popisu a zdroje načteny aktuální hodnoty a také jsou pod políčkem pro vkládání souvisejících témat zobrazeny již vložená témata s možností je odstranit. Jakmile uživatel odešle tento formulář, proběhnou téměř totožná validační pravidla (fotografie se nekontroluje, ta zůstává stejná) a poté je fotografie buď upravena, nebo je uživateli zobrazen výpis nedostatků.

Smazání fotografie

Pro operaci smazání fotografie platí naprosto stejná přístupová pravidla jako pro její úpravu. Tlačítko pro smazání se nachází hned vedle tlačítka „Upravit“. Jakmile uživatel toto tlačítko aktivuje, zavolá se metoda `handleDeletePhoto($photoId, $startChar)`. Ta zajistí nejen fyzické vymazání fotografie z disku, ale také vymazání všech informací o dané fotografii v databázi. Navíc musí ošetřit případy, kdy je daná fotografie součástí některých článků. V takovém případě nastaví metoda těmto článkům výchozí fotografii redakčního systému.

Systém musí také po smazání fotografie aktualizovat seznam v levém panelu, takže musí metoda znovu načíst fotografie podle začínajícího znaku, který je uveden jako druhý parametr metody `handleDeletePhoto($photoId, $startChar)`.

5.3.3 Správa uživatelů

K tomuto modulu má v administraci přístup pouze uživatel, který má přidělenou roli šéfredaktor. Ten může spravovat všechny uživatele v systému včetně sebe. Základní obrazovka modulu Správy uživatelů je vidět na obrázku 24.

Správa uživatelů

[+ Přidat nového uživatele](#)

Filtr:

Uživatel	Email (login)	Registrován	Role	Aktivní?	Blokován?	Akce
Lánský Bořivoj	hloch.jara@gmail.com	11. 12. 2016 12:37	uživatel	✓	✗	
Hloch Jaromír	hlochjaromir@seznam.cz	14. 03. 2016 12:08	šéfredaktor	✓	✗	

Obr. 24 Výchozí obrazovka modulu Správa uživatelů

Toto grafické rozhraní je podobné rozhraní modulu Správy článků. Ihned pod hlavním nadpisem se nachází tlačítko pro vložení nového uživatele do systému. Po aktivaci tohoto tlačítka se šéfredaktorovi zobrazí modální okno s formulářem pro vložení nového uživatele. Ten musí následně vyplnit jeho email, jméno, příjmení a také roli, ve které bude v systému budoucí uživatel vystupovat.

Po odeslání formuláře systém nejdříve zkontroluje vyplněné údaje a v případě jejich správnosti vloží uživatele do databáze a odešle na zadaný email aktivační zprávu, tak jako v případě registrace uživatele. Uživatel musí do 12 hodin od odeslání emailu potvrdit svou registraci kliknutím na aktivační odkaz. Až tehdy je uživatel v systému aktivován a může se přihlásit.

Pod tlačítkem pro přidání nového uživatele se nachází filtr uživatelů. Šéfredaktor může filtrovat na základě tří atributů, a sice podle role uživatele, zda je uživatel aktivní či neaktivní a nakonec zda je blokován či nikoliv.

Následuje výpis uživatelů v podobě tabulky. Data v tabulce lze třídit stejně jako v případě výpisu článků, a sice pomocí javascriptového pluginu TableSorter (popsaný v kapitole 5.3.1). Se stávajícími uživateli v databázi pak má šéfredaktor možnost provádět v zásadě dvě operace. Může uživateli změnit roli a může uživatele zablokovat, resp. odblokovat zpět. V obou, resp. všech třech případech je daný uživatel o konkrétní operaci informován pomocí emailové zprávy. Jako příklad níže je uvedena metoda `changeRoleFormSucceeded(Form $form, $values)`, která se zavolá v případě, že šéfredaktor změní roli ve výběrovém seznamu u některého z uživatelů v tabulce.

```
public function changeRoleFormSucceeded(Form $form, $values) {
    try {
        $user = $this->userFacade->findOneById($values->userId);
        $oldRole = $user->role;
        $user->setRole($values->role);
        $this->userFacade->save($user);
        $this->mailSender->send("zmenarole@sportwin.cz",
            $user->email,
            "Změna role",
            $this->createChangeRoleTemplate(
                $form,
                $oldRole,
                $user->role));
        $form->getPresenter()
            ->flashMessage("Role uživatele byla úspěšně změněna.");
    }
}
```



```
                Změnu však uživatel uvidí až po opětovném  
                přihlášení!",  
                "alert-success");  
    } catch (EntityNotFoundException $ex) {  
        \Tracy\Debugger::log($ex);  
        $form->getPresenter()  
            ->flashMessage("Požadovaný uživatel nebyl nalezen!",  
                "alert-danger");  
    } finally {  
        $form->getPresenter()->redirect("this");  
    }  
}
```

6 Uživatelské testování

Uživatelské testování jednotlivých modulů (resp. částí systému) probíhalo průběžně při vývoji (implementaci) redakčního systému jeho autorem. Nicméně jakmile byl celý redakční systém dokončen, bylo provedeno uživatelské testování nejen autorem, ale také ostatními redaktory, budoucími uživateli tohoto systému. Systém sice nebyl testován v reálném provozu, nicméně testování probíhalo na zcela reálných úlohách a testovací prostředí bylo nastaveno téměř stejně jako prostředí reálné.

Uživatelské testování bylo vcelku úspěšné, žádné závažné chyby, které by ovlivnily fungování celého systému, nebyly nalezeny. Jednotliví uživatelé však ukázaly na pár nedostatků, které by bylo vhodné opravit. Byly nalezeny celkem tři a jsou podrobněji rozebrány v textu níže.

Ochrana proti robotům. Jelikož mohou přihlášení uživatelé komentovat články, mohlo by se stát, že by nějaký uživatel mohl implementovat skript, který by automaticky přidával komentáře ke článku. Byla tedy navržena ochrana proti tzv. robotům formou antispamové otázky. Ta se však uživateli objeví ve formuláři pro přidání komentáře pouze v případě, že daný uživatel přidal do systému dva nebo více komentářů během jedné minuty.

Pravidla pro viditelnost kategorií. Původní myšlenka hlavního menu kategorií byla taková, že by se daná kategorie zobrazovala pouze v případě, že by obsahovala alespoň jeden článek, tj. nebyla by prázdná. Současně však v administraci bylo možné nastavit danou kategorii jako ne/viditelnou. Zobrazování kategorií se tak stalo nekonzistentním. Tím spíše, že v případě, kdy uživatel systému nastavil datum vydání článku na budoucí a zároveň ho přidal do kategorie, ve které ještě žádný článek není, daná kategorie se již zobrazovala v menu i přesto, že v ní v podstatě ještě nebyl žádný článek, pouze ten naplánovaný. Na toto téma tedy vznikla dlouhá diskuze, která nakonec vyústila v následující řešení. Bylo zrušeno pravidlo, které nezobrazovalo v hlavním menu prázdnou kategorii, viditelnost dané kategorie tedy určuje jen a pouze šéfredaktor v administraci v modulu Správa kategorií.

Opětovné posláni aktivačního emailu. Pokud uživatel provedl úspěšně registraci do systému, byl mu automaticky odeslán aktivační email. Nicméně i když funkce v PHP jazyce vrátí pravdivostní hodnotu `true` z metody pro odeslání emailu, přesto nemusí email uživateli fyzicky dorazit do schránky. Pokud se tak stane, tak uživatel je v databázi jako neaktivní a nemůže si účet aktivovat. Email musí navíc být v systému unikátní, tzn., že by kvůli opětovné registraci musel použít jiný email. Proto bylo do systému přidáno tlačítko s možností opětovného posláni aktivačního emailu v případě, že ten automaticky nedorazí. Toto tlačítko se uživateli objeví na obrazovce po odeslání registračního formuláře.

7 Budoucnost redakčního systému

V rámci této diplomové práce byla navržena a implementována základní verze redakčního systému tak, aby byl možný základní chod portálu (tj. aby byl klíčový obsah portálu editovatelný a spravovatelný). Portál však bude do budoucna potřebovat pokročilejší funkce nejen v administrační části, ale především v části veřejné, a to hlavně pro přihlášené uživatele. V současné době jsou tedy v plánu následující moduly či funkcionality.

Tipovací modul. Přihlášení uživatelé by měli možnost účastnit se tipovacích soutěží, které vypíše příslušná odpovědná osoba. Podobu konkrétní tipovací soutěže by definovala v administrační části právě odpovědná osoba. Každá tipovací soutěž by se skládala z několika otázek (tipů) a měla by určité datum uzávěrky. Poté by systém automaticky odpovědi vyhodnotil, výsledky by nakonec byly zveřejněny ve veřejné části portálu. Výherce by obdržel případné ceny, které by poskytl případný sponzor či partner portálu.

Uživatelské blogy. Zaregistrovaní uživatelé by měli možnost vést si vlastní sázkařský blog, na kterém by mohli publikovat své sázkařské úspěchy a zkušenosti ve formě klasických článků. Tento modul by tedy neměl být časově náročný, jelikož publikování článků již na portálu funguje, musela by se pouze upravit práva k jednotlivým možnostem nastavení článků. Např. přístup ke Správě fotografií atd.

Správa reklam. Jelikož portál má v současné době část svých příjmů z placené inzerce, je žádoucí realizovat modul, který bude umožňovat správu reklam, tedy přidání, resp. odstranění jednotlivých reklam umístěných ve veřejné části portálu. V současné době totiž musí reklamy spravovat programátor portálu přímo ve zdrojovém kódu, což není příliš flexibilní ani efektivní.

Spárování rezervací s konkrétními články. Protože drtivá většina článků představuje předzápasový rozbor dvou soupeřů, na který slouží modul Rezervace článků, bylo by vhodné implementovat spárování těchto rezervací s konkrétními články. Autor článku by tedy při jeho publikování vybral z dostupných rezervací tu správnou, tím pádem by systém věděl, že daná rezervace je splněná. Stoprocentnímu zabránění označení nesprávné rezervace zřejmě zabránit nejde, nicméně šlo by jej do určité míry znesnadnit. A to tak, že by systém autorovi článku nabídl označit pouze ty rezervace, které sám do systému zadal, nemají naplánováno datum vydání na minulé a plánované datum vydání souhlasí se skutečným datem vydání.

8 Závěr

V rámci diplomové práce byla nejdříve provedena teoretická rešerše dostupných nástrojů pro tvorbu webových aplikací. Byl popsán jazyk PHP se softwarovou architekturou MVC. Byly rovněž popsány a porovnány nejpoblárnější PHP frameworky současnosti. Zmíněny byly také ostatní nástroje, např. HTML, CSS, Javascript a také databázový systém MySQL.

V praktické části byla prvotně provedena analýza na nový redakční systém. Ta spočívala v rozboru stávajícího systému - zjištění jeho silných a slabých stránek, a v komunikaci mezi jednotlivými uživateli (redaktory, šéfredaktorem) a majiteli portálu. Z provedené analýzy následně vznikly funkční i nefunkční požadavky a také zvolené technologie pro realizaci tohoto redakčního systému.

Ve fázi implementace byla nejdříve definována struktura systému, poté byl vytvořen navržený databázový ER diagram spolu s architekturou modelové vrstvy. Následovala implementace veřejné části, poté byla vytvořena i administrační část. Ta byla tvořena po jednotlivých modulech, po každém dokončeném modulu bylo provedeno dílčí uživatelské testování.

Po dokončení bylo provedeno komplexní uživatelské testování více uživateli, nalezené chyby v systému byly opraveny. Redakční systém je připraven na spuštění v reálném provozu. Cíl diplomové práce byl tedy splněn. Systému byla navíc doporučena další možná vylepšení do budoucna.

Veřejná i administrační část systému je responzivní, uživatelé hodnotili přívětivost systému kladně. Systém je postaven na MVC architektuře a jeho modelová vrstva je rozdělena celkem do čtyř vrstev, systém tedy není závislý na použitém databázovém systému. Přidáním dalšího typu mapperů lze navíc docílit toho, že data lze ukládat do jiného typu úložiště, než je databáze, a to bez dalšího zásahu do zdrojových kódů systému.

9 Literatura

- ACHOUR A KOL., Mehdi. What is PHP? *PHP* [online]. 2016 [cit. 2016-11-15]. Dostupné z: <http://php.net/manual/en/intro-what-is.php>.
- ANAND, Manmeet. *7 Best PHP Framework for Enterprise Application* [online]. 2014 [cit. 2016-11-19]. Dostupné z: <http://www.tisindia.com/blog/7-best-php-frameworks-2014/>.
- AngularJS: HTML enhanced for web apps!* [online]. 2016 [cit. 2016-11-22]. Dostupné z: <https://angularjs.org/>.
- A Brief History of CodeIgniter. *EllisLab* [online]. 2016 [cit. 2016-11-19]. Dostupné z: <https://ellislab.com/codeigniter>.
- A PHP Framework For Web Artisans. *GitHub* [online]. 2016 [cit. 2016-11-16]. Dostupné z: <https://github.com/laravel/laravel>.
- BITTNER, Jan. 2. díl - Výběr technologií 1. *ITnetwork.cz* [online]. 2015 [cit. 2016-11-14]. Dostupné z: <http://www.itnetwork.cz/html-css/webove-aplikace/technologie-pro-vyvoj-webovych-aplikaci-vyber-technologie-1>.
- BITTNER, Jan. 3. díl - Výběr technologií 2. *ITnetwork.cz* [online]. 2015 [cit. 2016-11-14]. Dostupné z: <http://www.itnetwork.cz/html-css/webove-aplikace/technologie-pro-vyvoj-webovych-aplikaci-vyber-technologie-2>.
- Bootstrap 3 Tutorial. *W3Schools* [online]. 2016 [cit. 2016-11-20]. Dostupné z: <http://www.w3schools.com/bootstrap/default.asp>.
- CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.
- CodeIgniter User Guide. *Codeigniter* [online]. 2016 [cit. 2016-11-20]. Dostupné z: http://www.codeigniter.com/user_guide/index.html.
- CodeIgniter. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-19]. Dostupné z: <https://en.wikipedia.org/wiki/CodeIgniter>.
- CSS Introduction. *W3Schools* [online]. 2016 [cit. 2016-11-20]. Dostupné z: http://www.w3schools.com/css/css_intro.asp.
- DB-Engines Ranking. *DB - ENGINES* [online]. 2016 [cit. 2016-11-20]. Dostupné z: <http://db-engines.com/en/ranking>.
- Eloquent: Getting Started. *Laravel* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://laravel.com/docs/5.3/eloquent>.
- GARBADE, Michael J., Dr. How to choose a PHP framework. *Opensource* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://opensource.com/business/16/6/which-php-framework-right-you>.
- Gii Extension for Yii 2. *Yiiframework* [online]. 2016 [cit. 2016-11-19]. Dostupné z: <http://www.yiiframework.com/doc-2.0/ext-gii-index.html>.

- HTML Introduction. *W3Schools* [online]. 2016 [cit. 2016-11-20]. Dostupné z: http://www.w3schools.com/html/html_intro.asp.
- jQuery Introduction. *W3Schools* [online]. 2016 [cit. 2016-11-20]. Dostupné z: http://www.w3schools.com/jquery/jquery_intro.asp.
- KOUTNÝ, Jiří. *Symfony2, těší mě!* [online]. 2013 [cit. 2016-11-18]. Dostupné z: <https://www.zdrojak.cz/clanky/symfony2-tesi-me/>.
- Laděnka, jak se vám líbí? *PhpFashion* [online]. 2008 [cit. 2016-11-16]. Dostupné z: <https://phpfashion.com/ladenka-jak-se-vam-libi>.
- Laravel* [online]. 2016 [cit. 2016-11-20]. Dostupné z: <https://laravel.com/>.
- Laravel Homestead. *Laravel* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://laravel.com/docs/5.3/homestead>.
- Database: Migrations. *Laravel* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://laravel.com/docs/5.3/migrations>.
- Laravel. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-16]. Dostupné z: <https://en.wikipedia.org/wiki/Laravel>.
- Model–view–controller. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-22]. Dostupné z: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
- MySQL Customers. *MySQL* [online]. 2016 [cit. 2016-11-20]. Dostupné z: <http://www.mysql.com/customers/>.
- Nette Framework. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-16]. Dostupné z: https://en.wikipedia.org/wiki/Nette_Framework.
- Nette: Rychlý a pohodlný vývoj webových aplikací v PHP* [online]. 2016 [cit. 2016-11-16]. Dostupné z: <https://nette.org/>.
- Performance of Yii. *Yiiframework* [online]. 2016 [cit. 2016-11-19]. Dostupné z: <http://www.yiiframework.com/performance/>.
- PHP 5 Introduction: What Can PHP Do? *W3Schools* [online]. 2016 [cit. 2016-11-15]. Dostupné z: http://www.w3schools.com/php/php_intro.asp.
- PHP Frameworks: The Best 10 for Modern Web Development. *Noeticforce* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <http://noeticforce.com/best-php-frameworks-for-modern-web-development>.
- PHP MySQL Database. *W3Schools* [online]. 2016 [cit. 2016-11-20]. Dostupné z: http://www.w3schools.com/php/php_mysql_intro.asp.
- PROPST, Oliver. JavaScript. Mozilla Developer Network [online]. 2014, 2014-03-15 [cit. 2014-03-16]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- Symfony Contributors. *Symfony* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://symfony.com/contributors>.

- Symfony. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-18]. Dostupné z: <https://en.wikipedia.org/wiki/Symfony>.
- The Best PHP Framework for 2015: SitePoint Survey Results. *Sitepoint* [online]. 2015 [cit. 2016-11-15]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>.
- Unikátní vlastnosti Nette. *PHP triky: Weblog o elegantním programování v PHP pro mírně pokročilé* [online]. 2009 [cit. 2016-11-16]. Dostupné z: <https://php.vrana.cz/unikatni-vlastnosti-nette.php>.
- Usage of JavaScript libraries for websites. *W3Techs* [online]. 2016 [cit. 2016-11-20]. Dostupné z: https://w3techs.com/technologies/overview/javascript_library/all.
- Usage of server-side programming languages for websites. *W3Techs* [online]. 2016 [cit. 2016-11-15]. Dostupné z: https://w3techs.com/technologies/overview/programming_language/all.
- VRÁNA, Jakub. *1001 tipů a triků pro PHP*. Brno: Computer Press, 2012, 442 s. ISBN 978-80-251-2940-1.
- What is HTML? W3C HTML [online]. 2016 [cit. 2016-11-20]. Dostupné z: <https://www.w3.org/html/>.
- What is symfony? *Symfony* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://symfony.com/what-is-symfony>.
- KRILL, Paul. Why developers love and hate PHP. InfoWorld: FROM IDG [online]. 2014 [cit. 2016-11-22]. Dostupné z: <http://www.infoworld.com/article/2852329/php/reasons-for-developers-to-love-hate-php.html>.
- Yii. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-11-19]. Dostupné z: <https://en.wikipedia.org/wiki/Yii>.
- Yii2 Application Development Cookbook - Third Edition. *Packt* [online]. 2016 [cit. 2016-11-19]. Dostupné z: <https://www.packtpub.com/web-development/yii2-application-development-cookbook-third-edition>.
- Yii PHP Framework Version 2. *Yiiframework* [online]. 2016 [cit. 2016-11-19]. Dostupné z: <http://www.yiiframework.com/doc-2.0/>.
- ZAKAS, Nicholas Z. *JavaScript pro webové vývojáře: Programujeme profesionálně*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2009, 832 s. ISBN 978-80-251-2509-0.

Přílohy

A Ukázky veřejné části

eFotbal.cz | premiership.cz | milujufotbal.cz | fotbal.hattrick.cz | hokej.hattrick.cz | florbal.hattrick.cz | probasket.cz | dokonalazena.cz

SPORTwin.cz
...bavme se sportem

Fotbal Hokej Basketbal Tenis Ostatní Kontakt Všeobecné podmínky

Atraktivní sázkařské tipy

Poslední zápas ME 2016 určí celkového vítěze. Favoritem je pořadatelská země

Jaromír Hloch PŘED 5 MĚSÍCI Fotbal Počet komentářů: 2



Dva giganti změní síly ve druhém semifinále. Uspěje domácí celek nebo favorit ME?

Jaromír Hloch PŘED 5 MĚSÍCI Fotbal bez komentáře

První finalista letošního ME vzejde z duelu Portugalsko – Wales

Jaromír Hloch PŘED 5 MĚSÍCI Fotbal bez komentáře




Přihlášení

Email

Heslo

Přihlásit se

Nemáš ještě účet? Nebo si zapomněl heslo?

Facebook



SportWin.cz
2 216 To se mi líbí

Tohle se mi líbí

To se líbí vám a 4 přátelům

Reklama



MEGAKNIHY



Obr. 25 Část hlavní strany veřejné části portálu

eFotbal.cz | premierшип.cz | milujujufotbal.cz | fotbal.hattrick.cz | hokej.hattrick.cz | florbal.hattrick.cz | probasket.cz | dokonalazena.cz

SPORTwin.cz
...bavme se sportem

Fotbal Hokej Basketbal Tenis Ostatní Kontakt Všeobecné podmínky

Předvede v odvetě Tottenham zázračný obrat a postoupí přes Dortmund?

Jaromír Hloch PŘED 9 MĚSÍCI Fotbal Počet komentářů: 1



Marco Reus slaví branku se svými spoluhráči (zdroj: facebook.com)

Ve čtvrtek jsou na programu osmifinálové odvety Evropské ligy. Jedním z atraktivních duelů bude bezesporu souboj mezi Tottenhamem a Borussií Dortmund. Utkání začíná ve 21:05 hodin SEČ, přímým přenosem jej vysílá stanice Nova Sport 1.

PREVIEW

Borussia Dortmund předvedla v úvodním osmifinálovém zápase perfektní výkon a doma „Kohoutům“ nedala šanci. Smetla je třemi góly a do londýnské odvety si tak veze luxusní třígólový náskok (0:3). Dvěma brankami se předvedl Marco Reus, jednou se trefil Aubamevang.

Přihlášen

Jaromír Hloch (šéfredaktor) [Odhlásit se](#)
Vstoupit od administrace

Facebook



Reklama




Obr. 26 1. část obrazovky s detailem článku

I přes velký nezdar v úvodním osmifinále EL je na tom Tottenham v Premier League velice dobře. Po třiceti kolech se nachází na druhém místě hned za vedoucím Leicesterem, na který ztrácí pět bodů. Na třetí Arsenal má šestibodový náskok, ovšem „Gunners“ mají o zápas méně.

Borussia Dortmund je v Bundeslize v podobném postavení jako Hotspur v Premier League. Aktuálně okupuje druhou příčku, na vedoucí Bayern ztrácí rovněž pět bodů. Nicméně na třetí Herthu má náskok neuvěřitelných šestnácti bodů.

Hertha je tedy z boje o druhé místo už dávno diskvalifikována. „Žlutočerní“ se s „Kohouty“ utkají ve čtvrtek teprve podruhé ve své historii. Pro hosty z Dortmundu bude největší hrozbou domácí Argentinec Erik Lamela, naopak domácí budou muset dát velký pozor na útočné duo Reus – Aubameyang.

NÁZOR

Je velmi nepravděpodobné, že by Tottenham dokázal otočit nepříznivý vývoj tohoto dvojzápasu. Dortmund sice nebude mít takovou sílu jako v domácím klání, ovšem myslím si, že remízu je v pohodě schopný uhrát.


TIP SPORTWIN.CZ: 0 – remíza

Přibližný kurz: 3.60

Související témata

[ANGLIE](#)[PREMIER LEAGUE](#)[EVROPSKÁ LIGA](#)[NĚMECKO](#)[BORUSSIA DORTMUND](#)[BUNDESLIGA](#)

Komentáře

 **Hloch Jaromír před okamžikem** ✖

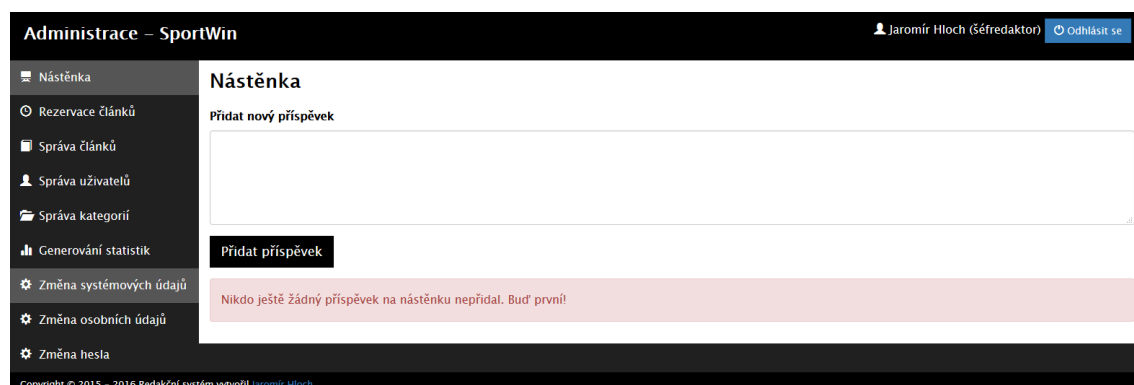
Vyhraje Tottenham, protože má menší marodku!

Napiš svůj názor!

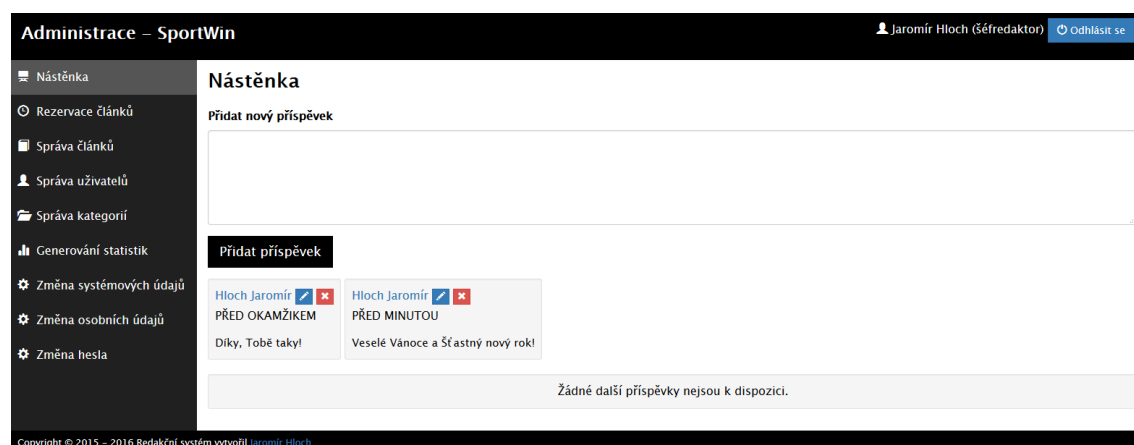
Přidat komentář

Obr. 27 2. část obrazovky s detailem článku

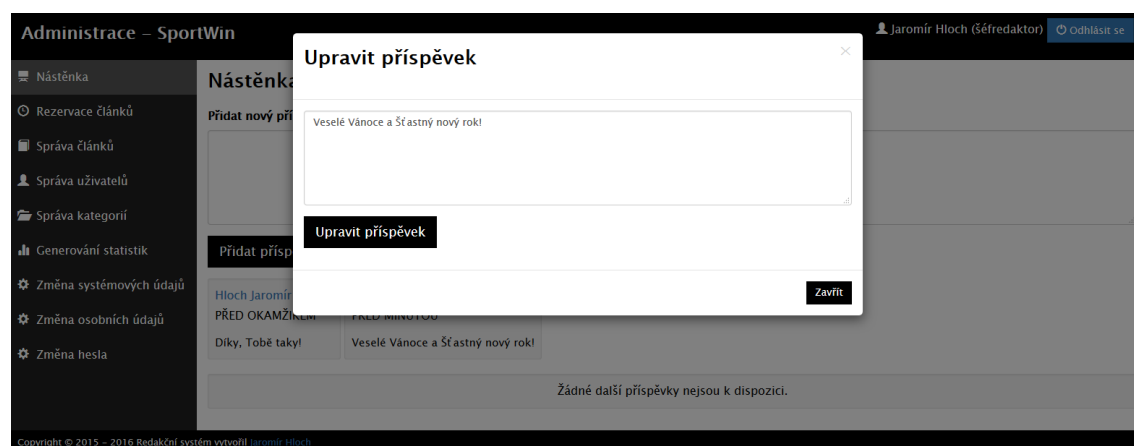
B Ukázky administrační části



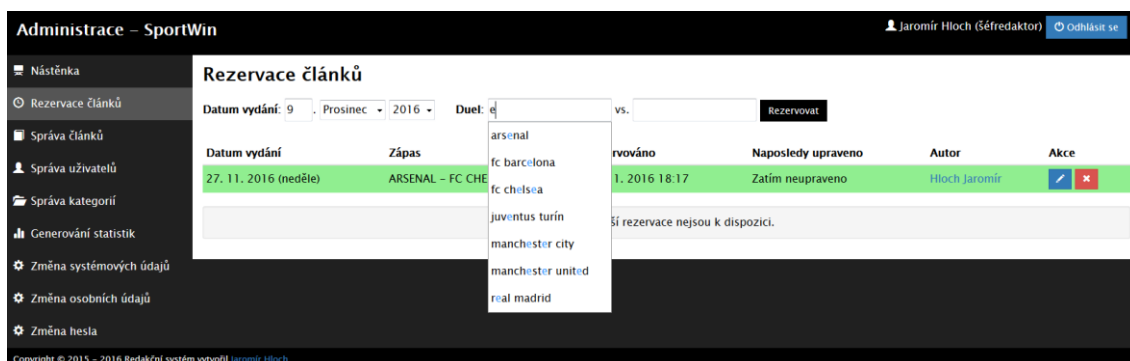
Obr. 28 Ukázka modulu nástěnky před tím, než vloží uživatel první příspěvek.



Obr. 29 Ukázka modulu nástěnky, která právě obsahuje dva příspěvky.



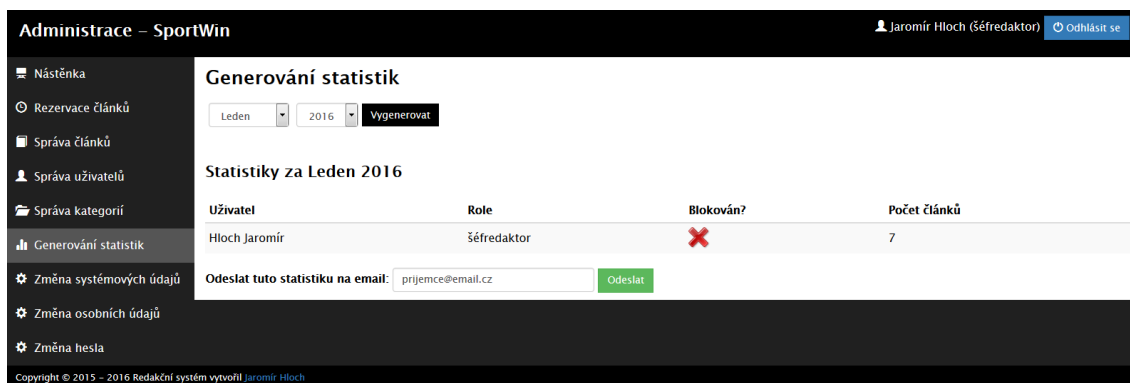
Obr. 30 Upravit příspěvek na nástěnce lze pomocí modálního okna.



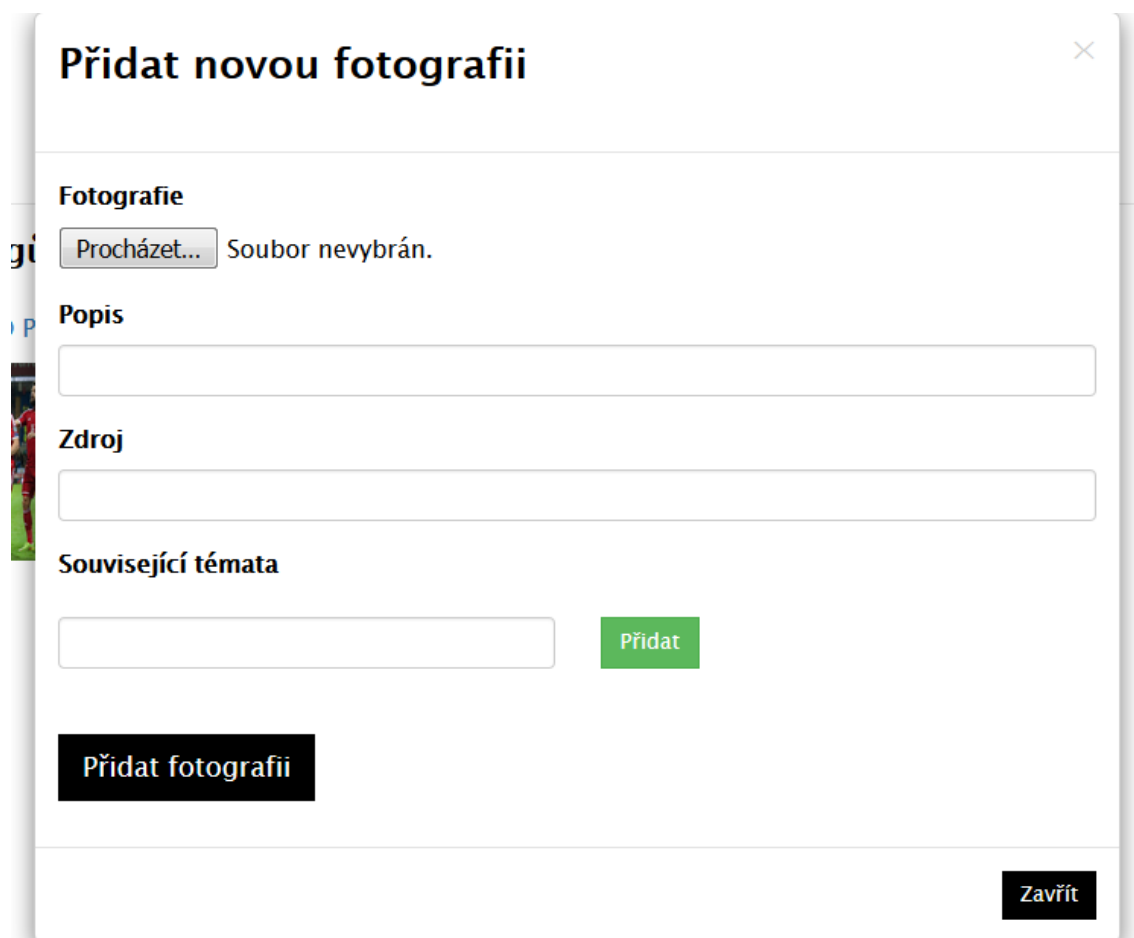
Obr. 31 Ukázka modulu Rezervace článků.



Obr. 32 Úprava rezervací je řešena pomocí modálního okna.



Obr. 33 Uživatel právě vygeneroval statistiku za leden 2016, kterou následně může odeslat na email vybranému příjemci.



Přidat novou fotografii ×

Fotografie
Procházet... Soubor nevybrán.

Popis

Zdroj


Související témata
 Přidat

Přidat fotografii

Zavřít

Obr. 34 Modální okno pro vložení nové fotografie do systému.

Upravit fotografii ×



Popis

Zdroj

Související témata

německo × bundesliga × wolfsburg ×

Upravit fotografii

Zavřít

Obr. 35 Modální okno pro upravení informací o fotografii v systému.

C Zdrojové kódy

Redakční systém pro sportovní portál je verzován a jeho aktuální zdrojové kódy jsou k nahlédnutí na adrese <https://github.com/jaromir92/Sportwin>.