

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta



Diplomová práce

Návrh a implementace redakčního systému do WWW

Bc. Tomáš Kliner

© 2014 ČZU v Praze

!!!

**Místo této strany vložíte zadání diplomové práce.
(Do jedné vazby originál a do druhé kopii)**

!!!

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh a implementace redakčního systému do WWW" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31. 3. 2014

Poděkování

Rád bych touto cestou poděkoval Ing. Jiřímu Brožkovi, který mi byl značně nápomocen jako vedoucí této práce a dále pak také Mrg. Janě Honcové a Bc. Tomáši Henslovi z občanského sdružení za poskytnutou pomoc při psaní této diplomové práce.

Návrh a implementace redakčního systému do WWW

Analysis and implementation of website content management system

Souhrn

Tato práce se věnuje analýze a návrhu nového redakčního systému pro občanské sdružení, které provozuje webové stránky o hudbě a dále také provozuje internetové rádio. Aktuálně občanské sdružení využívá 8 let starý CMS systém, na kterém v průběhu času byly prováděny nadstavby a implementace nových funkcí. Avšak jednotlivé zásahy bez celkového konceptu ve výsledku udělaly z redakčního systému nepřehlednou a těžkopádnou aplikaci, která nikdy nefungovala ani na 80%. V rámci analýzy současného systému a analýzy uživatelských potřeb bylo po vzájemné dohodě s občanským sdružením rozhodnuto, že je potřeba vytvořit kompletní CMS systém znovu.

Další část práce se věnuje vývoji frameworku, který bude tvořit jádro nového redakčního CMS systému a bude řešit všechny důležité systémové úlohy a systémové funkčnosti. K tomuto jádru se později mohou doplňovat pouze jednotlivé nové funkčnosti, které by v budoucnu s rozvojem webových stránek mohly být požadovány.

V závěrečné fázi této práce bude systém implementován na aktuální hardwarové řešení, ale především na současnou databázi, která obsahuje data za posledních 7 let a některá data byla po tuto dlouhou dobu špatně ukládána. Tato data byla například ukládána ve špatném formátu či bez jasné integrity se související daty.

Výsledkem této práce by mělo být vytvoření základní verze nového CMS systému, který bude plně modulární a bude tak do budoucna usnadňovat rozvoj celého systému.

Summary

This work is devoted to the analysis and the proposal of a new content management system for the civil association which keeps website about music and also carries out an internet radio. This civic association has been using the old cms system for 8 years. There have been made some upgrades and implementations, but this individual adjustments without an overall concept resulted in the confusing content management system and cumbersome application. So this application has never been working even for 80 %. In an analysis of the current system and user's needs the civil association decided that it is necessary to create a completely new CMS system.

The next part of this work deals with the development of framework, which will form the kernel of a new content management system and deals with all important system tasks and system functionality. This kernel can be later supplemented by new functions, which could be requested by the development.

In the final phase of this work, the system will be implemented on the current hardware solutions, but mainly on the current database which contains data for the past 7 years. During this time some data has been incorrectly stored. For example they were stored in the wrong format or without clear data integrity. The result of this work should be the basic version of the new CMS system which will be fully modular and will help in the future development of the entire system.

Klíčová slova: CMS, WCMS PHP, OOP, MVC, Framework, CSS, Javascript, Ajax, MySql, Databáze

Keywords: CMS, WCMS, PHP, OOP, MVC, Framework, CSS, Javascript, Ajax, MySql, Database

Seznam zkratek:

WCMS	Web content management system
GUI	Graphic user interface
UML	Unifikovaný modelovací jazyk
MVC	Model-View-Controller
MVP	Model-View-Presenter
HTML	HyperText Markup Language
DOM	Document Object Model
CSS	Cascading Style Sheets
SŘBD	System řízení báze dat
URL	Uniform Resource Locator
MySQL	Databázový system

Obsah:

Obsah:.....	4
1. Úvod	5
2. Cíle a metodika.....	8
2.1. Cíle.....	8
2.2. Metodika	10
3. Teoretická východiska	14
4. Vlastní práce.....	23
4.1. Analýza	23
Sběr informací	23
Sběr informací od zainteresovaných osob	23
Zhodnocení nasbíraných dat	36
4.2. Navrhované řešení	42
4.3. Návrh uživatelského rozhraní.....	51
4.4. Návrh jádra systému	54
4.5. Konkrétní implementace	64
4.6. Nasazení a testování.....	66
5. Závěr	70

1. Úvod

V úvodu této diplomové práce bude čtenář seznámen s důvody, které vedly autora k výběru tématu tvorby internetového administračního systému. Autor shrne aktuální stav na poli vývoje wcms (web content management system) a popíše hlavní problém, respektive hlavní cíl této práce.

Internetové technologie se za posledním pár let změnily více než dost. Čím více se stal internet přístupnější a běžnější pro uživatele a nadšence, tím více se celosvětová síť dostala do zájmů podnikatelů a mezinárodních korporací. Dnes můžeme vidět, že skoro miliarda lidí používá sociální síť facebook, kde denně vystavuje své fotky a šíří své myšlenky. Jídlo si můžeme objednat prostřednictvím webové stránky přímo až před dveře kanceláře anebo bytu. Tištěné noviny a magazíny se již několik let setkávají s úbytkem předplatitelů a kupujících, jednoduše proto, že informace jsou k dispozici zdarma právě na internetu.

Vyšší konkurenci se museli přizpůsobit všichni a ti co to neudělali, buď výrazně ztrácí anebo již zkrachovali a museli ukončit činnost. Přece jen, když dva dělají totéž, nedělají to stejně. Proto dnes společnosti investují milionové a miliardové částky do IT struktury a především do kvalitních informačních systémů, které nejen zvyšují pracovní efektivitu, ale nabízí také další možnosti práce s daty. Na základě kvalitně zpracovaných dat informačním systémem je možné se kvalitně rozhodovat. Efektivní informační systém pracovníky motivuje a zároveň nabízí možnosti jak práci pracovníka urychlit či co nejvíce zefektivnit. Na vyšších úrovních informační systém poskytuje vedoucím pracovníkům údaje pro podporu rozhodování, ale také základní metriky vývoje společnosti jako celku. Dobrý informační systém je dnes již samozřejmostí a to především z důvodu, že objem dat po celém světě neustále narůstá. I menší společnosti potřebují uchovávat a skladovat velké objemy dat, které posléze podrobí v čase analýze.

Jelikož se autor delší dobu věnuje vývoji webových stránek nebo aplikací v jazyce PHP a hledal v té době vhodnou inspiraci na téma Diplomové práce, přišlo mu velice zajímavé, navrhnout pro střední webový magazín a internetové rádio systém WCMS úplně nový, který by zefektivnil práci celého týmu a zároveň by na uživatele působil dobrým dojmem prostřednictvím designu a GUI.

Aktuálně se na poli webových systémů pro management obsahu webové stránky stalo populárních několik systémů, jako je například Word Press anebo JOMLA. Autor této práce chce však vytvořit nový informační systém s dalšími prvky, než je “pouhá” správa a vytváření obsahu. Nový systém pro občanské sdružení by měl po dokončení nabízet efektivnější práci a příjemnější uživatelské prostředí.

Nový systém nebude klasickým WCMS systémem, ale bude v sobě obsahovat prvky firemních ERP systémů pro plánování podnikových zdrojů. Právě tyto prvky by měly zefektivnit vnitřní redakční procesy tak, aby systém vyhovoval potřebám internetového rádia. Základní rozdělení bude na pracovníky, kteří budou obsluhovat pouze web a na pracovníky, kteří obsluhují rádio. Do systému potřebují mít pracovníci také přístup z důvodů aktualizace obsahu jednotlivých pořadů na webových stránkách.

Cílem této diplomové práce je efektivně a důkladně zanalyzovat aktuální stav a uživatelské potřeby napříč podnikovou organizační strukturou v rámci aktuálního informačního systému redakce rádia HipHopStage. Na základě analýzy dostupných informací bude proveden návrh nového informačního a redakčního systému, jeho úprava či zrušení a následné vytvoření nových funkcí systému. Po schválení návrhu bude provedeno vytyčení základních funkcí, které jsou zásadně potřebné pro fungování webového magazínu a rádia. Tyto funkce budou naprogramovány v rámci této diplomové práce, méně důležité funkce a rozvoj celého systému bude prováděn v budoucnosti s ohledem na budoucí potřeby.

K tomu, aby mohl být cíl této práce splněn, bude potřeba realizovat následující kroky.

- Sběr informací
- Analýza dostupných informací
- Vytvoření uživatelských požadavků a uživatelských rolí
- Návrh nového redakčního a informačního systému
- Tvorba jádra nového redakčního a informačního systém
- Implementace redakčního a informačního systému
- Spuštění redakčního a informačního systém

2. Cíle a metodika

Při pohledu na celý projekt jako na obecný informační systém, bylo definováno několik dílčích cílů a kritérií, které musí být provedeny a splněny, aby mohl systém fungovat komplexně. Splněním dílčích cílů bude splněn také hlavní cíl práce, kterým je návrh a implementace nové podoby administračního systému.

2.1. Cíle

Analýza

Prvním dílčím cílem této práce je analýza současného informačního systému, který aktuálně pracovníci využívají. Důraz bude kladen na pochopení současných potřeb celého týmu, které budou doplněny o předpokládané budoucí potřeby v rámci plánovaného rozvoje společnosti. Analýza poskytuje cenná data pro potřeby návrhu nového systému a pojetí jeho celkové koncepce tak, aby nový systém vyhovoval širokému spektru uživatelů.

Skrze analýzu vztaženou nikoliv pouze na informační systém, ale také na práci pracovníků s informačním systémem budou na výstupu k dispozici informace o působení současného systému na psychiku pracovníků a jejich vnímání. Důležitým parametrem pro tuto diplomovou práci bude především fakt motivování či demotivování pracovníků při práci se stávajícím systémem.

Samostatné analýze budou podrobena také současná data, která má společnost k dispozici a která souvisí přímo s informačním systémem či webovými stránkami. Hodnotit se bude především struktura uložených dat a možnost jejich úpravy či kompletní změny pro potřeby přechodu na nový informační systém.

Návrh nového systému

Na základě dat, která jsou k dispozici z předešlé analýzy, bude navržen nový systém, který bude brát zřetel na výstup analýzy, ale také na aktuální trendy v oborech jako jsou informační systémy, management, marketing anebo práce s analytickými daty.

Nejprve bude zvolena vhodná architektura nového informačního systému, která by měla co nejvíce vyhovovat současným potřebám informačního systému, ale také budoucím potřebám rozvoje celého systému a především rozvoji nových aktivit společnosti. Na základě vhodně zvolené architektury bude vytvořen nový datový model informačního systému a upravena část databáze, dle potřeb nového systému. Součástí návrhu dále budou nové uživatelské funkce či modifikované uživatelské funkce současného informačního systému. Na základě analýzy uživatelských potřeb bude návrh obsahovat doporučení na zakoupení konkrétní HTML šablony, která následně bude zpracována a nasazena na architekturu nového informačního systému.

Implementace

Po dokončení návrhu nového systému a po ukončení programátorských prací jádra systému, bude systém implementován na současná data a současnou datovou strukturu. Před samotnou implementací již budou vyřešeny problémy s datovou strukturou a problematické celky dat budou modifikovány. Předpokládá se proto, že implementace proběhne bez větších problémů, jelikož celý systém byl vyvíjen na kopii datové struktury současného informačního systému.

Po ukončení nasazení systému bude nový informační systém zaveden do provozu a řádně otestován. V případě, že testování proběhne bez komplikací anebo budou problémy rychle odstraněny, proběhne zavedení do ostrého provozu.

2.2. Metodika

Pro toto konkrétní téma byly vybrány jako nejvhodnější metodiky ve formě Analýzy a Syntézy. Analyzován je současný informační systém společnosti, který je aktuálně využíván a v rámci návrhu nového informačního systému bude návrh vycházet částečně z výstupu analýzy. Dále budou do návrhu zahrnuty další dostupné informace, jak z oboru informačních technologií, tak také z dalších oborů, které však úzce souvisí s vývojem analytických funkcí informačního systému.

V rámci analýzy je celý systém rozdělen do několika menších bloků, které se analyzují samostatně, a to především vzhledem k jejich velké obsáhlosti. Současný informační systém byl pro potřeby analýzy rozdělen na níže uvedené dílčí celky:

- Celková funkčnost informačního systému
- GUI – grafický vzhled informačního systému
- Datová struktura informačního systému
- Samotná práce pracovníků se systémem a jejich pracovní postupy
- Bezpečnost informačního systému

Prvním krokem z komplexního pohledu na analýzu bude sběr informací od pracovníků a zaměstnanců. Sběr těchto dat probíhá s ohledem na konkrétní oprávnění jednotlivých uživatelů a bude realizován formou osobních konzultací a pozorování. Primárním zdrojem informací jsou informace obdržené od manažera společnosti a šéfredaktorky webového magazínu. Dále bude použit unifikovaný modelovací jazyk UML [8], v rámci kterého budou pro potřeby analýzy využity use case diagramy.

Pomocí use case diagramů [8] jsou analyzovány konkrétní stávající funkčnosti systému a také proběhl návrh nových funkcí. Use case diagram uživateli pomůže v tom

smyslu, že by měl být schopen lépe pochopit novou funkčnost systému anebo naopak lépe vysvětlil problém s funkčností současného systému, jelikož se nejedná o IT odborníky.

Dalším krokem je sběr vlastních podkladů, který probíhal formou pozorování a zapisování. V rámci pracovní schůzky dostal konkrétní pracovník za úkol zaznamenat základní a nejčastěji prováděné úkony v informačním systému. Jeho pracovní postup byl podroben osobní analýze autora a byly také provedeny zápisky.

Analýza datové struktury celé aplikace byla provedena procházením jednotlivých tabulek v databázi. Na základě těchto dat byl vyhotoven ERP diagram [1], který reprezentuje současný stav databáze nynějšího systému. Na základě tohoto diagramu byl návrh databáze podroben další analýze a výstupem této analýzy jsou potřebné změny, které se musí před implementací provést.

Na základě výstupu analýzy byl proveden výběr odpovídající architektury informačního systému a také byl proveden výběr vhodných technologií. Tyto technologie budou využity při tvorbě návrhu a samotného informačního systému. Po vytvoření prototypu jádra bude analýza využita pro návrh a vývoj dalších funkcí nového informačního systému anebo modifikaci současných funkcí.

Následně proběhlo všeobecné hledání aktuálních trendů novinek a nejnovějších postupů v programování webových aplikací. Toto vyhledávání probíhalo především na internetu a použité informační zdroje byly řádně citovány. Díky získaným informacím bylo možné navrhnout takový systém, který odpovídá aktuálním požadavkům WCMS [10] a analytických aplikací.

Návrh nového systému vychází z použitých technologií, které mají zásadní vliv na návrh nových komponent a také jeho celkovou funkčnost. Pro nový informační systém budou využity následující internetové technologie:

- Programovací jazyk PHP
- HTML a CSS
- JavaScript
- MySQL databáze

Spojením těchto technologií bude možné vytvořit přehlednou, moderní a interaktivní webovou aplikaci, která bude obsahovat základní analytické funkce. Jelikož si společnost pronajímá vlastní virtuální server u soukromé společnosti, který dle dostupných informací vyhovuje navrhovanému použití technologií, nebudu se proto v návrhu věnovat serverové části systému. Serverová část včetně technické podpory je zajištěna třetí stranou.

Samotný návrh jádra nového informačního systému se bude řídit pravidly architektury MVC [13] (model, view, controler), která je v současné době velice populární a na principech této architektury funguje i řada největších PHP frameworků jako je například Nette anebo Zend Framework. Tato architektura poskytuje řadu výhod a plně podporuje principy objektového programování, kterých je v této diplomové práci využito.

Pro osobní účely vývoje systému, co se týče přehlednosti, byly využity další diagramy z UML jako je například diagram tříd anebo ERP model databáze. Jelikož se jedná o složitější objektový systém, budou tyto diagramy využívány pro potřeby vývoje (kódování), nikoliv k reprezentování systému jako celku. Počátečním krokem v rámci samotného kódování při tvorbě nového systému bylo vytvoření jádra systém s názvem KERNEL. Toto jádro bude řídit chod celého systému jako celku, po dokončení programování základního jádra jsou přidány další funkce a jednotlivé moduly aplikace.

Po dokončení práce na jednotlivých modulech aplikace bylo přistoupeno k testování rychlosti aplikace. To probíhalo analýzou informací z vývojářského rozšíření Debugger prohlížeče Firefox. Na základě získaných dat proběhla optimalizace obrázků a zdrojového kódu kaskádových stylů. Proběhla také samotná optimalizace jádra systém, a to

úpravou systémových tříd. Dále byl při vývoji nového informačního systému použit Nette Debugger, pomocí kterého bylo možné rychle odstranit konkrétní chybu v kódu, jelikož nabízí daleko podrobnější chybové hlášení a zvýraznění chybového řádku, na rozdíl od klasického hlášení chyb jazyka PHP [9].

K návržení nového uživatelského rozhraní bylo využito on-line softwaru MockFlow.com, který umožňuje návrh uživatelských rozhraní napříč platformami. Tato aplikace byla využita pro tvorbu základních wireframů. Na základě analyzovaných dat ohledně všeobecného povědomí pracovníků o stávajícím grafickém rozvržení aplikace, bylo vytvořeno několik wireframů, které byly následně konzultovány s manažerem projektu a po jejich schválení byly zapracovány do nového systému.

3. Teoretická východiska

Po zhodnocení technologických možností současného informačního systému a nového informačního systému bylo rozhodnuto, že vývoj bude probíhat za pomoci následujících technologií.

Pro analýzu, návrh a abstrakci navrhovaného systému bylo využito technologie UML (Unified Modeling Language), která slouží v softwarovém inženýrství pro grafické znázornění, vizualizaci, navrhování a dokumentaci projektů. Důvodem, proč bylo vybráno UML [8] je fakt, že široce podporuje objektově orientovaný přístup k analýze, návrhu a popisu programovaného systému jako celku. UML je velice pružným jazykem a lze ho využít jako podpůrný nástroj pro komunikaci mezi vývojáři a zákazníkem a také pro zaznamenání myšlenek. Dále lze UML využít pro kreslení detailních návrhů systému nebo jako programovací jazyk, kdy vývojář navrhne UML diagramy tak, aby z nich mohl být vygenerován spustitelný kód. Technologie UML je v současné době považována za standard při vývoji informačních systémů a využívá ji řada firem. Oblíbenou se stala také z důvodu, že UML neobsahuje konkrétní způsob, jak se má používat a neobsahuje ani konkrétní metodiku, kterou lze uplatnit na analýzu či navrhovaný systém. Poskytuje tak vývojovým pracovníkům značnou volnost avšak za využití unifikovaných nástrojů, které UML nabízí.

Níže je přiložen seznam nejpoužívanějších diagramů v rámci jazyka UML, které jsou rozděleny do tří základních skupin:

a) strukturní diagramy

- diagram tříd
- diagram komponent
- composite structure diagram
- diagram nasazení
- diagram balíčků
- diagram objektů

b) diagramy chování

- diagram aktivit
- diagram užití
- stavový diagram

c) diagramy interakce

- sekvenční diagram
- diagram komunikace
- interaction overview diagram
- diagram časování

Samotné programování bude probíhat v jazyce PHP [9], které bylo původně napsáno v jazyce PERL a posléze přepsáno v do jazyka C. I díky tomu má jazyk PHP velice podobnou syntaxi jako jazyk C a jako všechny moderní programovací jazyky podporuje také objektové programování. Nevýhodou možná je, že se jedná o interpretovaný, nikoliv kompilovaný jazyk. Jazyk PHP disponuje celou řadou užitečných funkcí, které programátorovi usnadní práci. PHP bude využíváno ve výchozím nastavení, které přednastavili tvůrci tohoto jazyka. Pro základní aplikaci nemusíme využívat žádného rozšíření, kterých je celá řada. Celý programovací jazyk využívá konfigurační soubor php.ini, který obsahuje základní nastavení. V tomto konfiguračním souboru lze nastavit

celou řadu dodatečných funkcí, od zvětšení limitu pro upload souborů na server až po limity přidělených systémových prostředků.

Další výhodou jazyka PHP jsou jeho vnitřní funkce, které umí pracovat samostatně se soubory (například se soubory jednotlivých tříd) a pomocí SPL funkcí [14] lze třídy před startem samotného systému načíst do paměti serveru. Tomuto postupu se říká autoload tříd a lze tak urychlit práci s velkým množstvím tříd, které větší systémy obvykle obsahují. Třídy jsou načtené v paměti serveru a čekají jen na příkaz programátora, který konkrétní třídu inicializuje.

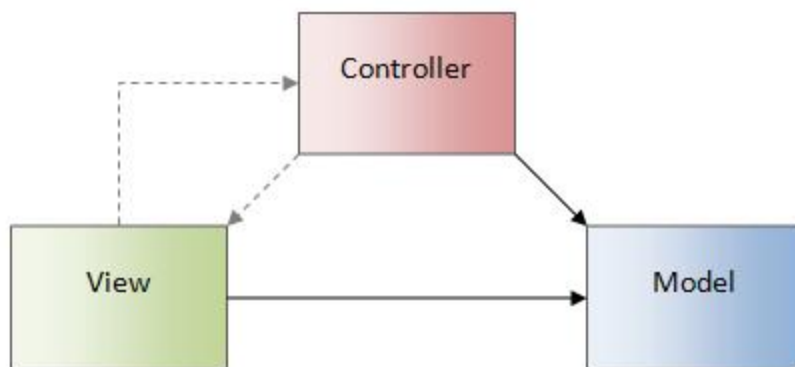
Na základě analýzy nasbíraných dat bylo ihned zřetelné, že společnost plánuje rozšiřování svých aktivit na internetu formou nových projektů anebo nových funkcí na stávajících projektech. Právě tyto nové projekty či funkce by měly být po dokončení zahrnuty do aktuálně používaného systému. Této zásadní potřebě nejvíce vyhovuje architektura MVC, které se využívá pro modulární vývoj aplikací. Jednoznačnou výhodou této architektury je právě modulárnost celé aplikace [15]. Lze tedy přidávat jednotlivé moduly aplikace, které reprezentují konkrétní funkčnost, aniž by došlo k zásahu do samotného jádra systému anebo byla omezena celková funkčnost aplikace. Pokud tedy v budoucnu bude přijato rozhodnutí o doprogramování nových funkcí do systému, vytvoří se pouze nové moduly, které se připojí k již existujícímu jádru a nové funkce jsou, dá se říci hotové. Nemusí se tak začínat od začátku, ale celý systém lze rozšiřovat pohotově o konkrétní požadovanou funkčnost.

Architektura MVC (Model-View-Controller) [13] poskytuje také další zásadní výhodu, kterou je rozdělení navrhované aplikace na tři základní části a to tak, aby je šlo samostatně upravovat bez dopadu změn na ostatní části aplikace.

Model tedy reprezentuje data a business logiku aplikace, jeho hlavním úkolem je provádět operace s daty. Model jsou tedy vlastně data a k tomu připojena business logika.

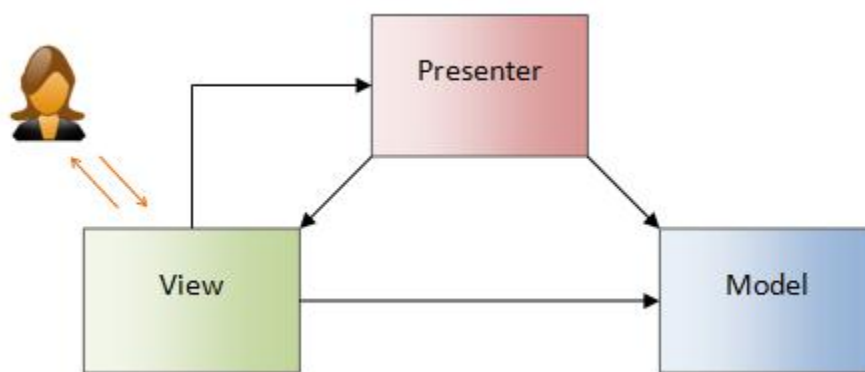
View reprezentuje uživatelské rozhraní, zobrazuje tedy výsledky, které dodal model aplikace. View je tedy zjednodušeně řečeno zobrazení modelu a dalších prvků uživatelského rozhraní.

Controller řídí tok událostí v aplikaci a obecně celou aplikační logiku, je tedy něco jako ústřední jednotka aplikace, která reaguje na podněty uživatele.



Obrázek č. 1.1: Návaznost jednotlivých součástí MVC. Převzato z [13]

Samotná rodina MVC se dále dělí na jednotlivé vzory, které se v zásadě liší drobnostmi. Asi nejznámějšími vzory je již zmíněný MVC a dále také MVP, kde je nejvíce rozdílů. Například u MVP [15], které je využito pro tvorbu nového systému, je více kontrolovaný uživatelský vstup částí view než v případě MVC vzoru. Uživatel ovládá aplikaci skrze ovládací HTML prvky jako button anebo input. Dále má view přímou vazbu na presenter a samotný presenter přímo spolupracuje s view. Nejvýznamnější změnou je, že presenter nyní obsahuje jak aplikační tak prezentační logiku aplikace, díky čemuž manipuluje s modelem a tak následně ovlivňuje view.



Obrázek č. 1.2: Vzor MVP. Převzato z [15]

Mezi další vzory MVC pak patří vzor Supervising Controller [15], kde řízení toku aplikace přebírá z větší části presenter a stará se částečně také o datovou logiku aplikace. Posledním používaným vzorem je Passive View, kde jsou možnosti view naprosto osekány. Ani zobrazení dat z modelu není zobrazeno přes view. V tomto vzoru je opět rozšířena funkčnost presenteru neboli controleru.

V rámci této architektury bude nový systém navržen tak, aby byl co nejvíce automatizován, a to jak pro programátora, tak pro kodéry anebo správce databáze. Automatizace u tohoto systému bude spočívat v načtení všech základních funkcí nutných pro validní zobrazení webové aplikace. Kromě systémových tříd se tedy automatizovaně načtou základní kaskádové styly, základní soubor s JavaScriptem a přidané knihovny. Dále budou automaticky zpracovány třídy konkrétního modulu a kaskádové styly s JavaScriptem, které jsou systémově přiřazeny k již zmíněnému modulu.

Tento systém tvorby webových aplikací, který aplikaci rozděluje na tři základní stavební bloky, přímo vybízí k dalšímu dělení aplikace. U větších projektů je žádoucí, aby byla oddělena aplikační vrstva aplikace od té prezenční, a to z důvodů efektivnosti práce. Při takovémto rozdělení aplikace může grafik anebo HTML kodér pracovat na projektu bez obav, že poškodí aplikační kód, v kterém se nevyzná a který vytvářel programátor. Pro dosažení takového oddělení se používají šablonovací systémy. Pro potřeby tvorby nového informačního systému byl použit šablonovací systém Smarty [16], který je vytvořen za

použití jazyka PHP a jedná se open source řešení. V praxi se pak šablonovací HTML [7] systém dá použít tak, že grafik anebo koder dostane od programátora seznam proměnných s popisem jejich obsahu a ten pak může pro tento obsah vytvořit konkrétní vzhled. Šablonovací systém Smarty navíc umožňuje kompilaci šablon, tedy při prvním načtení se zkompilovaná šablona uloží do složky na server a při dalším zavolání již neprobíhá její opětovný překlad, ale rovnou se zobrazí výsledek. Dále Smarty podporuje cachování, Smarty v rámci cachování neukládá šablonu se skripty, ale rovnou HTML kód celé stránky. Pomocí těchto funkcí lze znatelně zrychlit běh celé aplikace a ulevit tak serveru, který nemusí vždy provádět všechny kroky potřebné k převedení šablony.

Smarty pracuje na principu, kdy koder pracuje se šablonou, v které jsou umístěny různé značky. Tyto značky jsou dynamicky generovány pomocí PHP. Pro názornou ukázkou je připraveno textové pole s kódem níže.

```
<html>
<body>
Přihlášený uživatel: {$jmeno}
</body>
</html>
```

Z kódu je zřejmé, že tag \$jméno nebude plnit programátor v jádru systému, ale v jiném skriptu aplikace. U tohoto konkrétního systému se tak děje v modulu, který je aktuálně zpracováván. Programátor tedy například daty z databáze naplní proměnnou \$jmeno, kterou následně zpracuje Smarty svými vnitřními metodami a postupy. Příklad předání dat tagu \$jmeno je uveden pod tímto odstavcem. Skript předpokládá, že je knihovna Smarty integrovaná do systému.

```
<?php
$smarty->assign('jmeno', '$name');
$smarty->display('index.tpl');

?>
```

Dále šablonovací systém Smarty umí pracovat se základními funkcemi samotného PHP. Přímo v šabloně lze například vypsát aktuální datum a čas nikoliv pomocí příkazu `Date()` [9, 14] z PHP, ale pomocí funkcí, které poskytuje Smarty. Tyto funkce nabízejí také možnost zaznamenat POST a GET [9, 15] tzn. parametry při komunikaci klienta se serverem, takže ihned v šabloně lze reagovat na určitý podnět a zahrnout do aktuální šablony šablonu jinou například pomocí příkazu `include`.

Společně se šablonovacím systémem Smarty bude pro tvorbu vzhledu použita technologie CSS [6], tedy kaskádové styly, které konkrétně formují a graficky zobrazí konkrétní elementy, které bude obsahovat šablona šablonovacího systému.

Hlavním stavebním kamenem informačních technologií jsou v dnešní době data. Data jsou vlastně jakékoliv uložené hodnoty, které můžeme libovolně přenášet. S rozvojem informačních systémů se také vytvořila potřeba data nějakým způsobem organizovat. Pokud se data snažíme organizovat, vznikne nějaká báze dat neboli databáze. Databáze představuje organizovaný soubor dat [17] do určité struktury, čímž tvoří kompaktní celek.

Dnes již prakticky každý databázový systém disponuje SŘBD (systém šíření báze dat), který plní úlohu prostředníka mezi uživatelem a samotnou bází dat, která je uložena na disku serveru. SŘBD je vlastně taková interferenční vrstva, která má za úkol zprostředkovat dotazy, které uživatel zadá. Nejčastěji jsou dotazy psány v jazyce SQL [1], který podporuje dnes snad již každý SŘBD. Významnou roli hraje SŘBD také v bezpečnosti, jelikož data nejsou navenek jinak přístupná, než právě skrze tento systém řízení báze dat.

Pro nový systém byl zvolen databázový systém s názvem MySQL [11]. MySQL je velice rychlá a zároveň stabilní databáze, které se hojně využívá v kombinaci s programovacím jazykem PHP. I přes to, že některé funkce byly odebrány, aby se zvýšila rychlost databáze na pomalých strojích, zůstala zde jedna z hlavních funkcí a tou je správa transakcí.

Správa transakcí je jednou z klíčových funkcí každé databáze. Transakce [1, 11] je vlastně ochrana proti nekonzistentnosti dat, která může nastat například v případě poruchy

hardware. Transakce je tedy posloupnost akcí, které se do databáze promítnou jako jeden celek a tím zajišťuje, že data nebudou poškozena, nebo že do databáze se neuloží jen část dat.

Jelikož MySQL touto funkcí disponuje, je možné ji využít pro ukázkovou aplikaci. Jedinou větší nevýhodou, kromě toho, že se nehodí k vývoji robustního systému na data, je nulová podpora sekvencí. Tento nedostatek se MySQL snaží nahradit koncepcí autoinkrementace, kdy se primární klíče automaticky zvyšují. Pokud bychom však sekvence potřebovali využít, musíme si vytvořit například pomocnou tabulku a do ní vše ukládat.

Mezi snad již poslední použitou technologií patří JavaScript [4, 5] a Ajax [3]. JavaScript jako takový je v této práci použit především pro manipulaci s DOM [3] elementy a pro přesun designové náročnosti na klientovu stranu, tedy na stranu prohlížeče. Ajax bude pak pro aplikaci zajišťovat asynchronní komunikaci se serverem, tudíž je v aplikaci možné data načíst dle potřeby, bez nutnosti načítat znovu celou stránku a tím bude zajištěna vizuální dynamika celé aplikace a zároveň bude ulehčeno datovému proudu, kde prohlížeč musí parsovat celou stránku kompletně znova.

V rámci technologie JavaScript byla do aplikace integrována také JavaScript knihovna jQuery [12], která je velice populární a využívá jí široké spektrum vývojových odborníků. JQuery v této aplikaci bude sloužit především pro usnadnění práce s animacemi. Konkrétně pak pro rolování tabulek a formulářů, vyskakování pop up oken a pro další dynamické prvky aplikace. Animace a dynamické prvky dodávají každé aplikaci určitý znak modernosti, uživatel má pak dojem, že využívá moderní webovou aplikaci a nikoliv starý neohrabaný systém z 90. let.

JQuery také poskytuje vývojovým pracovníkům kompatibilitu napříč dostupnými prohlížeči, ne všude totiž probíhá zpracování JavaScriptu dle norem a standardů. Nejvíce rozdílů při zpracování JavaScriptu nalezneme klasicky u prohlížečů Internet Explorer, kde je nutné v mnoha případech skripty pro tyto prohlížeče upravit. Právě tomuto problému

jQuery předchází a poskytuje tak skvělý základ pro vývoj webových aplikací založených na JavaScriptu.

4. Vlastní práce

V této kapitole diplomové práce popíši vlastní práci na novém informačním systému. Celkový popis bude zahrnovat všechny potřebné kroky, které byly učiněny ke splnění dílčích cílů této diplomové práce a především pak pro splnění hlavního cíle práce.

4.1. Analýza

Sběr informací

Sběr informací bude prováděn přes jednotlivé konzultace především s šéfredaktorkou webového magazínu a dále pak s projektovým manažerem společnosti. Sekundárně pak proběhly rychlé konzultace s dvěma redaktory webového magazínu. Na těchto konzultacích byla v prvních fázích probírána funkčnost a nedostatky stávajícího redakčního systému a nové návrhy na úpravu stávajících funkcí, které budou zaznamenány textově anebo ve formě use case diagramu. Následně se řešila bezpečnost aplikace a její rychlost. Na poslední schůzce pak samotné požadavky na aplikaci z hlediska dalšího rozvoje projektu internetové rádia a webového magazínu.

Sběr informací od zainteresovaných osob

Na základě informací, které autor získal na konzultacích bylo možné vytvořit základní představu o fungování systému. Konkrétní redakční systém byl původně stažen z internetu jako open source systém pro tvorbu blogu na internetu. Stažený open source systém nabízel pouze omezené funkce, které s rozvojem blogu, respektive magazínu, přestaly dostačovat, a proto byl amatérsky postupně “doprogramován” a rozšiřován bez jakékoliv ucelené koncepce. Přesto základní funkce systému, tedy možnost přidávat nové články do magazínu, spravovat uživatele a reklamní kampaně, fungují a to kvalitně. Nenabízejí však uživateli přívětivé ovládání a žádné doplňující funkce, které by usnadňovaly práci.

Mezi podstatné informace také patří fakt, že nedávno proběhl re-design webové stránky magazínu, v rámci kterého se některé funkce zrušily a jiné přidaly. Tyto změny na front page byly zahrnuty do administrace pouze částečně. Nastala tak situace, kdy v menu jsou funkce, které se již nepoužívají a funkce, které se mají využívat, zde chybí a redaktoři k nim musí přistupovat přes pevný odkaz z e-mailu. Redesign front page byl také hlavním důvodem, proč se pustit do kompletní změny administrace. Na základě získaných informací lze současný redakční systém popsat následovně:

a) Uživatelé

Uživatelé jsou rozděleni do jednotlivých rolí, které mají určité oprávnění, podle kterého mohou či nemohou využívat konkrétní funkce aktuálního informačního systému. Základní rozdělení současných uživatelských rolí lze zapsat následovně:

1. Administrátor

Oprávnění je zaznamenáno v databázi číslem 1, hlavní oprávnění v rámci stávajícího systému, administrátor má nejvyšší oprávnění a může provádět také veškerou správu. Například může spravovat uživatele, editovat všechny články, zakládat nové kategorie a podobně. Hlavní funkcí administrátora je však správa článků, tedy jejich kontrola a vydávání. Funkci administrátor zastává pracovník na pozici šéfredaktora.

Mezi další funkce pak patří správa reklam, rozesílání hromadného newsletteru registrovaným uživatelům, vytváření nových uživatelů atd. Kompletní přehled funkcí aktuálního systému podle oprávnění uživatelů naleznete v tabulce č. 1.

2. Pomocný administrátor

Primární funkcí pomocného administrátora je odlehčení práce hlavnímu administrátorovi především, co se týče správy článků. Pracovník redakce se tak může stát pomocným administrátorem, kdy může kontrolovat a vydávat články, ale již nemá přístup k hlavním funkcím systému, tedy například přidávání nových uživatelů či obsluhování

reklamních kampaní, rozesílání hromadného emailu anebo správy uživatelských účtů.

3. Redaktor

Pracovník redakce v rámci svého přístupu do systému může psát články, využívat upload souborů pro potřeby tvorby článku a editovat již napsané články. Editovat lze pouze vlastní články autora, takže do cizích článků nemůže zasahovat, může si je však zobrazit.

Mezi další funkce, které může redaktor využít patří například vkládání multimediálního obsahu do článku např. videa z internetu, fotografie či písničky, které si pak čtenář může prohlédnout či pustit. Tím možnosti redaktora v současném systému končí.

4. Moderátor

Moderátor se do systému přihlašuje pro potřeby vysílání svého pořadu na internetovém rádiu. Před začátkem svého pořadu se moderátor přihlásí do systému a změní status “právě vysílá”, ve kterém je standardně umístěn nápis JUKEBOX. Dále by měl mít moderátor přístup do editace stránky svého pořadu, tato funkce však nikdy nebyla dotažena dokonce. Moderátor z funkcí systému vidí naprosté minimum a editace nefunguje. Jediný pořad, který lze upravovat je hitparáda.

5. Redaktor / Moderátor

Tato uživatelská role v sobě spojuje, co se týče funkcí, dvě předchozí a to z důvodu, že pracovník redakce může být zároveň moderátorem. Moderátor by si měl také psát články propagující pořad a proto je potřeba, aby mohl využívat funkci vložit článek a upload souborů do galerie či audio banky.

b) Hlavní funkce systému

Mezi hlavní funkce celého systému patří publikační funkce a management obsahu, tedy jeho vytvoření, kontrola a následné zveřejnění. Tuto základní funkci zajišťují následující funkčnosti:

1. Vložení článku

Skrze tuto funkčnost může redaktor či uživatel vložit do systému nový článek. Pro přidání nového článku je potřeba vyplnit následující pole:

- Obrázek o rozměrech 300x300 px ve formátu JPG
- Nadpis článku
- Tagy článku (klíčová slova vystihující obsah)
- Zařadit článek do odpovídající sekce
- Perex článku
- Obsah článku (samotný text článku)

Následně je článek uložen do databáze a čeká na schválení šéfredaktorem či pomocným administrátorem. Z nadpisu článku je automaticky vygenerován SEO odkaz, který celý nadpis převede na malá písmena a mezery mezi slovy nahradí pomlčkou. Výsledky odkaz pak vypadá například takto www.neco.cz/clanek-1-se-opravdu-povedl/

2. Nahrát soubor

Skrze tuto funkčnost je možné přidat do archivu soubor či obrázek, který je potřebný k tvorbě obsahu. Tedy například obrázek, který chce redaktor umístit přímo do textu článku.

3. Nahrát hudbu

Uživatelé také mohou nahrát hudební soubor ve formátu MP3, skrze tuto funkčnost, která hudební soubor uloží a vytvoří zdrojový kód pro webový přehrávač, který se připojí ke konkrétnímu článku.

4. Editace obsahu článku

Skrze funkčnost editace článku lze upravovat textový obsah samotného článku anebo jej například zařadit do jiné rubriky. Dále lze editovat nadpis, tagy či datum vydání. Editace článku je vlastně totožná s vložením článku s tím rozdílem, že v editaci jsou formuláře již vyplněny.

5. Nastavení článku

V rámci nastavení článku je možné změnit titulní obrázek článku, změnit autora článku, připojit video k článku, připojit audio k článku či článek zařadit do rubriky release anebo top článek. Mezi další funkčnosti patří:

- a. Změna obrázku u článku
- b. Změna autora článku
- c. Připojit video
- d. Připojit audio
- e. Připojit fotografie do galerie k článku
- f. Poznámky (nefunguje)
- g. Přidání release
- h. Umístit článek do flash modulu
- i. Vyhodnotit soutěž

6. Správa článků

Tato funkčnost umožňuje výpis všech článků anebo řazení dle vybrané rubriky. U zobrazených článků se zobrazuje jednoduchá nabídka, kde lze využít možnost náhled, editace anebo smazání.

7. Rozeslání hromadného e-mailu

Funkce v aktuálním systému poměrně nová. Díky této funkčnosti lze vybrat určité články, které jsou pak rozeslány v rámci hromadného e-mailu všem registrovaným uživatelům. Uživatel pak může prostřednictvím odkazů v e-mailu navštívit stránku anebo si rozkliknout požadovaný článek.

8. Správa uživatelů

V rámci funkčnosti správa uživatelů lze přidávat do systému nové uživatele a nastavovat jim jednotlivá oprávnění. Po vytvoření nového uživatele lze uživatele dále editovat či vymazat. Samotní uživatelé si mohou do systému nahrát vlastní fotku, změnit si email anebo přidat krátký popis o sobě samém.

9. Správa rubrik

Správa rubrik nabízí možnost spravovat jednotlivé rubriky, které již byly vytvořeny anebo zakládat rubriky nové. Vše pomocí velice jednoduchého formuláře, takže nehrozí, že by uživatel mohl něco špatně nastavit anebo zkazit. Tato funkce podléhá pouze nejvyššímu oprávnění.

10. Reklamní systém

Vlastní reklamní systém nabízí pouze základní možnosti pro práci s reklamními bannery, které se na webových stránkách objevují. Systém aktuálně umožňuje přidat nový banner a nastavit mu potřebné parametry, nelze však reklamní kampaň zpět editovat ani přenastavit.

c) Vedlejší funkce systému

1. Schválit články

Funkčnost, která si je velice podobná s funkcí správa článků. Rozdíl je pouze v tom, že v tomto konkrétním případě se zde zobrazují pouze články, které ještě nebyly schváleny nikým z administrátorů.

2. Schválit novinky

Skrze tuto funkčnost lze schválit a vydat novinku (krátký text informující například o technické údržbě), která se zobrazí na webové stránce.

3. Ankety

Systém pro správu a vytváření anketních otázek, které se zobrazují na webovém magazínu. Návštěvníci stránky pak mohou do vydané ankety hlasovat a vyjádřit tak svůj názor na konkrétní skutečnost. Tato funkčnost aktuálně podporuje vytvoření nové ankety, zveřejnění či nezveřejnění ankety, její smazání a přehled výsledků pro jednotlivé ankety.

4. Public systém

Pozůstatek z původní a staré webové verze front endu webového magazínu. Tato funkce v novém systému nebude zahrnuta, a proto bude odstraněna.

5. Komentáře

Funkčnost pro správu komentářů, které k článkům a příspěvkům píšou čtenáři a návštěvníci webového magazínu. Nikdy nebyla údajně naprogramována. S touto funkcí se do nového systému počítá a proto byla navržena kompletně znovu.

6. Info systém

Pozůstatek funkčnosti ze starého front endu, která již nefunguje. Tato funkce bude v novém systému odstraněna.

7. Download

Funkčnost ze starého front endu, která již nefunguje. Funkce obsluhovala sekci download na starém webovém designu. Požadavek je, aby tato funkce byla v pozměněném provedení integrována do nového systému.

8. Párty list

Funkčnost skrze kterou je možné editovat a upravovat seznam hudebních akcí na webových stránkách. Aktuálně lze pouze přidat novou akci, ale nefunguje zpětná editace či smazání vytvořené akce.

9. Neopravené články

Funkčnost, která byla vytvořena na základě požadavků korektur článků. Korektorovi jsou zobrazeny neopravené články, které může opravit a po opravě je článek nastaven na stav opraveno. Článek se tak již nezobrazuje mezi neopravenými články.

10. Vložit novinku

Vložení krátké novinky, která se zobrazuje na front endu webu. Jedná o rychlé zprávy, typu technické výluky či zveřejnění jmen výherců v soutěži.

11. Moderuje

Slouží k změně statusu moderátor. Vždy, když začne nějaký pořad, je moderátor povinen se přepsat v tomto status, aby o pořadu byli informováni návštěvníci webových stránek.

12. Vložit program

Zbývá funkčnost ze starého webového magazínu. Tato funkčnost bude v novém systému integrována a naprogramována od základu.

13. Editovat program

Tato funkčnost měla nabízet editaci vloženého programu, která však nefunguje. V novém systému tato funkčnost bude integrována a funkční.

14. Pořady

Funkčnost sloužící pro správu jednotlivých pořadů a jejich vytváření či editaci. Opět se jedná o nedokončenou funkčnost, která nefunguje korektně a není tak využívána.

d) Databáze

Samotná databáze a především její popis v rámci aktuálního informačního systému by zabral desítky stránek. Analýza byla tedy provedena nad hlavními tabulkami, které jsou nutné pro fungování základních funkcí aktuálního systému. V systému je aktuálně přes 15 000 článků, statisíce obrázků, přes 35 000 připojených videoklipů k jednotlivým článkům,

především z YouTube a více jak stotisíc tagů a spoustu dalších dat jako je 13 000 registrovaných uživatelů anebo komentáře k článkům, kterých jsou stovky až tisíce.

Hlavní tabulky databáze:

- Clanky
- Autori
- Rubriky
- Registrace
- Video
- Foto
- Tagy
- Novinky
- Live Show
- Program
- Audio
- Audio_soubory
- Flash
- Flash_new
- Historylist
- Hype_banner
- Komentáře
- Partylist
- Photoreports

Dále se v databázi vyskytuje celá řada složitě identifikovatelných tabulek, které nejspíše pocházejí z předchozí verze front endu a nebyly smazány z databáze při přechodu na novou podobu a funkčnost front endu.

Osobní názory pracovníků

Dotazování pracovníků v rámci jejich subjektivního vnímání systému nebylo provedeno klasickou formou dotazníku, a to z důvodů příliš úzkého rozpětí, který je dotazník schopen pokrýt. Samozřejmě výhodou dotazníku jsou strukturovaná data, ale pro účel této práce se autor rozhodl zvolit metodu brainstormingu a snažil se pracovníky donutit, aby nejprve vyjádřili názor na tyto podněty:

- 1) Komplexní informační systém
- 2) Grafické rozhraní
- 3) Efektivita práce
- 4) Rychlost systému
- 5) Bezpečnost systému
- 6) Rozhodování
- 7) Motivační faktor

Podněty byly vybrány na základě předchozích konzultací, kdy je na prvních třech sezeních sami zmiňovali konkrétní pracovníci. Tímto se podařilo odhalit nejpalčivější problémy, jelikož pracovníci na problémy, které je opravdu v aktuálním systému nejvíce trápily, reagovali ihned. U pracovníků s nižšími právy oprávnění se nejvíce a nejrychleji reagovalo na tyto podněty:

- Grafické rozhraní
- Efektivita práce
- Motivační faktor

Na ostatní faktory nebylo buď vůbec reagováno anebo se volně přešlo do diskuze nad konkrétními problémy.

U šéfredaktora a manažera systému šlo o tyto podněty:

- Komplexní informační systém
- Bezpečnost
- Rozhodování
- Efektivita práce

Grafické rozhraní

Současné grafické rozhraní systému je vnímáno jako strohé, nebarevné a nezáživné. Rovněž bylo uvedeno, jako požadavek na nový systém, celkově lepší grafické zpracování nového informačního systému. Dále je současné grafické prostředí okoukané a několik let staré, což má za následek demotivující efekt na pracovníky. Grafické prostředí je také málo interaktivní a celkově působí ponurým dojmem.

Efektivita práce

Současní pracovníci, kteří se věnují psaní článků anebo správou celého webu, nemají k dispozici efektivní nástroje pro jejich práci. Základní postupy systému sice fungují, ale nenabízí nic navíc. Několikrát bylo v rámci interview zmíněno, že systém nenabízí žádné informace a informační zdroje pro práci redaktorů anebo alespoň základní statistické a analytické funkce.

Motivační faktor

Celý tým bere svou práci v aktuálním informačním systému jako nutnou rutinu a po několika letech, kdy se prakticky nic v informačním systému nezměnilo, působí celý systém jako demotivující faktor, který ubírá na kreativě.

Komplexní informační systém

Pro pracovníky s vyšším oprávněním v systému je hlavní problém komplexnost aktuálního systému. Mnoho funkcí nefunguje správně anebo fungují jen částečně, chybí zpětná editace vloženého obsahu a aktuální systém rozhodně nepokrývá aktivity společnosti kompletně. Chybí především větší provázanost s internetovým rádiem, kde není možné na webových stránkách přidávat informace o nových pořadech, či nahrávat nové reprízy jednotlivých dílů anebo zde umístit playlist pořadu.

Pro projektového manažera aktuální systém kromě správy reklamních bannerových pozic nepřináší vůbec nic.

Bezpečnost

Systém byl několikrát vystaven několika útokům a dvakrát se podařilo obranu prolomit a zneužít uživatelská data či poškodit samotnou společnost rozesláním hromadného e-mailu s pohoršující zprávou.

Rozhodování

Pracovníci s vyšším oprávněním, jako je například produkt manažer anebo šéfredaktor, nemají k dispozici žádné informace o provozu stránky anebo samotného systému. K dispozici jsou pouze tři základní informace a těmi jsou celkový počet článků, celkový počet uživatelů a celkový počet rubrik.

Osobní názor autora

Při zkoumání aktuálního informačního systému musím vesměs souhlasit se vším, co je systému pracovníky vytýkáno. Snad jediným faktorem, který se zdá být naprosto v pořádku, je rychlost aktuálního systému. Jelikož se jedná o velice jednoduchý systém, jak z programátorského hlediska, tak z designerského, systém reaguje rychle, svižně a ukládání či mazání dat anebo fotek netrvá nikterak dlouho. Jako hlavní problém vidí autor

bezpečnost systému, hesla pracovníků v databázi nejsou kódována, využívá se přihlášení přes cookies, kde je v cookies uloženo přímo id pracovníka a systém obsahuje další bezpečnostní chyby, které byly podrobeny analýze.

Dalším velkým neduhem je nefunkční zaslání zapomenutého hesla a také omezení přihlášení na tři pokusy. Po třech neplatných pokusech je účet zablokován a není možné se přihlásit, je nutné kontaktovat administrátora. Při konzultacích mi bylo řešeno, že měsíčně jsou zaslány manažerovi projektu minimálně 4 požadavky na odblokování účtu z důvodů špatného zadání hesla pracovníkem.

Grafické rozhraní systému je opravdu zastaralé a kombinuje v sobě pouze tři barvy a to šedou, bílou a černou. Na dnešní poměry se jedná o naprosto nevyhovující design takovéto aplikace.

Zhodnocení nasbíraných dat

Při závěrečném zhodnocení nasbíraných dat došel autor k závěru, že dat je tolik a potřeby společnosti jsou značně rozsáhlé, že není v jeho možnostech zpracovat kompletně nový systém v rámci této diplomové práce, a to především z důvodu velké obsáhlosti výsledného díla, ale také pro jeho složitosti. Z tohoto důvodu se autor v této diplomové práci zaměří především na hlavní funkce aktuálního systému, které pracovníci nejvíce využívají. Dále pak především na potřeby budoucího rozvoje společnosti a jejich aktivit a stejně tak na požadavky, které byly jasně definovány.

Uživatelské požadavky na nový systém

- Základním uživatelským požadavkem je atraktivní a moderní design celé aplikace. Jelikož autor není zkušený designer a ani mu k tomu nebyly dány vlohly, byla společnosti doporučena vhodná šablona, kterou společnost zakoupila a poskytla autorovi pro práci na novém systému.

- Mezi další uživatelské požadavky patří zachování rozmístění ovládacích elementů a alespoň částečné zachování základních pracovních postupů.
- Budoucí modifikovatelnost a rozvoj systému patří také do naprosto zásadních požadavků. Aby bylo vyhověno tomuto požadavku, byla použita modulární architektura a celý nový systém je navrhován na základě jádra a jednotlivých modulů.
- Zachování aktuálních dat je třetím a posledním uživatelským požadavkem, systém tedy bude v závěru integrován na současnou a upravenou databázi.

Identifikace slabin systému

- Zastaralý systém nevyhovující aktuálním potřebám
- Složitý rozvoj a správa systému z developerského pohledu
- Nedostatečné funkce systému
- Bezpečnostní riziko je vysoké
- Nekomplexní systém
- Částečně špatně navržená databáze
- Nespolehlivý
- Špatná integrita systému

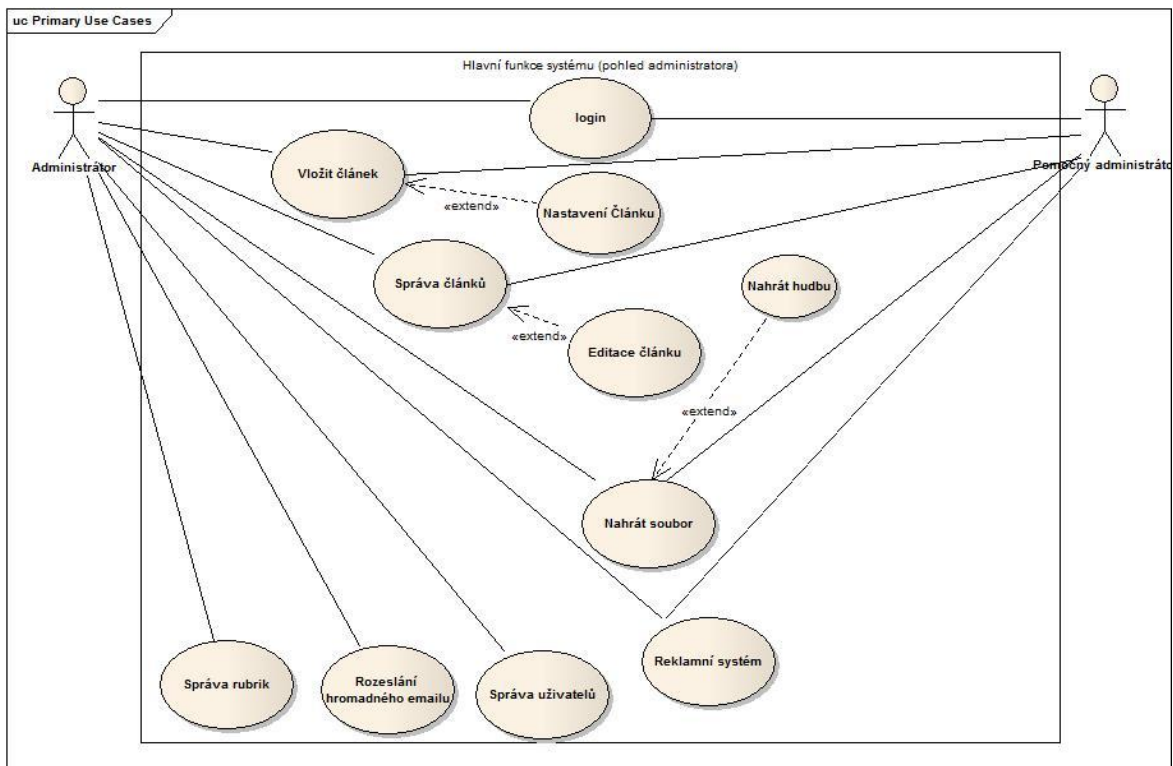
Základní pracovní postupy pracovníků

- 1) Přidání nového článku
 - a) Brouzdá po internetových zdrojích a hledá informace. Pokud narazí na zajímavé informace, začne psát článek. Najde na internetu titulní obrázek k článku, který na počítači upraví do čtverce.
 - b) Přihlásí se do systému
 - c) Vybere funkci vložit článek
 - d) Přidá připravený obrázek a tvoří obsah

- e) Pokud je potřeba připojí video, audio nebo další fotky
- f) Kontrola
- g) Publikace

- 2) Schválení článku
 - a) Přihlásí se do systému
 - b) otevře článek
 - c) čte
 - d) upravuje
 - e) kontroluje
 - f) publikuje

Use Case diagramy hlavních funkcí



Obrázek č. 2.1: Use Case diagram funkcí v rámci pohledu administrátora

Další use case diagram, který hlavní funkce znázorňuje z pohledu role redaktora a redaktora / moderátora je přiložen v příloze č. 2 na konci této diplomové práce.

Oprávnění uživatelů

	Administrátor	Pomocný Administrátor	Redaktor	Moderátor	Redaktor / Moderátor
Vložení článku	•	•	•		•
Nahrát soubor	•	•	•		•
Nahrát hudbu	•	•	•		•
Editace článku	•	•	•		•
Nastavení článku	•	•	•		•
Správa článků	•	•			
Rozeslání e-mailu	•				
Správa uživatelů	•	•			
Správa rubrik	•				
Reklamní systém	•				
Schválit články	•	•			
Schválit novinky	•	•			
Aneekty	•	•			
Partylist	•	•			
Vložit novinku	•	•	•		•
Moderuje	•	•		•	•
Vložit program	•	•		•	•
Editovat program	•	•		•	•
Pořady	•	•		•	•

Tabulka č. 1: Rozložení uživatelských oprávnění ve stávajícím systému.

Úprava databáze

Po provedení analýzy byly některé tabulky upraveny a rozšířeny o data, které vyžaduje návrh nového systému. Jedná se především o data informativního charakteru, například rozšíření informací o uživatelích systému, včetně jejich kontaktních informací. Dále byly upraveny jednotlivé propojení a asociace mezi tabulkami. Mezi poslední změny,

které pro účely této práce byly potřebné, patří zavedení referenční integrity nad novými i stávajícími tabulkami. Tím bude zlepšena celková integrita databáze a bude zabráněno výpadkům dat a tedy celkovému nepropojení tabulek. Konkrétní změny jsou uvedeny v další kapitole Navrhované řešení.

Z jednotlivých schůzek s pracovníky a z autorovy osobní analýzy je jasné, že nový systém společnost potřebuje jako prioritní faktor, ke svému dalšímu rozvoji. Aktuální systém není schopen plnit na 100% ani základní funkce, natož aby se k němu přidávaly nové. Proto bylo rozhodnuto, že nový systém bude vytvořen kompletně znovu od základů. Data budou ponechány a zálohovány pro případ, že by se při modifikaci databáze poškodily anebo jinak znehodnotily. Do nového systému budou zahrnuty především hlavní funkce systému a to buď ve stávající podobě anebo modifikovány. Funkce, které nový systém nebude využívat a budou odstraněny.

Navrhované řešení, které bylo manažerem společnosti schváleno, vychází z této analýzy a je schopno pojmout všechny požadavky společnosti i v budoucnu.

Bezpečnostní analýza

V rámci bezpečnostní analýzy byla provedena analýza databázové tabulky *log_login* a *log_pokusy*. Právě tyto tabulky v sobě obsahují údaje o tom kolikrát se kdo pokoušel do systému připojit. Tabulky kromě těchto údajů obsahují také uživatelský vstup, tedy jaké bylo zadáno uživatelské jméno a heslo uživatele.

Analýzou těchto dat bylo zjištěno, že systém byl dříve mnohokrát napaden pomocí techniky sql injection, kdy je na uživatelském vstupu systému podsunut spustitelný kód napsaný v jazyce sql.

			id ▲	login	pokus
<input type="checkbox"/>			1	lucie.sommerova@hiphopstage.cz	1
<input type="checkbox"/>			2	raek@hiphopstage.cz	3
<input type="checkbox"/>			3	worlos001@dmail.com	1
<input type="checkbox"/>			4	mia@hiphopstage.cz	1
<input type="checkbox"/>			5	admin	4
<input type="checkbox"/>			6	x	1
<input type="checkbox"/>			7	2CZnak	2
<input type="checkbox"/>			8	AC	1
<input type="checkbox"/>			9	m	3
<input type="checkbox"/>			10	worlos001@gmail.com	2
<input type="checkbox"/>			11	Patty	2
<input type="checkbox"/>			12	worlos001@seznam.cz	2
<input type="checkbox"/>			13	blcegro@hiphopstage.cz	1
<input type="checkbox"/>			14	life five	2
<input type="checkbox"/>			16	l	1
<input type="checkbox"/>			17	'	1
<input type="checkbox"/>			18	!SIWCRTESTINPUT000000!E!	4
<input type="checkbox"/>			19	!SIWCRTESTINPUT000000!E!' aNd '7'='7	2
<input type="checkbox"/>			20	!SIWCRTESTINPUT000000!E!' aNd '7'='2	2
<input type="checkbox"/>			21	1 aNd 7=7	3
<input type="checkbox"/>			22	1 aNd 7=2	2
<input type="checkbox"/>			23	1' aNd '7'='7	2
<input type="checkbox"/>			24	1' aNd '7'='2	2
<input type="checkbox"/>			25	99999999 oR 7=7	2
<input type="checkbox"/>			26	99999999 oR 7=2	2
<input type="checkbox"/>			27	99999999' oR '7'='7	2
<input type="checkbox"/>			28	99999999' oR '7'='2	2
<input type="checkbox"/>			29	99999999'] * user[@role='admin	2
<input type="checkbox"/>			30	99999999	2
<input type="checkbox"/>			31	' oR '%='	1

Obrázek č. 3.1: Ukázka tabulky login_pokusy

4.2. Navrhované řešení

Jádro nového informačního systému a zvolená architektura

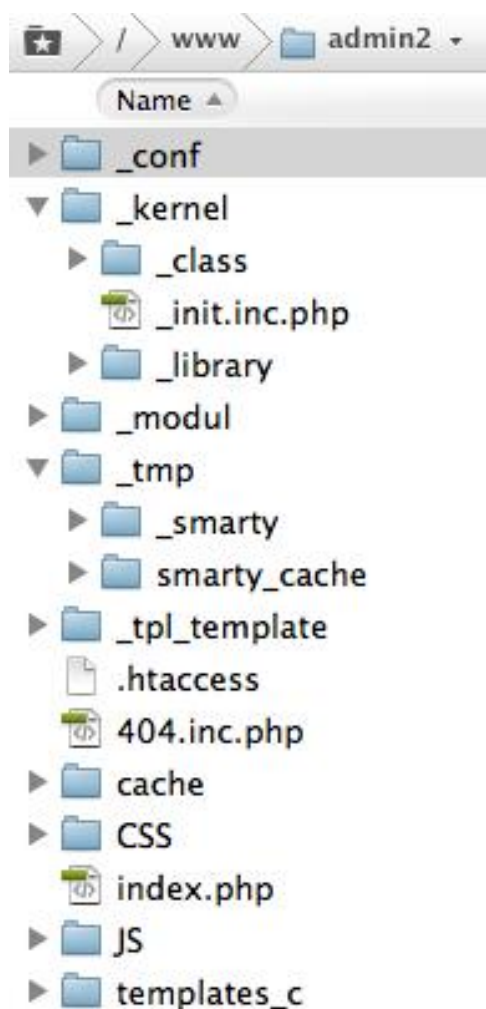
Po zhodnocení všech uživatelských požadavků a všech slabin systému je zřejmé, že nový informační systém musí být navržen znovu a to od základů. Proto byl zvolen objektový přístup při programování a autor se bude držet architektury MVC [13] respektive MVP [15], na základě které bude navrženo jádro systému. Nový informační systém bude z developerského hlediska rozdělen na několik podsystémů, které spolu vzájemně budou spolupracovat. Základním stavebním kamenem je jádro systému, které bylo nazváno KERNEL. Jádro je srdcem celého systému a z velké části také automatizuje práci programátora a kodéra. Po skončení práce jádra bude výsledkem zpracování interní business logiky aplikace, zpracování modulů, zpracování šablonovacího systému a kompletní HTML [7] výstup.

Samotné jádro systému obsahuje pole v PHP, které v sobě ponese základní informace o aktuálním dění v systému. Jádro zároveň obsluhuje všechny třídy, které informační systém bude obsahovat a které při jeho běhu nebo běhu konkrétního modulu budou informačním systémem využívány.

Právě jádro systému bude zajišťovat flexibilitu a variabilitu výsledného systému. V případě budoucí potřeby zavedení nových funkcí, budou vyvíjeny a navrhovány pouze konkrétní funkce, které se připojí k jádru. Zvolená architektura v kombinaci s funkcemi jádra zajistí připojení nového modulu automatizovaně, programátor pouze vytvoří v systémové struktuře novou složku pro modul, kam nahraje kompletní třídy, kaskádové styly anebo javascript soubory. Jedná se tedy o značně rozšiřitelný systém, který může být modifikován za běhu, jelikož vývoj nových modulů neovlivní funkčnost samotného systému, díky oddělené funkčnosti jádra nového systému a jednotlivých podaplikací, což jsou v tomto konkrétním případě již zmíněné moduly. Tento přístup umožňuje, aby na systému mohlo pracovat více programátorů, jelikož si práci mohou v budoucnu rozložit na jednotlivé aplikace, které budou vytvářet. Pro usnadnění programátorské práce budou mít k

dispozici několik statických tříd, které mohou využívat při programování modulu anebo například již zrealizované připojení k databázi.

V rámci této diplomové práce bude realizováno především stabilní a otestované jádro systému a základní požadované funkčnosti nového systému. Další funkce a rozvoj systému se autor zavázal dokončit po ukončení této diplomové práce. A to z důvodu velkého záběru nových funkcí i jejich složitosti.



Obrázek č. 3.2: Systémové rozložení nového informačního systému

V kořenovém adresáři aplikace `www` budou umístěny další složky a soubory, které systém bude využívat.

Ve složce `_conf` je umístěn konfigurační soubor nového systému, který obsahuje základní nastavení.

Ve složce `_kernel` je umístěno jádro systému, konkrétně pak inicializační soubor a v podsložkách `_class` a `_library` jsou jednotlivé třídy jádra systému či podpůrné a rozšiřující knihovny.

Ve složce `_modul` jsou umístěny složky jednotlivých modulů, každá složka modulu obsahuje všechny potřebné třídy, které zajišťují funkčnost modulu. Dále obsahuje také podsložky, do kterých jsou umístěny rozšiřující soubory kaskádových stylů anebo javascriptové rozšíření modulu.

Složka `_tmp` slouží jako mezipaměť pro jádro systému a pro systémové soubory. Jádro systému `kernel` zde může vytvořit dočasné soubory, do kterých uloží obsah, který je potřeba držet v paměti. Dále jsou zde složky pro šablonovací systém Smarty, který tyto složky využívá pro uložení kompilovaných anebo cachovaných šablon. Složka má nastaveny práva pro čtení a zápis, serverově jde o stav `7777`, takzvaný `chmod`.

Do složky `_tpl_template` jsou ukládány všechny html šablony, které jsou používány pro modul anebo samotné jádro systému. Jedná se o konkrétní html podobu webové stránky, kterou aplikace bude zobrazovat uživateli. V šablonách však nejsou uloženy data, jelikož je v rámci nového systému navzájem oddělena aplikační logika a prezentační logika.

Složky *CSS* a *JS* slouží k uložení kaskádových stylů, které se starají o rozložení grafického rozhraní a také k ukládání javascriptových souborů, které jsou následně systémem načteny a prohlížečem zpracovány.

Navrhované bezpečnostní kroky

Z dostupných informací je zřejmé, že se několikrát útočníkům podařilo prolomit zabezpečení stávajícího informačního systému a následně došlo k zneužití dat. V návrhu nového informačního systému byly tedy zahrnuty základní bezpečnostní kroky, které zvýší komplexně ochranu a sníží potencionální bezpečnostní rizika. Kroky pro zlepšení bezpečností byly podniknuty v několika rovinách.

V první řadě byla všechna aktuální hesla v databázi převedena do kódované podoby a to skrze funkci jazyka PHP `md5`, která znaky hesla převede na hash řetězec. Hashovací funkce má tu výhodu, že heslo je převedeno na řetězec znaků téměř ihned, ale zpětné rozkódování je značně komplikované a zabere poměrně hodně času. Tím je tedy znemožněno útočníkovi při úspěšném průlomu do databáze získat například heslo od administrátorského účtu, útočník získá “pouze” hash otisk hesla.

V rámci samotné aplikace byly podniknuty další bezpečnostní kroky, a to ošetřením uživatelského vstupu. Uživatel zadává uživatelské jméno a heslo, na základě kterého je vytvořen dotaz do databáze. Právě zde hrozí bezpečnostní riziko průlomu do databáze a to takzvanou metodou `sql injection`. Jedná se o techniku, kdy útočník místo uživatelského hesla zadá kus kódu, obvykle kus kódu `sql` jazyka. K tomu aby nedošlo ke spuštění takového `sql` kódu byla využita funkce databáze `MySQL`, kterou je `mysql_real_escape_string` [1, 11]. Z bezpečnostní analýzy je jednoznačně vidět, že k tomuto dochází a proto je proti `sql injection` aplikace chráněná.

Posledním krokem, realizovaném v rámci zajištění bezpečnosti, je funkce na automatické blokování účtů. Pokud bude zadáno špatně heslo více jak 3x, bude účet automaticky zablokován. Pro odblokování účtu bude nutné kontaktovat administrátora

informačního systému. Tím by mělo dojít k zamezení hacknutí hesla pomocí automatického generátoru, který je velice efektivní v případě slabých hesel typu 12345 a podobné.

Navrhované uživatelské prostředí

Na základě požadavku na nové uživatelské rozhraní byl zvolen po vzájemné dohodě autora této práce a manažerem společnosti následující postup. Autor týmu pracovníků doporučil několik profesionálních template, které mají demoverzi. Po vzájemné dohodě tým zvolil template s názvem Metronic, kterou autor této diplomové práce využije při návrhu nového systému.

Návrh nového uživatelského prostředí vychází z vybrané template a jsou do něj zahrnuty také další uživatelské požadavky, jako je například zachování pracovního postupu anebo zachování podobnosti, co se týče rozmístění nových elementů. Bude kladen důraz především na jednoduchost a přehlednost nového grafického rozhraní, které však bude působit moderním dojmem a budou do něj zakomponovány dynamické prvky. Celé nové prostředí bude zpracováno pomocí procentuálního rozložení a při změně rozměru prohlížeče anebo využití zabudované funkce pro rozšíření plochy se bude uživatelské prostředí z části přizpůsobovat tomuto novému rozměru.

Navrhované základní moduly a upravení pracovního postupu

Do nového informačního systému byly nejprve zahrnuty základní moduly, které odpovídají hlavním funkcím systému, jež byly stanoveny a definovány v předchozí analýze. Tyto moduly budou navrženy tak, aby komplexně zajistily chod systému a požadovanou funkčnost.

Vložení článku

Vložení článku je základní funkcí systému, a proto bude zpracováno jako první. Pro vkládání článku bude zachováno stávající rozložení formulářových prvků, v rámci kterých

budou provedeny drobné úpravy především ke zlepšení přehlednosti.

Zároveň však zůstane zachován aktuální pracovní postup, kde se multimediální soubory připojují skrze možnost nastavení článku.

Nahrání souboru na server

Tato funkce byla zachována ve stejném stavu jako je nyní a bylo provedeno pouze rozšíření funkčnosti, aby bylo možné v rámci jednoho uploadu nahrát na server více souborů najednou. Nyní se musí více souborů nahrávat po jednom, což značně snižuje efektivitu.

Nahrání hudby na server

Tato funkce byla také zachována v současném stavu funkčnosti, jelikož je potřeba ke každému hudebnímu souboru vyplnit také dodatečné informace. Proto zde není vhodné využití hromadného uploadu souborů.

Editovat obsah článku

V rámci editace obsahu článku byla tato funkce pouze zasazena do nového designu aplikace, kdy je zde uživatelský požadavek na zachování stejného postupu. V rámci této funkce je požadováno, aby byl obsah článku v textové formě a obsah článku v multimediální formě oddělen. Multimediální obsah článku lze tedy spravovat jako doposud, přes nastavení článku.

Nastavení článku

U této funkčnosti v rámci nového informačního systému proběhne její zpřehlednění a zasazení do nového designu aplikace. Po funkční stránce nebude zásadně modifikována.

Základní správa uživatelů

Tato funkčnost bude přepracována znovu od počátku. Aktuální řešení není dostačující a nefunguje korektně. Správa uživatelů bude přehlednější a bude umožňovat vytvoření nového uživatele, zlepšení editace uživatelského profilu a zlepšenou správu jednotlivých uživatelů.

Správa rubrik

Jelikož je na webovém magazínu nastaven mod-rewrite, je potřeba při vytvoření nové rubriky manuálně editovat soubor .htaccess a novou rubriku do něj přidat. Z tohoto důvodu zůstane tato funkčnost totožná, jelikož její rozvoj z popsaného důvodu není možný.

Reklamní systém

Funkčnost reklamního systému bude zachována a modifikována pro potřeby přidání nových funkcí, které budou do této funkčnosti doprogramovány. Jedná se především o možnost zpětné správy již přidané kampaně, archivace reklamní kampaně a možnosti využívat reklamní bannery od reklamních agentur.

Navrhovaná úprava databáze

V rámci databázových změn proběhly editace tabulek autori a clanky. Tabulka autori byla rozšířena o doplňující informace o autorovi a následně k ní byla vytvořena nová tabulka, které se jmenuje autori_detail. Tato nová tabulka ukládá detailní informace o autorovi pro vnitřní využití informačního systému, jedná se například o kontaktní informace. V tabulce články byl odstraněn jeden sloupec, který se aktuálně v systému jak na front endu tak back endu nevyužíval.

Pro splnění uživatelského požadavku ohledně funkčnosti plánování reklamních kampaní byly vytvořeny dvě nové tabulky, a to konkrétně klient a kampaň. Stávající tabulka hype_banner byla modifikována a přidružena k dvěma novým tabulkám. Tím bude

zachována funkčnost systému pro zobrazování reklam na front endu. Pomocí těchto tabulek, v kterých budou uloženy informace o klientovi a jeho kampaní, dojde k vytvoření datového základu pro přepracování funkčnosti vkládání reklam. Nová funkčnost bude přehlednější a lépe spravovatelná, což bylo také cílem této změny.

V rámci realizace dvou nových funkcí, kterými jsou *Týmový chat* a *Úkoly*, bylo potřeba vytvořit dvě nové tabulky, kterými jsou `team_chat` a `task_manager`. Právě do těchto tabulek se budou ukládat data, která budou generovat dvě nové funkce.

V poslední řadě bylo odstraněno velké množství tabulek, které nebyly již řadu let využívány a zbytečně znehledňovaly databázi. Data v nich uložená byla zálohována z důvodu, že by došlo k narušení databáze anebo nějaké skryté funkčnosti, které si autor nevšiml. Nad vybranými tabulkami bude zavedena referenční integrita [1, 11], která zajistí celistvost databáze a pomocí kaskádových funkcí databáze budou data závislá na sekundárním klíči mazána či upravována tak, jak bude docházet k úpravě či mazání primárních dat. Toto řešení rovněž ulehčí práci programátorovi, jelikož se jedná o interní proces databáze.

4.3. Návrh uživatelského rozhraní

Návrh nového uživatelského rozhraní vycházel z jasného uživatelského požadavku a to, aby se toho změnilo co nejméně. Mezi požadavky byly následující položky, které se autor snažil co nejvíce dodržet:

1. Zachování rozložení vzhledu aplikace
2. Zachování rozložení jednotlivých elementů či jejich minimální změna
3. Využití moderní šablony

Jelikož autor není designér, byla z jeho strany doporučena vyhovující HTML šablona, která vyhovuje zejména třetímu bodu. Tato šablona bude autorem přepracována a pozměněna pro potřeby vzhledu nového administračního systému.

V rámci dat získaných z analýzy, která proběhla na začátku projektu, bylo vytvořeno několik základních wireframů, které reprezentují základní rozložení nového vzhledu celé aplikace a zároveň také znázornění rozložení ovládacích prvků a formulářů v hlavních funkcích aplikace.

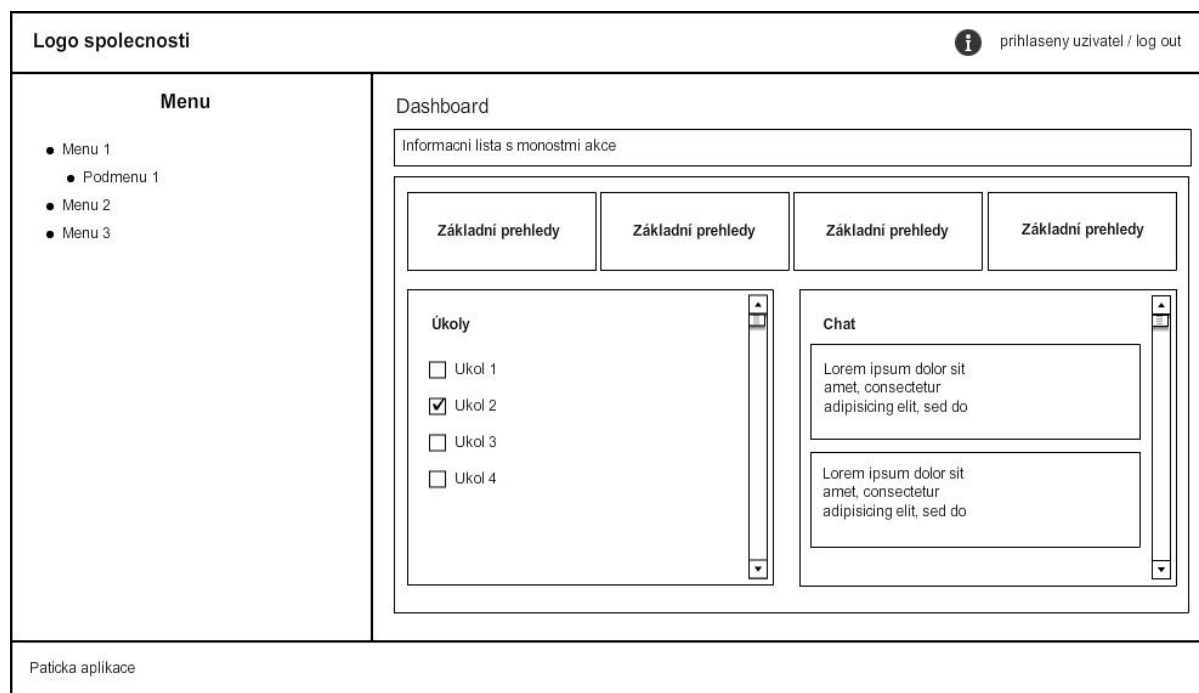
Dashboard

Asi největší změnou v rámci uživatelského rozhraní bude zakomponování dashboardu. Dashboard bude sloužit jako hlavní stránka aplikace, kdy se po přihlášení uživateli zobrazí vždy tato stránka. Na dashboardu bude několik základních ukazatelů. Konkrétně se pak jedná o tyto ukazatele:

- Počet nevydaných článků v systému
- Počet hlášených a nevyřízených chyb
- Čtenost článků od začátku aktuálního měsíce
- Akce přidáné do pártyleftu

Pod těmito ukazateli následují dvě větší okna, která jsou umístěna vedle sebe. Levé okno obsahuje malý úkolníček, který nabízí základní možnost přidání nového úkolu a odškrtnutí úkolu jako splněný. Pravé okno v sobě obsahuje základní redakční chat, kde si pracovníci mohou zanechávat krátké textové zprávy.

Základní wireframe aplikace



Obrázek č. 5.1: Náhled hlavního modulu Dashboard

Pro jednu z hlavních funkcí celého informačního systému, kterou je vkládání článku, bylo vytvořeno pozměněné uživatelské rozhraní. Samotné obsahové zpracování článku zůstalo prakticky beze změny, ale po desénové stránce bylo změněno prakticky vše.

Wireframe vložení článku

Logo společnosti i přihlaseny uzivatel / log out

Menu

- Menu 1
 - Podmenu 1
- Menu 2
- Menu 3

Vložení clanku

Informacni lista s monostmi akce Akce ▾

Nadpis

Tagy

Rubrika ▾

B / U Font ▾ -abe-

Obsah

Vložit clanek

Paticka aplikace

Obrázek č. 5.2: Návrh GUI pro funkci Vložit článek

4.4. Návrh jádra systému

Jádro nového informačního systému bude, jak již bylo v této práci zmíněno, srdcem celého systému. Kromě zabezpečení základní funkčnosti informačního systému, bude také poskytovat vysokou automatizovanost vnitřních procesů. Skrze zvolenou architekturu bude veškerý provoz na webové stránce technicky přeměrován na soubor index.php, který tak bude vždy spuštěn. Aby takovéto přeměrování bylo možné, bylo nutně přenastavit fyzický server. Konkrétní nové nastavení bylo realizováno skrze soubor .htaccess, který se nachází v kořenovém adresáři aplikace a serveru hostingu. Konkrétní kód, který zajišťuje přeměrování po technické stránce, je uveden pod tímto odstavcem.

```
RewriteEngine On

RewriteCond %{REQUEST_URI} !\.[[:alnum:]]+$
RewriteRule ^(.+[^/])$ %{REQUEST_URI}/ [R=301,L]

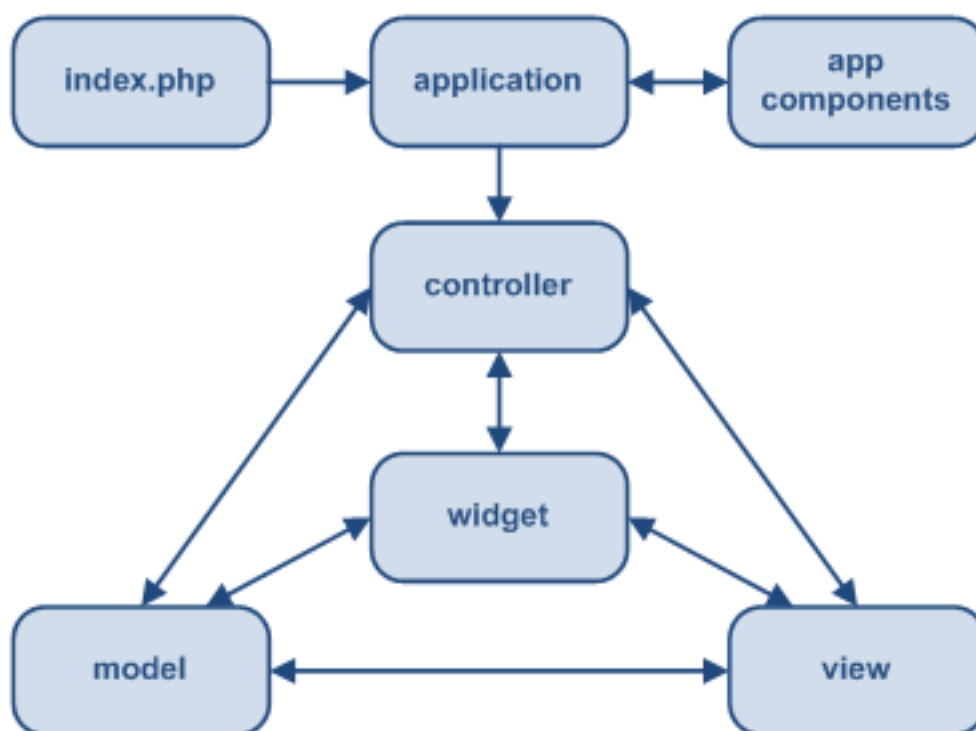
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^([^\(\.]+)$ index.php?p=$1 [QSA]

RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^([^\(\.]+)$ index.php?p=$1 [QSA]

RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^([^\(\.]+.*) index.php?p=$1 [QSA]

AddCharset utf-8 .js .php .html
```

Přeměrování provozu webové stránky zajistí, že se vždy spouští soubor index.php, který zároveň inicializuje a nastavuje jádro systému. Pomocí příkazu RewriteEngine On je spuštěn tento mód na serveru a pomocí příkazů Cond a Rule jsou vytvořeny jednotlivá pravidla. Jedná se o přeměrování provozu, jak při zadání klasické url adresy, tak také url cesty ke konkrétnímu souboru. Skrze parametry -d a !-d budou přeměrovány také existující i neexistující soubory a složky. O informování uživatele, zda požadovaná stránka nebo soubor existuje či neexistuje, se bude starat samotné jádro systému.



Obrázek č. 6: Logika funkčnosti jádra systému

V rámci souboru `index.php`, dojde k inicializaci a načtení konfiguračního souboru systému jádra a také dojde k spuštění autoloaderu. V posledním kroku pak dojde k inicializaci kontroleru systémového jádra.

Konfigurační soubor, který je umístěn ve složce `conf`, je načten pomocí příkazu jazyka PHP `require_once`. Konfigurační soubor obsahuje základní nastavení celého jádra včetně informací pro spojení s databází systému a obsahuje následující kód:

```

/*****
****
deklarace základního nastavení pomocí konstant

****
****/
define ( '__WEB__', 'http://www.hiphopstage.cz/admin_kernel/ ');
define ( '__DSEP__', '/' );
define ( '__KERNEL__', '_kernel' );
define ( '__DIRLIB__', '_kernel/_library' );
define ( '__DIRCLS__', '_kernel/_class' );
define ( '__DIRMOD__', '_modul' );
define ( '__DIRTPL__', '_tpl' );
define ( '__LANG__', 'cs' );
define ( '__START_PAGE__', 'dashboard' );
define ( '__KERNEL_TEMP__', 'hhsadmin' );
define ( '__KERNEL_LIBARY__', serialize ( array (
    'kernel_js_lib'
)));

define ( '__APP_ROZSIRENI__', serialize ( array (
    'rozsireni'
)));

define('__KODOVANI__', 'utf-8');

define ( '__NAZEV_COOKIE__', 'cookie' );
define('__DOBA_AUTOMATICKEHO_ODHLASENI__', 9999999);
define('__JEN_JEDNA_POVOLENA_INSTANCE_PRIHLASENI__',
true);
define('__DEBUG__', false);

/*****
****
* db config
*
****
****/

define ( '__DB_HOST__', 'mysql.satoya.cz' );
define ( '__DB_USER__', 'hiphop' );
define ( '__DB_PASS__', '*****' );
define ( '__DB_NAME__', '*****' );
define ( '__DB_CHARSET__', 'utf8' );

```

Po načtení a zpracování konfiguračního souboru je načtena třída `autoloader.inc.php`, která zajistí načtení všech systémových tříd a také tříd konkrétního volaného modulu. Všechny tyto třídy jsou vloženy do paměti serveru a čekají na zavolání, které provede programátor novou instancí třídy. K tomuto uložení tříd do paměti serveru jsou využity vestavěné funkce jazyka php, kterým se říká SPL funkce. Konkrétně se takovému řešení říká SPL autoload a využívá se ve všech frameworkcích jako je třeba Nette anebo Zend Framework.

Kromě načtení všech systémových tříd dojde také k zahrnutí používaných knihoven a rozšíření. Pokud chce programátor používat určitou knihovnu PHP, například Nette Debugger, stačí pouze třídu s kódem knihovny nakopírovat do složky `_kernel/_library` a systém při každém spuštění tuto knihovnu zpřístupní pro využití. Autoload systém je hlavním stavebním kamenem celé automatizace nového systému, jelikož se postará o samotnou konfiguraci jádra a zahrnutí všech potřebných tříd a rozšíření.

Po načtení autoloaderu přichází na řadu v souboru `index.php` poslední krok a tím je inicializace a zavolání systémového kontroleru. Systémový kontroler se nazývá `_init.inc.php` a obsahuje v sobě vytvoření instance třídy `autoloader`, která, jak již bylo řečeno, provede kompletní načtení systémových souborů. Dále kontroler obsahuje proměnou `$JADRO`, skrze kterou je vytvořena nová instance hlavní třídy jádra systému.

Ukázka souboru index.php

```
/* **** */
* načtení konfiguračního souboru
* **** */
require_once ('_conf/_conf.inc.php');

/* **** */
* načtení třídy s autoloadem
* **** */

require_once (__DIRCLS__ . __DSEP__ . '_autoloader.inc.php');

/* **** */
* načtení kontroleru
* **** */
require_once (__KERNEL__ . __DSEP__ . '_init.inc.php');
```

Ukázka systémového kontroleru

```
/* **** */
* spuštění autoloadu tříd
* **** */
new autoloader ();
/* **** */
* spuštění kernelu – jádro systému
* **** */
$JADRO = new KERNEL ();
```

Hlavní systémová třída `KERNEL` je tedy v tomto okamžiku spuštěna a startuje se systém. Prvním krokem, který systémová třída realizuje, je vytvoření interního systémového pole pod názvem `$KERNELARR`. Toto pole je nastaveno jako `protected static`, tudíž k němu lze přistupovat pouze v rámci třídy `KERNEL` anebo v rámci tříd, které z této hlavní systémové třídy dědí.

Jelikož tento systém využívá návrhového vzoru faktory, jsou všechny další systémové komponenty samostatnými třídami, které jsou však zděděny z hlavní systémové

třídy. Pro vnořené systémové pole se najde využití, jak v rámci interního systému, tak při tvorbě nových modulů.

V rámci kontroleru systémové třídy `KERNEL` dále probíhá postupné volání všech metod, kterými třída disponuje a také načtení dalších rozšíření systému. Jednotlivé systémové metody naplní interní pole základními informacemi a připraví například databázové spojení. Jedná se vlastně o přípravu základů pro budoucí další malé továrny, tedy nové funkce systému.

Jádro systému v aktuální verzi obsahuje následující metody, které se starají o chod systému jako celku. Disponuje tedy následujícími metodami:

- **startKernel**

Tato metoda je vykonána jako první a aktuálně obsahuje jediný příkaz `ob_start`, což v jazyce PHP zapíná výstupní buffer. V případě, že je tento buffer zapnut, není žádnému výstupu skriptu umožněno odeslání, místo toho je uložen do vnitřního bufferu jazyka PHP, respektive serveru, na kterém PHP běží.

- **sestavURL**

Tato metoda je volána ihned po vykonání předcházející metody. Jedná se o parser, který rozparsuje zadanou url, kterou návštěvník anebo uživatel zadá do prohlížeče. Jak již v této práci bylo zmíněno, vždy dojde k přesměrování provozu webové stránky na soubor `index.php`, na který je také přesměrována konkrétní url a její parametry. URL a parametry lze získat ze serveru pomocí příkazu `$_GET` a samotná url je pak v této metodě rozložena na jednotlivé parametry podle lomítek, které obsahuje. Skript takto dostane url například ve tvaru `www.neco.cz/rubrika/novinky`, která je následně rozložena na jednotlivé parametry. V tomto případě pak konkrétně na parametr *rubrika* a parametr *novinky*. Tyto

parametry a další informace, jako je například příchozí adresa, jsou uloženy do interního pole KERNELARR a jsou tak programátorovi kdykoliv přístupné, pokud pracuje v systémové třídě anebo ve zděděné třídě ze systémové třídy. Z bezpečnostního hlediska je každý vstup před načtením samotným jádrem zpracován funkcí pro kontrolu nebezpečných znaků. Tato funkce převede celý řetězec na malá písmena a odstraní znaky, jako jsou hvězdičky či uvozovky. Tím je tedy do jádra systému puštěn takto ošetřený a bezpečný řetězec pro další zpracování.

- **startDB**

Voláním této metody systémové třídy dojde k navázání spojení s databází. Z konfiguračního souboru aplikace jsou načteny konstanty, které obsahují informace potřebné pro navázání spojení s databází a spojení je navázáno.

- **Odpoved**

Tato metoda vyvolá novou instanci třídy systému s názvem kernel_odpoved.php, která provede kontrolu existence volaného modulu a zároveň také proběhne volání metody pro ověření kontroly relace přihlášení. Pokud tedy volaný modul existuje, ale uživatel není přihlášený, dojde k přesměrování na stránku s loginem. Naopak, pokud modul neexistuje, je zobrazeno chybové hlášení. Zároveň je existující modul a informace k němu patřící uloženy do integrovaného pole KERNELARR. Spolu s těmito informacemi jádro ukládá také stav systému, který je při správné funkčnosti nastaven na hodnotu 200.

- **load_rozsireni**

Aby bylo možné do nového informačního systému po programátorské stránce přidávat nové metody, které budou používány mimo systémové jádro, je zde zavedena metoda load_rozsireni. Při volání této metody, dojde k načtení konstanty z konfiguračního souboru, která obsahuje názvy daných tříd. Názvy rozšiřujících

tříd jsou uloženy v této konstantě jako serializované pole a po načtení tohoto pole z konstanty je zahájen průchod tohoto pole a v případě, že obsahuje hodnoty, jsou tyto hodnoty zpracovány. Tedy dojde k vytvoření nové instance rozšiřující třídy.

- **Model**

V rámci této metody dojde k ověření stavu systému, který je ověřován na základě dat uložených v interním poli jádra systému. Pokud je vše v pořádku a v systémovém poli je uložen stav 200, dojde k vytvoření nové instance třídy model, které jsou předány další parametry, jako je název konkrétního modelu. Tyto parametry používá pak metoda třídy model k načtení a inicializaci konkrétního modelu.

Posloupnost volání metod a tříd systémového jádra

```
$this->startKERNEL ();           // KROK č. 1
$this->sestavURL ();             // KROK č. 2
$this->startDB ();              // KROK č. 3
$this->odpoved();               // KROK č. 4
$this->load_rozsireni();        // KROK č. 5
self::$sablonu = new template(); // KROK č. 6
$this->model();                 // KROK č. 7
self::$sablonu->zobraz();       // KROK č. 8
```

Mimo standardní posloupnost volání systémových metod jsou pro samotný chod celého systému důležité také kroky č. 6 a 8. Tyto činnosti ve dvou krocích ovládají a nastavují šablonovací systém Smarty. V pomocné třídě s názvem *functionkernel.class.php* je umístěna třída s názvem *templateEngine*, která je děděna od hlavní třídy šablonovacího systému, která se jmenuje Smarty. V rámci kontroleru třídy *templateEngine* probíhá konfigurační nastavení šablonovacího systému. Jedná se především o nastavení složek pro

ukládání komprimovaných a zpracovaných šablon, ale také o možnosti cachování šablon a dalšího chování šablonovacího systému. Pro aktuální potřeby administračního systému nebude cachování šablon využito.

```
class templateEngine extends Smarty {  
  
    public function __construct(){  
        parent::__construct();  
        $this->template_dir = './_tpl_template/';  
        $this->compile_dir = './_tmp/_smarty/templates_c/';  
        $this->cache_dir = './_tmp/smarty_cache/';  
        $this->caching = 0;  
    }  
}
```

Právě při šestém kroku základní posloupnosti systémového jádra je vytvořena nová instance třídy `template`, která ve svém kontroleru vytvoří novou inicializaci třídy `templateEngine`, jež byla popsána výše. Tím je tedy šablonovací systém nastaven a čeká na přidělenou práci, respektive na data k zobrazení. Architektura celé aplikace a logika šablonovacího systému si vyžaduje, aby byl engine šablonovacího systému inicializován, dříve než samotný modul aplikace. A to z toho důvodu, že v modelu modulu programátor potřebuje pracovat se šablonovacím systémem, respektive s jeho pamětí, do které jsou uložena data, která se mají vypsat po zpracování konkrétní šablony.

Dále dojde v posledním kroku č. 8 k volání metody `zobraz`. Tato metoda se nachází v třídě `template.class.php` a obstarává několik zásadních věcí v rámci zobrazovacího systému celé aplikace. Hlavní výhodou této metody je další stupeň automatizace, který přináší. Při jejím volání dojde k načtení hlavních souborů, které obsahují kaskádové styly a javascript a dále pak k načtení sekundárních souborů modulu, které taktéž obsahují kaskádové styly a javascript. Jedná se o plně automatizovaný proces, který načte všechny definované soubory ze složek CSS a JS a to samé u modulu, kde jsou tyto složky uloženy ve složce `media`. Dále zde dojde ke zpracování dat z paměti

šablonovacího systému a jejich vytištění do šablony, které je provedeno příkazem šablonovacího systému display.

Tímto posledním krokem je ukončena činnost systému a výsledkem je zobrazená webová stránka se všemi náležitostmi a daty, jak byla naprogramována. Dá se tedy říci, že jádro systému se stará o základní šablonu systému, například o hlavičku, menu a patičku a prostřední obsah je vytvářen pomocí jednotlivých modulů aplikace.

4.5. Konkrétní implementace

Po vytvoření jádra systému a hlavních modulů, jako je login anebo dashboard, došlo k implementaci nové aplikace na současný server, který společnost hostuje u poskytovatele hostingových a housingových služeb. Aby mohl být nový informační systém implementován do provozu, bylo potřeba učinit následující kroky:

- 1) Upravit databázovou strukturu
- 2) Nahrát systém na server do root adresáře
- 3) Nastavit systém
- 4) Upravit a zpřesnit databázové dotazy
- 5) Doprogramovat následné funkce nového systému na konkrétních datech

Úprava databázové struktury proběhla v rámci kapitoly Navržené řešení. Mezi hlavní změny patří již zmíněné převedení hesel uživatelů do hashovaného otisku, které lze jen velmi těžko dekodovat. Skrze tuto změnu proběhla také změna na straně nové aplikace, konkrétně pak v rámci modulu login, který nyní vložené heslo převede také na hash otisk a ten je následně porovnán s otiskem uloženým v databázi. Byly také odstraněny tabulky, které neměly dalšího využití a nad vybranými tabulkami byla zavedena referenční integrita. Musela být také rozšířena tabulka autori, jelikož nový informační systém z bezpečnostních důvodů ukládá do databáze ip adresu, ze které se uživatel přihlásil

Upload nového informačního systému na server proběhl značně rychle, vytvořená struktura složek byla jako celek vložena do nově vytvořené složky v root adresáři serveru. Systém byl nakopírovat do složky *admin2* a to z důvodů zanechání provozu starého řešení informačního systému po dobu kompletního dodělání všech nových funkcí a integrování nových funkcí, které jsou však zatím pouze ve fázi uvažování vzhledem k současným aktivitám společnosti.

V rámci nastavení nového systému došlo ke změně konfiguračního souboru, který byl upraven následujícím způsobem. Konstanta pro definici kořenové složky systému byla upravena na novou cestu, kterou si vyžádalo nahrání systému do složky admin2. Dále byla nastavena doba automatického odhlášení na dobu 3600 sekund a proběhla také konfigurace databázového připojení, konkrétně pak zadání všech údajů potřebných pro připojení k databázi, jako je heslo, databáze, server a podobné.

Jelikož byl nový systém implementován na stávající databázi, kde jsou uložena data za šest let, postupovalo se při tvorbě databázových návrhů přesně podle návrhu nové databáze. Dotazy byly optimalizovány na konkrétní sloupce a položky, které je potřeba vybrat. Dále se používaly ve větší míře složené dotazy, které sníží jak počet celkových dotazů na databázi, tak také její zátěž. Dále byly vytvořeny dvě nové tabulky a to pro funkčnosti nového systému s názvem *Úkoly* a *Chat*. Do tabulky *úkoly* jsou ukládány jednotlivé úkoly týmu a v rámci tabulky *Chat* jsou ukládány jednotlivé zprávy, které uživatelé systému mohou napsat.

Dále bylo z důvodů zvolené architektury nutné rozložit kaskádové styly na hlavní styly, které se načítají při zpracování jádra a styly, které se načtou z jednotlivých modulů. Takovéto rozložení kaskádových stylů má výhodu v tom, že je možné vzhled jednotlivých modulů od sebe samých a také od základní šablony jádra odlišit. Tím je zajištěno, že prohlížeč zpracuje opravdu to, co je potřeba pro zobrazení konkrétní webové stránky a nikoliv další zbytečné elementy a styly k nim. Stejná situace je také v načtení javascriptových souborů, které jsou rovněž rozloženy na hlavní javascript soubor, který obsahuje hlavní funkčnost a javascript soubor, který obsahuje funkčnost jednotlivých modulů.

Na konci úspěšné implementace, kdy proběhlo v pořádku spojení s databází a kdy systém nyní vidí všechny databázové tabulky a může k nim přistupovat, bylo přistoupeno k doprogramování vedlejších modulů aplikace, které vyžadovaly práci na konkrétních datech.

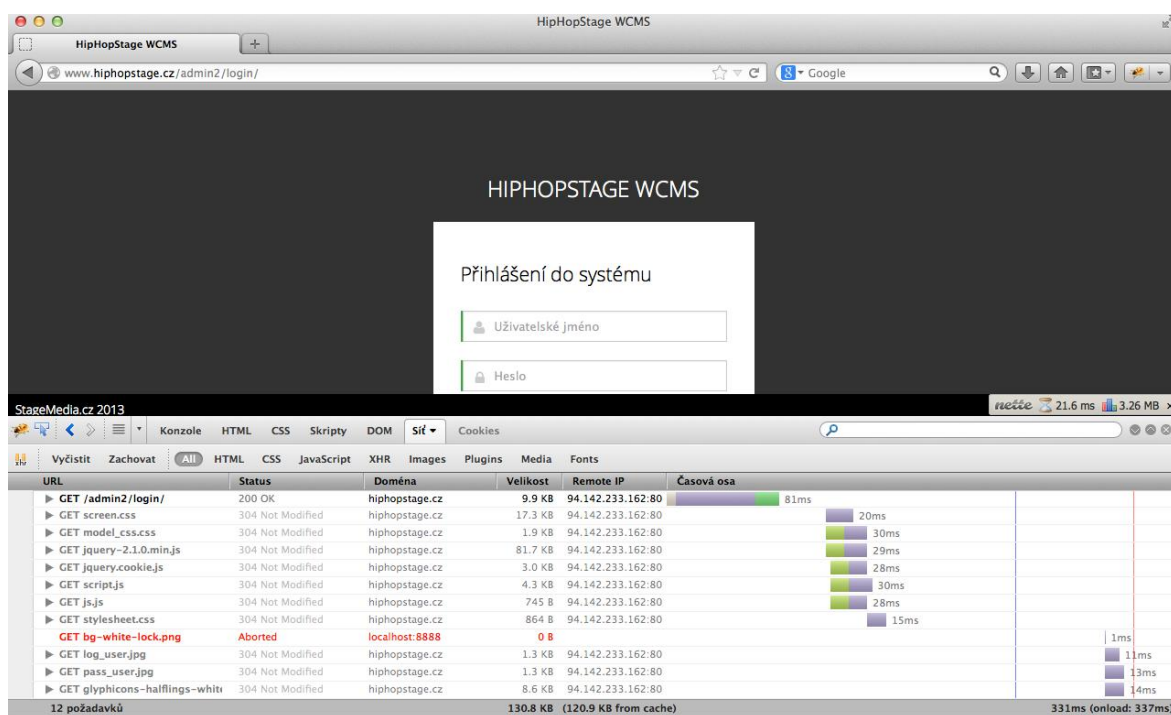
4.6. Nasazení a testování

Nasazení nového informačního systému do provozu, jak již bylo zmíněno v kapitole Implementace, probíhalo při zachování stávajícího systému. A to z toho důvodu, že do nového informačního systému budou ještě v průběhu času přidány nové funkce, které se budou realizovat na základě rozvoje webového magazínu, který má společnost aktuálně naplánovaný. Teprve až budou všechny tyto budoucí funkce vytvořeny, což záleží na době, po kterou budou tyto funkce navrhovány a zapracovány do front endu, bude starý informační systém úplně odstraněn a dojde k plnému přechodu na tento nový informační systém. Z časových důvodů musel být zvolen tento postup, protože v opačném případě by tato diplomová práce byla dopsána až za několik měsíců.

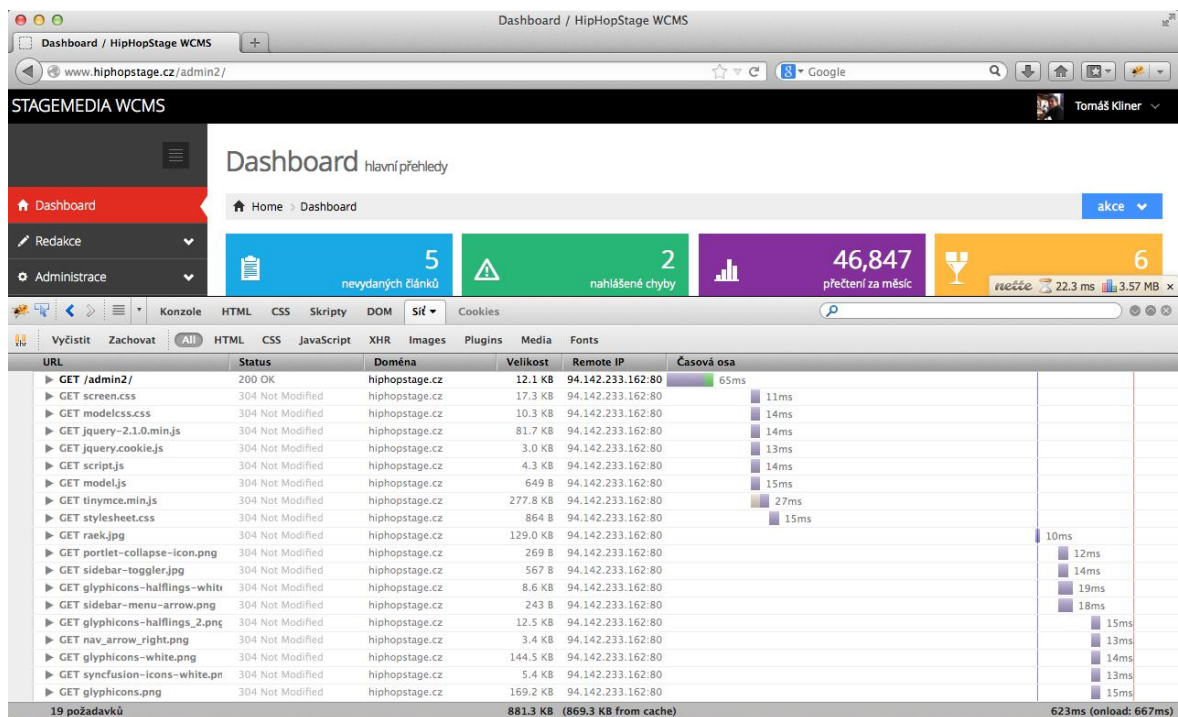
Nový informační systém byl tedy nasazen do provozu v rámci složky `admin2` a poskytuje funkčnost hlavních funkcí informačního systému tak, jak je uvedeno v kapitole Navrhované řešení. Pro nasazení systému v takovéto podobě muselo být použito několik změn a náhradních řešení. Především pak složka `archiv`, kterou využívá starý informační systém a front end aplikace, musela být ponechána na stejném místě a se stejnými daty. A to z důvodu jejího mohutného obsahu, kdy jsou ve složce statisíce obrázků a tisíce souborů s hudbou. Její kopírování či přesun by znamenal dlouhé hodiny a neúnosné náklady na správce serveru. Proto byl nový informační systém napojen na tuto složku a využívá ji pro stejné potřeby jako starý informační systém. Teprve po zrušení starého systému, bude nový systém přesunut do složky `admin`, kde bude mít přímý přístup k této složce.

Testování systému probíhalo především po celou dobu tvorby systému, kdy byl kladen velký důraz na rychlost základních funkcí systému. K testování rychlosti byl využit zabudovaný Nette Debugger, který je integrován v jádru systému. Debugger se zapíná a vypíná v konfiguračním souboru nového informačního systému. Jelikož se rychlost jádra a hlavních funkcí systému stabilně držela mezi 25ti a 40ti milisekundami, nebyl důvod provádět podrobné měření, protože se jedná o velmi dobrý výkon systému, co se týče rychlosti.

Po připojení grafického rozhraní a prvních modulů aplikace se běh aplikace na straně prohlížeče znatelně zpomalil v porovnání s načtením pouze systému. Prohlížeč musí zpracovat velké množství kaskádových stylů, obrázků a v neposlední řadě také javascript soubory včetně javascriptové knihovny jQuery anebo wisiwing editoru TinyMCE. Pozorováním bylo zjištěno, že čím je modul konkrétní funkčnosti složitější a obsahuje složitější grafiku, tím se více systém zpomalí. Výsledek takového rozdílu v modulech je ukázán pod tímto odstavcem, kde je vyfoceno síťové zatížení aplikace v modulu login, kde se nachází pouze formulář pro přihlášení a síťové zatížení aplikace při modulu dashboard, kde jsou základní statické ukazatele a další grafické elementy.



Obrázek č. 7.1: Rychlostní data pro modul aplikace Login



Obrázek č. 7.2: Rychlostní data pro modul aplikace Dashboard

Pro testování samotných hlavních funkcí systému byl využit pracovník redakce, který dostal za úkol pracovat s hlavními funkcemi systému a také zaznamenat či ohlásit jakékoliv nestandardní chování systému. Pomocí tohoto postupu byly odhaleny drobné chyby, které měly spíše designový charakter, nikoliv funkční. Funkčních chyb bylo zaznamenáno celkem 5 a tyto chyby byly posléze odstraněny.

Testování uživatelského rozhraní probíhalo na platformě Mac OS a Windows. Jelikož byl celý nový informační systém vyvíjen na platformě Mac OS, v rámci této platformy designové chyby pozorovány nebyly. V případě systému Windows se chyby designu objevily, ale vzhledem k rozsahu možnosti zařízení a verzí jednotlivých systému došlo k opravě pouze závažných designových chyb. Další ladění uživatelského rozhraní po designové stránce bude probíhat již mimo rozsah této diplomové práce, a to s přihlédnutím na časovou náročnost a na nutnost využití hacků či optimalizace pro jednotlivé prohlížeče. Nový informační systém se tak v rámci platformy Mac OS standardně zobrazuje v prohlížeči Safari, Chrome a Firefox Mozilla. Na platformě Windows se pak nový informační systém zobrazuje korektně především v prohlížeči FireFox a Chrome. Nejvíce

problémů působí vzhledu informačního systému prohlížeč Internet Explorer a to především ve verzích 7,8 a částečně také ve verzi 9. Pro tento internetový prohlížeč bude doporučena a provedena kompletní optimalizace kaskádových stylů.

Hlavní funkčnost nového uživatelského rozhraní, kterou je možnost pracovat s velikostí pracovní plochy zatažením panelu s menu a především pak přizpůsobení velikosti všech elementů vzhledem k pracovní ploše, funguje ve všech prohlížečích korektně. Nebyl zaznamenán problém se špatným obtékáním hlavních elementů a ani s jejich rozložením.

5. Závěr

Tato diplomová práce měla za cíl navrhnout a implementovat nový redakční systém do www. K tomu, aby byl hlavní cíl práce splněn, bylo nutné nejprve splnit dílčí cíle této práce. Dílčí cíle práce byly rozděleny do třech hlavních kategorií.

Prvním dílčím cílem byla analýza současného systému a analýza uživatelských potřeb. Tento dílčí cíl práce byl splněn v kapitole číslo 4. s názvem Vlastní práce a konkrétně pak v podkapitole 4.1 s názvem Analýza. Jelikož tým pracovníků zadavatele spolupracoval ochotně s autorem této práce, bylo možné analýzu udělat komplexně a do hloubky. Celková analýza byla přínosná pro autora práce, který na základě poskytnutých dat, mohl efektivně navrhnout nový informační systém.

Právě návrh nového informačního systému byl dalším dílčím úkolem práce, který z velké části plnil hlavní cíl této diplomové práce. Tento dílčí cíl práce byl splněn v podkapitole číslo 4.2 s názvem Návrh nového systému a v podkapitole č. 4.3 s názvem Návrh jádra systému. Nový informační systém, který byl v rámci této diplomové práce vytvořen, splňuje nejnovější postupy v programování webových aplikací a vytváří dostatečně kvalitní základ pro budoucí rozvoj. Samotné jádro systému poskytuje maximální možnou automatizaci pro pohled na systém z programátorského hlediska a také poskytuje kvalitní výkon jádra systému. Zvolená architektura se aktuálně využívá v komerční sféře právě pro výhody, které poskytuje.

Autor vytvořil vlastní jádro systému na principech architektury MPV, kterou si na základě načtených informací ze zdrojů a vlastní praxe vyložil po svém a zvolil postupy, které přesně nekopírují práci klasických frameworků jako je Nette anebo Zend Framework. Samotné jádro systému lze dále zlepšovat a více se přiblížit ke klasické práci framework. Pro potřeby zadání ze strany občanského sdružení a pro potřeby splnění cíle této práce, je dle mínění autora rozsah této diplomové práce dostačující. V případě rozvoje aktivit společnosti bude pouze navrženo a doprogramováno takové množství funkcí, které bude

potřeba pro tvorbu nového modulu aplikace. Jádro je navíc tak flexibilní, že jej lze využít i pro úplně jiné aplikace, či jen pro běžnou webovou stránku.

Grafické rozhraní aplikace splnilo požadavky na moderní design, který bude působit motivačně na pracovníky a zároveň bylo z co největší možné části zachováno rozložení jednotlivých elementů a pracovních postupů z původního systému. Grafické rozhraní bylo co nejvíce optimalizováno pro nejvíce populární prohlížeče současnosti, ale mezi platformami Mac OS a Windows jsou vidět značné rozdíly. Vzhledem k tomu, že Windows je daleko rozšířenější platforma než Mac OS, navrhuje autor pozdější optimalizaci pro systémy běžící na Windows a později pak pro Mac OS.

Po bezpečnostní stránce byly podniknuty zásadní kroky k tomu, aby bylo co nejvíce ztíženo proniknutí do aplikace a citlivé údaje, jako přihlašovací hesla, byly zakódovány. Těmito kroky byla posílena bezpečnost mnohonásobně a aplikace se tak nyní může řadit ke standardním WCMS aplikacím, která obsahují a podporují podobné zabezpečení. V případě, že by zadavatel chtěl mít naprostou jistotu a přehled o zabezpečení této aplikace, autor doporučuje placený bezpečnostní audit u profesionální společnosti, která se touto bezpečnostní činností zabývá. Samotné jádro systému může být zabezpečeno více ve smyslu přijímání uživatelského vstupu, ale vzhledem k tomu, že je možné uživatelský vstup zadat až po přihlášení do aplikace, je základní bezpečnostní podmínka splněna.

Posledním dílčím cílem byla implementace a otestování systém. Tento dílčí cíl diplomové práce byl splněn v podtitole č. 4.4 s názvem Konkrétní implementace a v podtitole č. 4.5 s názvem Nasazení a testování. Systém po základních testech a provozu funguje korektně a základní funkce systému fungují rovněž bez problémů. Již mimo tuto diplomovou práci budou navrženy anebo upraveny nové funkčnosti, které zadavatel do systému plánuje zavést.

Autor se snažil vývoj nového systému přiblížit spíše z teoretického hlediska, co se týče samotného návrhu a programování systému, než z hlediska praktického. Ostatně, kompletní kód aplikace je přiložen jako příloha na DVD. Ke spuštění aplikace je potřeba server, kde běží systém Apache, PHP a MySQL databáze. Následně je potřeba provést vytvoření databázové struktury aplikace, kterou lze vytvořit uploadem souboru z DVD, který je zde pod názvem create_db. Tento soubor je nutné importovat skrze aplikaci pro MySQL, která se jmenuje phpMyAdmin. Mezi poslední kroky patří konfigurace databázového připojení v konfiguračním souboru aplikace, a to na přihlašovací údaje vlastní databáze.

Seznam literatury a použitých zdrojů:

- [1] CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7.
- [2] Steven Holzner. *Mistrovství v Ajaxu*: Computer Press, 2007. ISBN 978-80-251-1850-4.
- [3] Ryan Asleson, Nathaniel T. Schutta. *Ajax, Vytváříme vysoce interaktivní webové aplikace*: Computer Press, 2006. ISBN 80-251-1285-3.
- [4] Rastislav Škultéty *JavaScript - programujeme internetové aplikace*: Computer Press, 2001. ISBN 80-7226-457-5.
- [5] Slavoj Písek *JavaScript*: Grada, 2001. ISBN 80-247-0014-X.
- [6] CROFT, Jeff, Ian LLOYD a Dan RUBIN. *Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře*. Vyd. 1. Překlad Josef Bábík. Brno: Computer Press, 2007, 409 s. ISBN 978-80-251-1705-7.
- [7] KENNEDY, O`REILLY. *HTML a XHTML kompletní průvodce*. Vyd. 1. Praha: Computer Press, 2000, 633 s. ISBN 80-722-6407-9.
- [8] HANA, Kanisová. *UML srozumitelně*. Vyd. 1. Brno: Computer Press, 2004, 157 s. ISBN 80-251-0231-9.
- [9] GUTMANS, Stig Sæther BAKKEN a Derick RETHANS. *Mistrovství v PHP 5*. Vyd. 1. Překlad Bogdan Kiszka. Brno: CP Books, 2005, 655 s. ISBN 80-251-0799-X.

Webové stránky:

- [10] IBM CORPORATION. The business value of Web 2.0 technology.: IBM's vision for tapping the collective knowledge of the extended value chain [PDF]. 2007. Dostupné z: ftp://public.dhe.ibm.com/software/lotus/pub/lotusweb/web20/10709800_Web_2.0_brochure.pdf
- [11] ORACLE CORPORATION. *MySQL* [online]. 2012 [cit. 2014-03-11]. Dostupné z: <http://www.mysql.com>
- [12] THE JQUERY FOUNDATION. JQuery [online]. 2006 [cit. 2014-02-10]. Dostupné z: www.jquery.org
- [13] BERNARD, Borek. Úvod do architektury MVC. *Zdrojak.cz* [online]. 2009 [cit. 2014-03-19]. Dostupné z: <http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>
- [14] *PHP Documentation: Function Reference* [online]. 2006 [cit. 19. 3. 2014]. Dostupné z: <http://www.php.net/manual/en/book.spl.php>
- [15] BERNARD, Borek. Prezentáční vzory z rodiny MVC. *Zdroja.cz* [online]. 2009, č. 09 [cit. 2014-03-19]. Dostupné z: <http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>
- [16] *Smarty 3 Manual*. 2013. Dostupné z: <http://www.smarty.net/docs/en>
- [17] DOLÁK, Ondřej. SYSTEMONLINE.CZ. *Big data: Nové způsoby zpracování a analýzy velkých objemů dat*. 2011. Dostupné z: <http://www.systemonline.cz/clanky/big-data.htm>

Seznam tabulek:

Tabulka č. 1: Kompletní přehled uživatelských oprávnění současného systému

Seznam obrázků:

Obrázek č. 1.1: Návaznost jednotlivých součástí MVC

Obrázek č. 1.2: Vzor MVP

Obrázek č. 2.1: Use Case diagram funkcí v rámci pohledu administrátora

Obrázek č. 3.1: Ukázka tabulky login_pokusy

Obrázek č. 3.2: Systémové rozložení nového informačního systému

Obrázek č. 4: Návrh upravené databáze

Obrázek č. 5.2: Náhled hlavního modulu Dashboard

Obrázek č. 5.2: Návrh GUI pro funkci Vložit článek

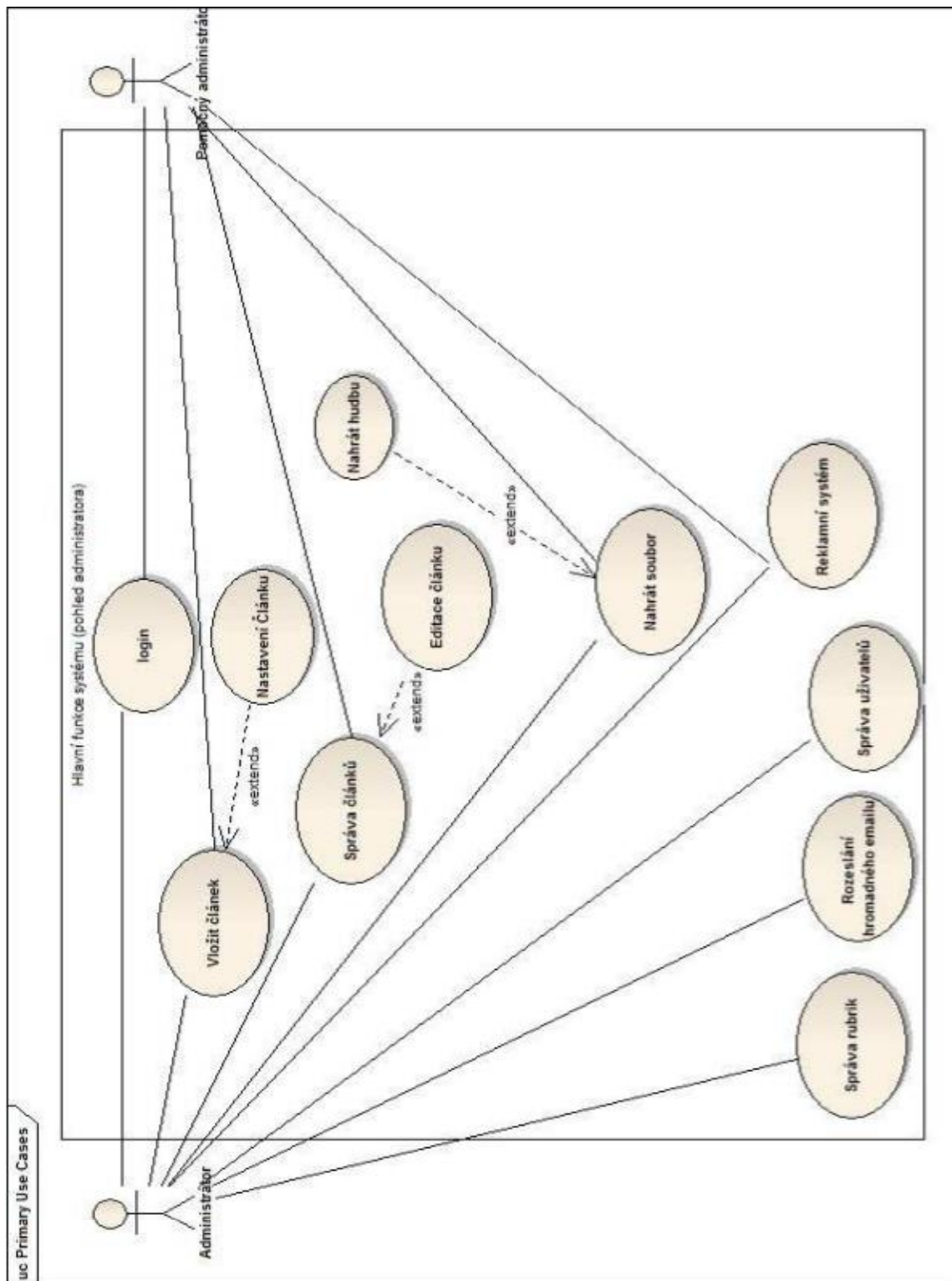
Obrázek č. 6: Logika funkčnosti jádra systému

Obrázek č. 7.1: Rychlostní data pro modul aplikace Login

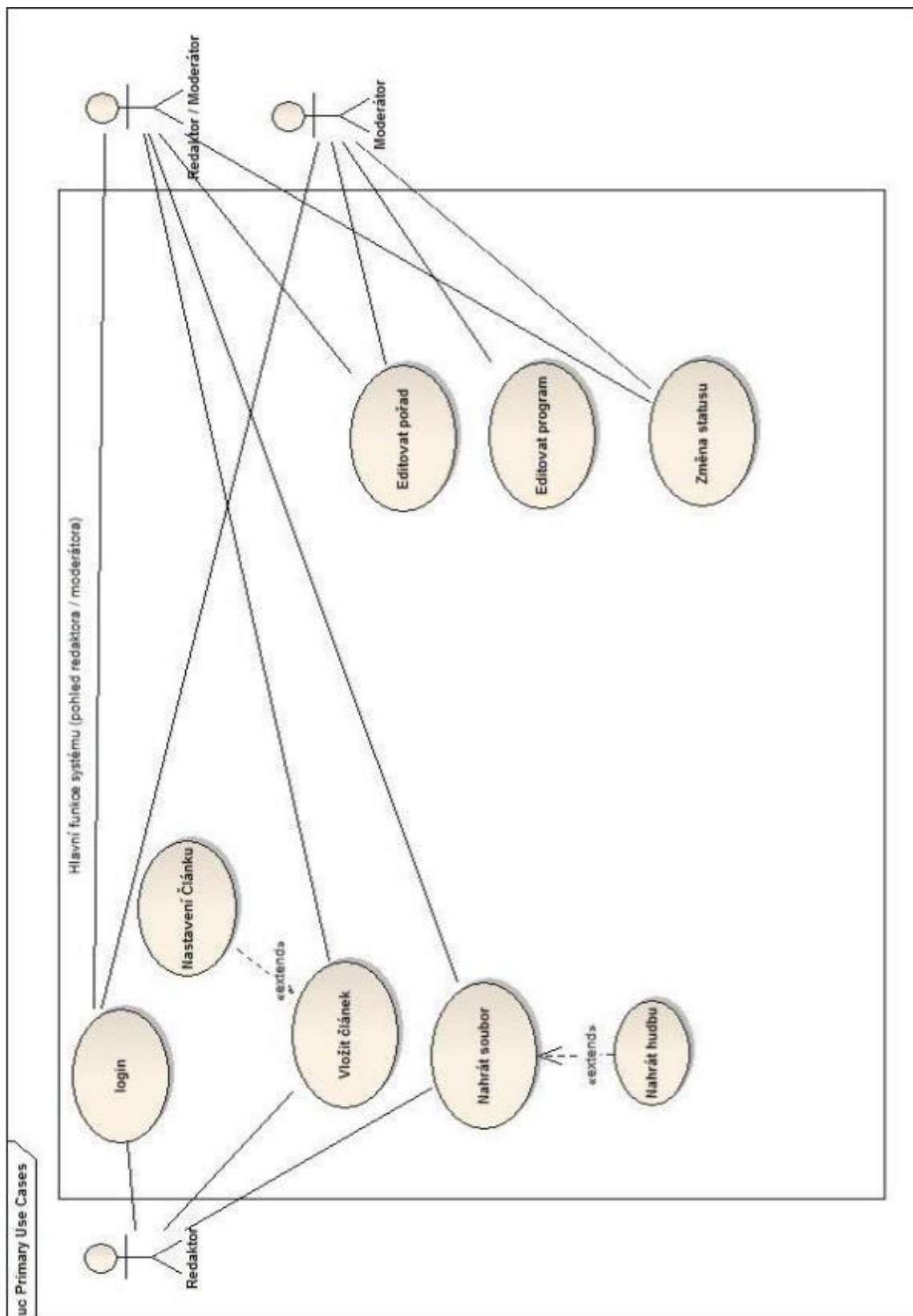
Obrázek č. 7.2: Rychlostní data pro modul aplikace Dashboard

Přílohy:

Příloha č. 1: Use Case Diagram z pohledu administrátora



Příloha č. 2: Use Case Diagram z pohledu redaktora a redaktora / moderátora



Příloha č. 4: Kompletní zdrojový kód aplikace

Celá aplikace je k této práci přiložena na DVD, které je umístěno na konci této diplomové práce.