

Titulná strana – Tomáš Ballon

TODO

Abstrakt

TODO

Pod'akovanie

TODO

Obsah

1 Úvod.....	6
1.1 Motivácia.....	6
1.2 Cieľ.....	6
1.3 Štruktúra práce.....	6
2 Analýza.....	7
2.1 Použité technológie.....	7
2.1.1 (X)HTML.....	7
2.1.2 CSS.....	7
2.1.3 JavaScript.....	7
2.1.4 Apache.....	7
2.1.5 PHP.....	8
2.1.6 Ajax.....	8
2.1.7 JQuery.....	8
2.1.8 PHP frameworky.....	8
2.1.9 MySQL.....	8
2.2 Prehľad podobného softvéru.....	9
2.2.1 Webové aplikácie.....	9
2.2.1.1 Facebook.....	9
2.2.1.2 Google+.....	10
2.2.2 Desktopové aplikácie.....	10
2.2.2.1 Jabber.....	10
2.2.2.2 Skype.....	10
2.2.2.3 ICQ.....	10
2.2.2.4 Windows Live Messenger.....	11
3 Špecifikácia systému.....	11
3.1 Cieľové skupiny.....	11
3.2 Základná architektúra systému.....	11
3.3 Používateľské role.....	12
3.3.1.1 Klasický používateľ.....	12
3.3.1.2 Superadministrátor.....	12
3.4 Databáza.....	12

3.5 Triedy.....	13
3.5.1 Knižnice.....	13
3.5.2 Modely.....	14
3.5.3 Core.....	14
3.5.3.1 Main_controller.....	15
3.5.4 Controllery.....	15
3.5.4.1 Page.....	15
3.5.4.1.1 Page.....	15
3.5.4.2 User.....	16
3.5.4.2.1 Home.....	16
3.5.4.2.2 Logged.....	16
3.5.4.2.3 People.....	17
3.5.4.2.4 Chat.....	17
3.5.4.2.5 Adminpage.....	18
3.5.4.2.6 Filemanager.....	18
3.5.4.3 Admin.....	19
3.5.4.3.1 Home.....	19
3.5.4.3.2 Logged.....	19
3.5.5 Zobrazovanie (views).....	20
3.5.6 Pluginy.....	20
3.5.7 JavaScript a ajax.....	20
4 Implementácia systému.....	22
4.1 Problémy pri implementácii.....	22
5 Záver.....	22
5.1 Zhodnotenie výslednej aplikácie.....	22
5.2 Návrhy na vylepšenie v budúcnosti.....	22
6 Zoznam použitej literatúry.....	22
7 Prílohy.....	23

1 Úvod

1.1 Motivácia

Sme v dobe plnej digitálnych technológií a informatizácie. Internet nám každým dňom zjednodušuje život. Používame ho na rôzne druhy komunikácie a to či už na súkromnú alebo pracovnú. Na internete sa nachádza obrovské množstvo komunikačných aplikácií, no ich používanosť je veľmi odlišná. Najpoužívanejšie aplikácie sú centralizované a kontrolované z jedného miesta. Nie sú ľahko prispôsobiteľné, ale sú predkonfigurované na to, aby sa páčili čo najväčšiemu množstvu ľudí a sú prepchaté reklamami. Takéto aplikácie by sa s malými úpravami (ktoré však nie sú možné) dali používať napríklad vo firemnom sektore. Veľká časť firiem zakazuje svojim zamestnancom prístup k takýmto aplikáciám, no pri tom by mohli byť veľmi prínosné pre efektivitu práce.

1.2 Cieľ

Mojim cieľom je vytvoriť webovú komunikačnú aplikáciu, ktorá bude ľahko konfigurovateľná, upraviteľná a kompatibilná s najpoužívanejšími serverovými riešeniami súčasnosti. Nie pre všetky organizácie je vhodné aby mali jej členovia prístup na internet a tým pádom by sa mohla využívať aj na intranete. Bude jednoducho upraviteľná a rozšíriteľná pomocou rôznych pluginov. Týmto spôsobom by si ju mohli používatelia upraviť tak, aby vyhovovala aj náročnejším používateľom. V neposlednom rade, môže pomôcť odtrhnúť sa od centralizovaných riešení. Aplikácia by mohla byť využívaná od jednoguchej blogovacej aplikácie až po menšiu sociálnu sieť a to aj bez využitia pluginov.

1.3 Štruktúra práce

V druhej kapitole sa budem venovať analýze technológií, medzi ktorými som sa pred programovaním rozhodoval. Zhodnotím jednotlivé technológie a rozoberiem niektoré frameworky medzi ktorými som si vyberal. Ďalej sa v tejto kapitole zameriam na prehľad podobného softvéru, ktorý aj do hĺbky rozoberiem.

V tretej kapitole sa zameriam na samotnú špecifikáciu systému, ktorý navrhujem. Dopodrobna v nej rozoberiem návrh a riešenie databázy, štruktúru systému, ktorej jadro tvorí framework a nad ním triedy rozdelené na modely a controllery.

V ďalšej kapitole sa zameriam na implementáciu systému. Rozoberiem nasadenie systému

na server, jeho základnú konfiguráciu a možnosti, ako je ho možné upraviť pomocou nastavení v hlavnej administrácii a rozšírení funkcionality pomocou pluginov.

V závere zhrniem celú prácu a postup pri tvorbe aplikácie, jej možné využitie a tak isto aj možné úpravy a vylepšenia do budúcnosti.

2 Analýza

V tejto kapitole sa zameriavam na rozoberanie samotných technológií použitých pri tvorbe aplikácií a rozboru existujúcich aplikácií s podobným zameraním a funkcionality.

2.1 Použité technológie

Pri výbere technológií som sa zameral na technológie, ktoré sú bezplatné a kompatibilné s najznámejšími serverovými riešeniami. Aplikácia by mala byť tým pádom jednoducho použiteľná aj bez väčších zásahov softvérového vybavenia servera.

2.1.1 (X)HTML

(X)HTML je skratka pre HyperText Markup Language, teda HyperTextový značkovací jazyk, ktorý slúži na vytváranie webových stránok. V aktuálnej stabilnej verzii je kompatibilný so všetkými webovými prehliadačmi v najnovších verziách.

2.1.2 CSS

CSS sú kaskádové štýly. Je to rozšírenie jazyka (X)HTML, ktoré je využívané na dizajnovanie a grafické formátovanie webových stránok. Je tak isto kompatibilné s najnovšími verziami prehliadačov, no v každom z nich sa správa o niečo inak.

2.1.3 JavaScript

JavaScript je skriptovací jazyk. Tento jazyk sa vykonáva na strane klienta, v tomto prípade najčastejšie vo webovom prehliadači. Tento jazyk sa využíva hlavne na programovanie webových aplikácií. Je kompatibilný so všetkými najpoužívanejšími webovými prehliadačmi.

2.1.4 Apache

Apache je jeden z najpoužívanejších webových serverov. Je dostupný pre všetky používané platformy, akými sú Linux, Windows a unixové operačné systémy. Spracováva HTTP požiadavky a

k nim adekvátne výstupy podľa jeho konfigurácie.

2.1.5 PHP

PHP je skriptovací programovací jazyk, ktorý sa vykonáva na strane servera. Je používaný na programovanie dynamických webových stránok a klient-server aplikácií. Sú v ňom naprogramované aj známe webové aplikácie akými sú napríklad phpBB, ktorý slúži na vytváranie diskusných fór, MediaWiki, na ktorom beží Wikipedia, alebo aj samotný Facebook. PHP je kompatibilné so serverom Apache.

2.1.6 Ajax

Slúži na vytváranie webových stránok a aplikácií, ktoré aktualizujú svoj obsah bez nutnosti znovunačítania celej stránky. Ajax je spojenie viacerých technológií a to môžu byť napríklad (X)HTML, CSS, JavaScript, PHP atď.

2.1.7 JQuery

JQuery je najpoužívanejšou JavaScriptovou knižnicou. Uľahčuje prácu s JavaScriptom a ajaxom. Umožňuje odchyťovanie udalostí, čo je jej obrovskou výhodou.

2.1.8 PHP frameworky

Framework slúži na uľahčenie práce programátorom. Frameworky môžu obsahovať knižnice, alebo aj podprogramy, ktoré pomáhajú programátorom riešiť najčastejšie problémy v danej oblasti. PHP frameworky majú v sebe najčastejšie riešené databázové spojenia, správu používateľov a veľmi často riešia aj štruktúru danej aplikácie. Najčastejšie to je MVC (Model View Controller). V modeloch sú programované funkcie, ktoré sa vykonávajú dopyty na databázu, v controlleroch je riešená príprava dát pre zobrazenie a vo viewoch sa zobrazujú spracované dáta.

TODO NAJZNÁMEJŠIE FRAMEWORKY

TODO POROVNANIE FRAMEWORKOV

TODO PREČO SOM SI VYBRAL CODEIGNITER

2.1.9 MySQL

MySQL je najznámejší databázový server. Je bezplatný a dostatočne efektívny. Je podobne ako Apache podporovaný na najpoužívanejších platformách. Práca s databázou MySQL prebieha

pomocou dotazov, ktoré sú písane v jazyku SQL (Structured Query Language).

2.2 Prehľad podobného softvéru

Pri analýze som sa venoval porovnávaniu rôznych komunikačných aplikácií. Možno ich rozdeliť do dvoch skupín a to na webové a desktopové aplikácie.

2.2.1 Webové aplikácie

Medzi webovými aplikáciami, ktoré som sa rozhodol analyzovať sú aj dve sociálne siete a to Facebook a Google+. Tieto aplikácie sú veľmi obširne a stavané pre masu ľudí. No ich nastavitelnosť nie je pre veľa organizácií dostatočná a to najmä preto, že sa tu nedajú vytvárať samostatné skupiny používateľov s rôznymi oprávneniami a nie je ich možné používať mimo internetu. Cieľom oboch sociálnych sietí je mať čo najviac používateľov. To im umožňuje zarábať na reklame a zisťovať aktuálne trendy na internete a aj pre to sú nevhodné pre nasadenie v pracovnom prostredí.

2.2.1.1 Facebook

Facebook je najväčšia a najznámejšia sociálna sieť. Väčšina jeho serverov sa nachádza na pôde USA. Facebook je navrhnutý na zdieľanie textov a multimédií medzi používateľmi. Aplikácia obsahuje vnútorný chat medzi používateľmi.

Mnohé spoločnosti zakazujú používateľom prístup na Facebook. A to z viacerých dôvodov. Najčastejšie je to kvôli tomu, že tam ľudia počas pracovného času trávajú veľa času a nevenujú sa práci. Ďalším závažným dôvodom je nebezpečenstvo, že sa cez Facebook dostanú von z firmy rôzne citlivé dáta a dokumenty. Keď sa už niečo dostane na Facebook a začne sa to zdieľať medzi používateľmi, je veľmi zložitá a takmer nemožná zastaviť šírenie takéhoto obsahu. Ďalej tu je nebezpečenstvo zneužitia dát, ktoré sú uložené v dátových centrách. Niektoré organizácie by ich pre to radšej videli pod svojou strechou.

Na Facebooku je možné vytvárať rôzne skupiny, ale členov týchto skupín nemožno odrezat' od zvyšku tejto sociálnej siete. A to by bolo pre niektoré spoločnosti potrebné a tým by sa eliminovali aj niektoré vyššie spomenuté nevýhody. Okrem vytvárania skupín existujú na Facebooku aj aplikácie. Tieto aplikácie majú k dispozícii Facebook API a môžu to byť ľubovoľné webové aplikácie (Flash, HTML5, obyčajná webová stránka, atď.). No ani pomocou nich nie je možné používateľov odrezat' od zvyšku Facebooku.

Facebook je vhodný pre reklamu, vytváranie propagačných stránok, ale nie na vnútornú komunikáciu vo firmách, alebo iných organizáciách.

2.2.1.2 Google+

Google+ je sociálna sieť, ktorá vznikla ako odpoveď Googlu na úspech Facebooku. Funkcionalitou je veľmi podobný Facebooku. Facebook a Google+ sa navzájom predbiehajú a prinášajú množstvo vylepšení, ale sú vhodné len pre súkromnú sféru a reklamné účely.

TODO ďalšie webové aplikácie

TODO tabuľka porovnaní webových aplikácií

2.2.2 Desktopové aplikácie

Najväčšou nevýhodou desktopových aplikácií je to, že sú to väčšinou len nástroje, ktoré slúžia na chatovanie. Chýba im aj permanentné ukladanie súborov a vytváranie diskusných tém. No okrem toho to zvyknú byť klienti, ktorí rozširujú, alebo prenášajú niektorú funkcionality webových aplikácií do desktopového prostredia, mimo webových prehliadačov. Aj tieto aplikácie sú vo veľkej väčšine závislé na internete a nie je ich možné využívať iba v rámci intranetu.

2.2.2.1 Jabber

Jabber je komunikačný protokol, nie samostatná desktopová aplikácia. Komunikácia je typu klient – server. Tento protokol sa nazýva Extensible Messaging and Presence Protocol (XMPP). Je otvorený a decentralizovaný, čo prináša množstvo výhod. XMPP využíva napríklad aj Google Talk.

2.2.2.2 Skype

Skype je peer – to – peer aplikácia. Služi predovšetkým na hlasovú komunikáciu medzi používateľmi. Jedna z nevýhod je to, že keď je používateľ, ktorému chcete poslať správu offline, správa ostáva iba v odosielajúcom klientovi a odošle sa iba v prípade, že sú obaja používatelia online. Skype má ďalšiu veľkú nevýhodu, ktorá ho posúva ešte nižšie ako ostatné aplikácie. A tým je to, že skype dokáže zahlcovať veľké množstvo pásma. Preto ho niektorí sieťoví provideri zakazujú.

2.2.2.3 ICQ

ICQ je podobne ako skype peer – to – peer komunikačná aplikácia. ICQ má na serveri

uložených množstvo dát o svojich používateľoch. Na rozdiel od skype sa po odoslaní správy, používateľovi, ktorý nie je online, táto správa uloží na serveri a odošle sa mu okamžite po tom ako sa dostane do stavu online. ICQ nezahluje pásmo ako skype. Ale hlavnou nevýhodou ICQ sú jeho licenčné podmienky, v ktorých sa nachádza pasáž o tom, že na všetko čo zdieľate cez ICQ strácate autorské práva.

2.2.2.4 Windows Live Messenger

Windows Live Messenger je komunikačná aplikácia od Microsoftu. Beží na komunikačnom protokole od Microsoftu, Microsoft Notification Protocol (MSNP). Disponuje veľmi podobnou funkcionalitou ako ICQ. Windows Live Messenger je plne integrovaný do Windowsu.

TODO ďalšie desktopové aplikácie

TODO tabuľka porovnaní desktopových aplikácií

3 Špecifikácia systému

Systém bude navrhnutý tak, aby bol čo najviac používateľsky prívetivý a čo najjednoduchšie upraviteľný a rozšíriteľný.

3.1 Cieľové skupiny

Cieľovými skupinami budú rôzne organizácie a používatelia, ktorí potrebujú webovú aplikáciu, ktorá umožňuje správu obsahu (CMS), realtimeový chat, alebo nástenku pre zdieľanie obsahu medzi používateľmi. No primárne bude určená pre používateľov, ktorí potrebujú aj funkcionalitu naviac, alebo upravenú funkcionalitu danej aplikácie. Aplikáciu bude vhodné využiť aj ako jednoduché fórum, či menšiu sociálnu sieť. Toto všetko bude možné bez prídavných pluginov. V aplikácii bude kladený dôraz na to, aby bola nastaviteľná a každá jej súčasť bola použiteľná samostatne.

3.2 Základná architektúra systému

Celý systém bude postavený na PHP frameworku CodeIgniter, ktorý bude komunikovať s MySQL databázou. Framework je nemeniteľná súčasť systému (ak neberieme do úvahy jeho aktualizáciu na novšie verzie). Nad týmto frameworkom sa bude nachádzať samotný systém s plnou funkcionalitou, ktorého kód by sa pri používaní a konfigurácii tak isto nemal meniť. Tento systém bude mať na starosti všetky základné funkcie a v neposlednom rade sa bude starať o inštaláciu a beh

pluginov. Nad samotným systémom sa bude nachádzať posledná vrstva už spomenutých pluginov. Tieto pluginy budú v prvom rade komunikovať so systémom, cez ktorý môžu komunikovať aj so samotným frameworkom.

TODO – obrázok

3.3 Používateľské role

V systéme budú obsiahnuté dve hlavné používateľské role, ktoré budú nemeniteľné. Bude to klasický používateľ a superadministrátor. Systém bude podľa týchto rolí rozdelený na dve časti a to frontend a backend, do ktorých sa bude prihlasovať cez samostatné prihlasovacie formuláre.

3.3.1.1 Klasický používateľ

Klasický používateľ sa bude môcť do systému prihlásiť a využívať všetku funkcionality, ktorú mu superadministrátor povolí. Teda v systéme, v ktorom bude všetko povolené, bude môcť vyhľadávať používateľov, pridávať si ich do známych, komunikovať s nimi pomocou realtimeového chatu, zdieľať s nimi obsah na nástenke, nahrávať a spravovať vlastné súbory a používať všetku prídavnú funkcionality, ktorú budú v systéme zaisťovať nainštalované pluginy.

Klasickí používatelia sa budú môcť deliť do skupín, ktoré budú mať rôzne oprávnenia. Túto funkcionality bude možné dopĺňať aj pluginy. Pluginy môžu priniesť do systému aj ďalšie používateľské role, no žiadna z nich by sa nevyrovná superadministrátorovi.

3.3.1.2 Superadministrátor

Superadministrátor bude mať všetky práva ako klasickí používatelia, okrem toho bude mať možnosť spravovať celý systém. Bude môcť zapínať a vypínať súčasti systému, spravovať používateľov a pluginy. Správa pluginov bude umožnená kvôli bezpečnosti len superadministrátorovi. Do backendu sa bude prihlasovať cez samostatný prihlasovací formulár. No z backendu bude mať prístup aj do frontendu. Ak sa prihlási cez frontendový formulár, v systéme sa mu priradia len práva klasického používateľa a prístup do backendu bude mať až po odhlásení a nasledovnom prihlásení cez backendový formulár.

3.4 Databáza

TODO

TODO – db diagram

3.5 Triedy

3.5.1 Knižnice

V systéme bude naprogramovaná jedna asistenčná knižnica `assist.php`. Bude slúžiť na uľahčenie programovania niektorých často používaných funkcií ktoré framework neobsahuje. Táto knižnica bude potrebná takmer vo všetkých častiach aplikácie, preto bude vo frameworku načítaná automaticky pri každom spustení (bude zadefinovaná v konfiguračnom súbore `autoload.php`).

Bude obsahovať nasledujúce metódy:

- **toId** – na vstupe dostane dva parametre a na výstupe z nich vráti vygenerované iid.
- **fromId** – na vstupe dostane iid a na výstupe vráti pole údajov a to id a mark, pričom mark je príznak tabuľky ktorej dané id patrí.
- **getRealIp** – vráti reálnu IP adresu používateľa.
- **generateRandomString** – na vstupe môže dostať celočíselnú hodnotu. Ktorá bude udávať dĺžku náhodne vygenerovaného reťazca znakov. Predvolená dĺžka je 10 znakov.
- **formatTime** – na vstupe dostane čas vo formáte UNIXTIME a na výstupe vráti upravený dátum a čas vo formáte deň. mesiac. rok hodina:minúta:sekunda.
- **stripDiacritics** – na vstupe dostane reťazec znakov aj s diakritikou a na výstupe ho vráti bez diakritiky.
- **toUrl** - na vstupe dostane reťazec znakov aj s diakritikou, veľkými aj malými písmenami a medzerami. Na výstupe zmení veľké písmená na malé, medzery nahradí pomlčkami a odstráni diakritiku.
- **issetPost** – na vstupe môže dostať reťazec znakov, alebo dva reťazce znakov, pole znakov, alebo dve polia znakov. Prvý parameter, alebo pole parametrov hovorí o názve / názvoch postov, ktoré kontrolujeme. Ak ne zadáme druhý parameter, metóda skontroluje, či sú nastavené dané posty a či neobsahujú iba prázdny reťazec znakov. Ak ho neobsahujú vracia TRUE, inak vracia FALSE. Ak dostane na vstupe aj druhý parameter, zisťuje, či sú hodnoty daných postov zhodné s hodnotou / hodnotami v druhom parametre.
- **issetGet** – podobne ako predchádzajúca metóda, ale pre metódu get.

3.5.2 Modely

Modely v systéme slúžia na prácu s databázou. Sú rozdelené do logických celkov podľa tabuliek v databáze. V každom modeli sa môže nachádzať práca s jednou, alebo viacerými tabuľkami.

Modely obsiahnuté v systéme:

- **mcomments** – bude pracovať len s jednou databázovou tabuľkou a to s tabuľkou comments. Bude mať na starosti prácu s komentármi, pridávanie a vyberanie z databázy.
- **mfiles** – tento model bude pracovať až s tromi tabuľkami, a to s tabuľkou files, directories a shared_files. Bude mať na starosti prácu so súbormi a primárne bude určený pre správcu súborov a zdieľanie súborov na nástenke medzi používateľmi.
- **mchats** – model mchats bude pracovať s dvoma tabuľkami, a to s tabuľkou chats a opened_chats. Tento model bude určený pre realtimeový chat.
- **mpages** – aj tento model bude pracovať s dvoma tabuľkami. Tými tabuľkami sú pages a page_contents. Tento model bude určený pre CMS, ktoré bude integrované v systéme.
- **mposts** – model mposts bude mať na starosti prácu s tabuľkami posts a friends. Bude určený pre príspevky na nástenke medzi používateľmi.
- **msettings** – model msettings bude pracovať iba s tabuľkou settings. Bude určený pre správu globálnych nastavení systému.
- **museractivity** – tento model bude mať na starosti prácu s tabuľkou user_activity a bude zaznamenávať a čítať poslednú aktivitu používateľov.
- **musers** – bude sa starať o prácu s tabuľkami users a friends, teda bude mať za úlohu prácu s používateľmi.
- **TODO**

3.5.3 Core

Core bude časť systému, ktorá bude stáť nad všetkými ostatnými, no bude sa nachádzať nad frameworkom. V tomto systéme sa v core bude nachádzať jeden dôležitý controller a tým je MAIN_controller.

3.5.3.1 Main_controller

Bude základom všetkých ostatných controllerov. Bude v ňom prebiehať kontrola oprávnení používateľa. Tu sa budú zhromažďovať základné dáta, pre ďalšie časti aplikácie. Budú sa tu nachádzať pomocné funkcie, ku ktorým bude mať prístup celý systém. Tieto funkcie nebudú volateľné pomocou url, preto bude pred ich názvom znak „_“ (CodeIgniter ich bude detekovať ako nevolateľné pomocou url). Ďalej sa tu bude zaznamenávať posledná aktivita prihlásených používateľov, ktorá bude využívaná napríklad pri zisťovaní, či je daný používateľ online, alebo offline.

Pomocné funkcie v main_controller:

- **_pagesToArray** – vráti pole stránok aj s ich materskou stránkou. Bude slúžiť napríklad pre navigáciu na používateľských stránkach.
- **_getUserFriends** – vráti pole uid známych. Na vstupe sa bude zadávať do prvého vstupného parametra iid, koho známych chceme dostať a do druhého zadáme, či chceme aj nepotvrdených známych.
- **_getAllFriends** – bude vracať dvojrozmerné pole priateľov aj s bližšími informáciami o nich. Budú tu napríklad informácie o tom, či sú práve online, alebo offline, ich iid a všetky ostatné informácie, ktoré sú o nich uložené v databáze.

3.5.4 Controllery

Controllery budú v systéme rozdelené do troch logických skupín. Prvá časť bude s názvom page a bude mať na starosti zobrazovanie stránok jednotlivých používateľov. Druhá časť bude s názvom user, a bude určená pre frontend. Tretia časť bude časť admin a bude určená pre backend.

3.5.4.1 Page

Page sa bude nachádzať priamo v zložke controllers a bude to iba jeden controller.

3.5.4.1.1 Page

funkcie v controlleri page:

- **index** – na vstupe môže dostávať dva parametre. Prvým bude meno používateľa, ktorého podstránka sa má zobrazit'. Tento parameter bude povinný. Druhým parametrom bude názov podstránky, ktorá sa má zobrazit'. Tento bude nepovinný a ak sa nezadá, zobrazí sa

podstránka, ktorá sa nachádza v navigácii na prvom mieste.

3.5.4.2 User

Táto skupina controllerov bude uložená v zložke user a bude obsahovať nasledujúce controllery: home, logged, people, chat, adminpage a filemanager. **TODO**

3.5.4.2.1 Home

Tento controller bude mať na starosti prihlasovanie používateľov, odhlasovanie a registráciu nových používateľov. Ak sa cez tento controller prihlási superadmin, budú mu kvôli bezpečnosti pridelené práva klasického používateľa.

funkcie v controlleri home:

- **index, confirmlogin a logout** – z názvu funkcií sa dá jednoducho vyčítať, že budú slúžiť k zobrazovaniu prihlasovacieho formulára, overeniu prihlásenia a k odhláseniu.
- **registration a confirmregistration** – obe funkcie budú slúžiť k registrácii. Prvá k zobrazeniu registračného formulára a druhá k overeniu údajov a zapísaniu ich do databázy.

TODO – screeny

3.5.4.2.2 Logged

Hneď po prihlásení používateľa sa dostane k úlohe controller logged. Bude mať za úlohu načítanie hlavnej stránky používateľa. Tento controller bude dostupný iba prihláseným používateľom s dostatočným oprávnením.

funkcie v controlleri logged:

- **index** – bude slúžiť na načítanie neajaxového obsahu stránky, predá viewom základné informácie o používateľovi.
- **confirmaddpost** – bude slúžiť na spracovanie pridávaného postu na nástenku.
- **downloadfile** – na nástenke sa budú dať zdieľať aj súbory, bude slúžiť na ich stiahnutie.
- **ajax_getposts, ajax_getcomments, ajax_savecomment** – tieto funkcie budú slúžiť pre predávanie a spracovávanie ajaxových dát. Funkcie vracajúce dáta budú vracať dáta v json formáte. Budú vracať iba bezprostredne potrebné dáta. Ajax bude s funkciami komunikovať metódou post.

TODO – screeny

3.5.4.2.3 People

Tento controller bude mať za úlohu prácu s používateľmi a známymi. Aj tento controller bude dostupný iba prihláseným používateľom s dostatočným oprávnením.

funkcie v controlleri people:

- **friends** – bude slúžiť na zobrazenie všetkých potvrdených známych
- **search, addfriend, deletefriend** – vyhľadávanie používateľov, ich pridávanie a mazanie zo známych.
- **requests, acceptrequest, denyrequest** – zobrazenie žiadostí, ich potvrdenie, alebo zamietnutie.
- **ajax_getrequestscount** – táto funkcia bude vracať výstup pre ajax. Bude vracať počet nevybavených žiadostí.

TODO – screeny

3.5.4.2.4 Chat

Controller chat sa bude starať o realtimeový chat. Bude sa zobrazovať v samostatnom vyskakovacom okne. Väčšina funkcií bude spolupracovať s ajaxom. Bude dostupný iba prihláseným používateľom s dostatočným oprávnením.

funkcie v controlleri chat:

- **chatwindow** – funkcia, ktorá sa postará o načítanie okna s chatom a ajaxovým kódom.
- **ajax_loadnextlines, ajax_saveline** – **ajax_loadnextlines**, bude na výstupe vracať riadky z chatu, ktoré ešte neboli prečítané. **ajax_saveline** uloží do databázy riadok z konverzácie. Výstup je vo formáte json.
- **ajax_getusers** – vráti používateľov, ktorých má prihlásený používateľ v známych aj s doplňujúcimi informáciami, či od používateľa existuje neprečítaná konverzácia a či je daný používateľ online. Ak na vstupe nedostane funkcia požiadavku na dáta, pošle len MD5 hash z týchto dát (tento hash a jeho význam je popísaný v kapitole JavaScript a ajax). Výstup je vo formáte json.

TODO – screeny

3.5.4.2.5 Adminpage

Tento controller sa stará o správu používateľských stránok, teda je tiež dostupný len prihláseným používateľom s dostatočným oprávnením.

funkcie v controlleri adminpage:

- **index** – zobrazenie podstránok daného používateľa aj s prvkami na pridávanie a editáciu.
- **addpage, deletepage, editpagename** – pridávanie podstránok s možnosťou zmeny url danej stránky, úprava ich názvu a ich mazanie aj so všetkým obsahom.
- **orderup, orderdown** – zmena poradia podstránok
- **editpage, addcontent, deletecontent, editcontent** – zobrazenie obsahu podstránky, s možnosťou pridávania nového obsahu, a prvkami na zmenu poradia a mazania príspevkov.
- **contentorderup, contentorderdown** – zmena poradia príspevkov na podstránke.

TODO – screeny

3.5.4.2.6 Filemanager

Controller filemanager bude mať za úlohu pracovať so súbormi používateľa a bude sa starať aj o to, aby sa používateľmi nahraté spustiteľné súbory nedali spustiť na serveri. Bude sa zobrazovať v samostatnom vyskakovacom okne a bude dostupný len pre prihlásených používateľov s dostatočným oprávnením.

funkcie v controlleri filemanager:

- **myfiles** – zobrazuje súbory používateľa. Zobrazuje ich podľa informácií v databáze, lebo reálne súbory budú mať na serveri šifrované mená.
- **uploadfile, createdir** – uploadfile bude slúžiť na nahrávanie súboru na server, zašifrovanie jeho mena a zapísanie informácií do databázy. Createdir bude slúžiť na vytváranie zložiek. Tieto zložky nebudú reálne na serveri existovať, budú sa nachádzať len v databáze.
- **renamedir, renamefile, deletefile, deletedir** – tieto funkcie budú určené na premenovávanie a mazanie súborov a zložiek.
- **downloadfile** – táto funkcia sa postará o stiahnutie súboru používateľom, bez toho, aby sa spustiteľné súbory (napr. PHP skripty) spustili.
- **addpost, writepost** – addpost zobrazí používateľove súbory a dá mu na výber, ktorý z nich

che pridať na nástenku. Po výbere súboru sa k slovu dostane funkcia writepost, ktorá zobrazí formulár, v ktorom môže používateľ pridať komentár k zverejňovanému súboru a dá mu na výber aj známych, pre ktorých má byť daný súbor na nástenke viditeľný. Po úspešnom odoslaní formulára sa vyskakovacie okno automaticky zavrie.

TODO – screeny

3.5.4.3 Admin

Táto skupina controllerov bude uložená v zložke admin a bude obsahovať nasledujúce controllery: home a logged. TODO

3.5.4.3.1 Home

Controller home bude mať v tejto skupine za úlohu iba administrátorské prihlasovanie. Do tejto skupiny sa bude vedieť prihlásiť iba superadministrátor. Tieto controllery budú mať na starosti všetky dôležité nastavenia.

funkcie v controlleri home:

- **index** – zobrazí administrátorský prihlasovací formulár.
- **Confirmlogin** – overí prihlásenie a ak je používateľ superadministrátor a zadá správne prihlasovacie údaje, presmeruje používateľa do administračného controllera.

TODO – screeny

3.5.4.3.2 Logged

Tento controller bude mať za úlohu správu základných nastavení po prihlásení superadministrátora.

funkcie v controlleri logged:

- **index** – zobrazenie základných nastavení systému s možnosťou ich zmeny.
- **change** – zmena nastavenia, ktoré má v databáze booleanovskú hodnotu.
- **changewindow** – obrazovka na zmenu iného ako booleanovského nastavenia.
- **TODO**

TODO – screeny

...TODO ďalšie controllery

3.5.5 Zobrazovanie (views)

Zobrazovanie bude v systéme podobne ako controllery rozdelené do troch logických skupín, ktoré budú kopírovať skupiny týchto controllerov. Skoro každá funkcia bude mať svoj vlastný view. No v skupine vedľa page bude view s názvom window_close. Bude využívaný pri vyskakovacích oknách. Keď sa tento view zavolá, zatvorí sa dané okno v prehliadači pomocou JavaScriptu.

3.5.6 Pluginy

Pluginy budú najvyššou vrstvou nachádzajúcou sa nad frameworkom a samotným systémom. S frameworkom budú komunikovať pomocou systému a budú rozširovať a meniť jeho funkcionality.

TODO – viac o fungovaní pluginov

TODO – ako možno využiť systém za pomoci pluginov

3.5.7 JavaScript a ajax

Výsledný systém bude možné rozdeliť aj iným spôsobom. A to na aplikáciu serverovú a aplikáciu klientskú. Serverová aplikácia bude napísaná v PHP a klientská aplikácia bude generovaná tou serverovou. Klientská aplikácia sa bude zobrazovať používateľom v prehliadači vo formáte HTML v kombinácii s JavaScriptom a jeho ajaxovými volaniami. V JavaScripte sa bude využívať knižnica JQuery. JavaScript bude využívaný najmä na hlavnej stránke a pri chate.

Všetky ajaxové volania nebudú vždy vyžadovať od systému všetky údaje. Toto riešenie sa bude implementovať kvôli zníženiu dátového traffiku. Pri inicializácii ich vyžiada, ale okrem údajov, dostane aplikácia aj MD5 hash vygenerovaný z týchto údajov. Ak sa MD5 hash zhoduje s predchádzajúcim, už si aplikácia údaje pýtať nebude a bude dostávať iba MD5 hash. Hneď ako sa zmení hash (čo bude znamenať, že aj údaje boli zmenené), aplikácia to zaregistruje a pri ďalšom ajaxovom volaní si vypýta od systému aj zmenené údaje. Takéto MD5 hashe sa budú vytvárať pre každú potrebnú funkciu zvlášť a to len s údajov ktoré budú pre ňu potrebné. Nebudú sa hashovať kompletne všetky údaje, ale len tie, ktoré budú bezprostredne potrebné pre zistenie, či nastala zmena v údajoch.

JavaScriptové funkcie na hlavnej stránke (controller homepage, funkcia index):

- **súbor homepage.js** – tu sa bude nachádzať objekt `home` aj s pomocnými funkciami, ktoré sú potrebné pre zobrazovanie hlavnej stránky a aktualizovanie jej údajov.
 - **refreshUsers** – ajaxovým volaním získa zo systému zoznam používateľov aj s detailnými informáciami, ktorých má prihlásený používateľ v známych. V detailných informáciách budú informácie o tom, či je daný používateľ online, alebo offline, a či od neho existuje neprečítaná správa. Táto funkcia bude využívať vyššie popísaný MD5 hash.
 - **getPosts** – ajaxovým volaním bude získavať zo systému príspevky na nástenke. Aj táto funkcia bude využívať vyššie popísaný MD5 hash.
 - **addComent** – pomocou ajaxu uloží komentár do databázy.
 - **getComments** – ajaxovým volaním načíta komentáre prislúchajúce danému príspevku na nástenke. Nebude potrebovať MD5 hash, lebo v hashy, ktorý dostane funkcia `getPosts` bude obsiahnutý aj počet komentárov k príspevku. Až keď sa tento počet zmení, funkcia `getPosts` zavolá funkciu `getComments` s parametrom daného príspevku.
 - **getRequestsCount** – táto funkcia bude načítavať počet žiadostí o pridanie používateľa do známych. Nebude potrebovať MD5 hash, keďže jedno číslo zaberie pri prenose menej ako celý MD5 hash.
- **súbor generovaný systémom**
 - **init** – táto funkcia sa zavolá pri načítaní stránky a budú sa v nej inicializovať funkcie v súbore `homepage.js` a nastaví sa tu ich časovanie pre obnovovanie.

JavaScriptové funkcie v chate (controller chat, funkcia chatwindow):

- **súbor chat.js**
 - **loadLines** – ajaxovým volaním načíta neprečítané časti danej konverzácie. Nebude musieť využívať MD5 hash, lebo systém sám pošle iba neprečítané konverzácie.
 - **updateLine** – pomocou ajaxu odovzdá systému riadok chatu ktorý systém uloží do databázy.
- **súbor generovaný systémom**
 - **onKeyUp** – pomocou JQuery bude kontrolovať, či nebol stlačený enter. Ak bol, zavolá funkciu `updateLine` a pošle jej text ktorý sa uloží do databázy.

4 Implementácia systému

Implementácia prebiehala takmer bez problémov podľa špecifikácie systému. Systém som programoval na linuxe, využíval som webový server apache a databázový server MySQL. Na editáciu kódu som používal multiplatformový editor Geany.

4.1 Problémy pri implementácii

Pri implementácii som narazil iba na málo problémov, ktoré som musel riešiť. **TODO**

5 Záver

TODO

5.1 Zhodnotenie výslednej aplikácie

TODO

5.2 Návrhy na vylepšenie v budúcnosti

TODO

6 Zoznam použitej literatúry

- Lednár, Matej.: Príručka programátora - Prehľadný sprievodca jazykmi XHTML 1.1, CSS 2.1 a JavaScript 1.5+ s úvodom do DOM. Bratislava: MLD Group, 2009, s.776. [ISBN 978-80-89448-00-5](#)
- Lednár, Matej.: Príručka programátora - Aplikujeme DOM s jazykmi XHTML, CSS a JavaScript. Bratislava: MLD Group, 2009, s. 207. [ISBN 978-80-89448-01-2](#)
- Lednár, Matej.: Príručka programátora - Prehľadný sprievodca jazykom XHTML 1.1. Bratislava: MLD Group, 2009, s.178. [ISBN 978-80-89448-04-3](#)
- Lednár, Matej.: Príručka programátora - Prehľadný sprievodca jazykom JavaScript 1.5+. Bratislava: MLD Group, 2009, s. 238, [ISBN 978-80-89448-02-9](#)
- <http://www.phpframeworks.com/>
- <http://codeigniter.com/>

7 Prílohy

TODO