

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
KAROLA ADLERA 5, 841 02 BRATISLAVA



Webová aplikácia pre server s OS Ubuntu

Richard Boldiš

Bratislava 2018

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
KAROLA ADLERA 5, 841 02 BRATISLAVA

Webová aplikácia pre server s OS Ubuntu
Praktická časť odbornej zložky maturitnej skúšky

Richard Boldiš

Študijný odbor: 26 75 M elektrotechnika

Oblasť prípravy: Informatické a telekomunikačné systémy

Konzultant: Ing. Asena Benetinová

Bratislava 2018

ZAMEŇTE STRANU ZA ZADANIE PRÁCE

Abstrakt

Práca je obsahovo zameraná na spôsob uľahčenia administrácie serveru pomocou grafického rozhrania, ktoré komunikuje s rôznymi spravovanými službami. Popisuje rôzne časti a návrh aplikácie, vysvetľuje metódy, ktorými bola správa služieb uskutočnená.

Abstract

The aim of the work is focused on making server administration easier using a graphical interface, which communicates with various managed services. The work describes various parts and concepts of the application, explains implemented methods and technologies which were used in administration.

Čestné vyhlásenie

Vyhlasujem, že predloženú maturitnú prácu na tému „Webová aplikácia pre server s OS Ubuntu“ som vypracoval samostatne, s použitím uvedených zdrojov a neporušil som autorský zákon.

Bratislava, 13. apríl 2018

.....

Richard Boldiš

PodĎakovanie

Ďakujem svojmu konzultantovi Ing. Asene Benetinovej za pripomienky a odbornú pomoc pri vypracovávaní práce, ako aj zvyšku učiteľského zboru za podporu a pomoc či už pri dokumentácií, alebo samotnej práci. Zároveň ďakujem priateľom a mojej rodine, ktorá mi bola v čase písania práce oporou.

Obsah

Abstrakt.....	3
Abstract.....	4
Čestné vyhlásenie	5
Poďakovanie.....	6
Úvod.....	10
1. Operačný systém.....	11
2. Výber služieb pre správu.....	11
2.1. Stránky WWW	11
2.2. Databáza.....	11
2.3. DNS	12
2.4. Elektronická pošta	12
2.5. Firewall	12
2.6. Úlohy	12
3. Výber vývojového prostredia.....	13
3.1. PhpStorm.....	14
3.2. VirtualBox	15
4. Výber technológií.....	15
4.1. PHP	15
4.1.1. Laravel.....	15
4.2. JavaScript.....	16
4.2.1. Vue.....	16
4.2.2. Axios.....	17
5. Webová aplikácia pre server s OS Ubuntu	17

5.1.	Všeobecná funkcionálnosť	18
5.1.1.	Ovládanie služby	18
5.1.2.	Ovládacie prvky	19
5.2.	Dashboard	20
5.2.1.	Spôsob získavania údajov	22
5.3.	Stránky WWW	23
5.3.1.	Konfiguračné súbory	24
5.3.2.	Aktivácia HTTPS a PHP	25
5.4.	Databáza	25
5.4.1.	Správa užívateľov	26
5.4.2.	Prideľovanie práv	26
5.5.	DNS	27
5.5.1.	Správa zón a ich záznamov	28
5.5.2.	Ukladanie záznamov	29
5.5.3.	Export & Import	29
5.6.	Elektronická pošta	30
5.7.	Firewall	31
5.8.	Úlohy	32
6.	API	32
7.	Inštalácia	33
8.	Cenová kalkulácia nákladov	33
	Záver	36
	Zoznam použitej internetovej dokumentácie	37
	ZOZNAM PRÍLOH	38
	PRÍLOHA A	39

PRÍLOHA B	40
-----------------	----

Úvod

Administrácia služieb serveru môže byť niekedy náročnejšia na čas a vedomosti. Grafické rozhrania výrazne skracujú čas strávený čas za konzolou a chybovosť konfigurácie.

V súčasnosti existuje niekoľko grafických administračných rozhraní na základe webovej aplikácie pre správu služieb na serveri. Väčšinou sa jedná o správu základných služieb ako napríklad DNS (Domain Name Service), firewall, databáza a podobne. Ich úlohou je umožniť jednoduchú a rýchlu správu vyžadovanej služby bez väčších skúseností. Grafické administračné rozhrania si väčšinou získavajú obľubu u vlastníkov domácich serverov, alebo malých firiem.

Prácu som sa rozhodol vypracovať najmä z dôvodu nadobudnutia nových skúseností v oblasti programovania webu a práce s operačným systémom Ubuntu, ale taktiež aj s cieľom uľahčiť systémovým administrátorom prácu pri správe serveru, keďže niektoré úkony sa pri správe môžu niekoľkokrát opakovať.

V práci som sa zameral na najčastejšie vyskytované služby, ktorých konfigurácia môže byť v niektorých oblastiach zdĺhavejšia a vykonáva sa frekventovanejšie. Ide predovšetkým o správu DNS zón a ich záznamov, ktorá celkový nápad tohto projektu vo mne vzbudila. Neskôr sa k tomu pridala automatizácia úpravy stránok bežiacich pod službou Nginx a správa pravidiel firewallu, ktorý tvorí súčasť každého zariadenia pripojeného do internetovej siete. Projekt sa zaoberá taktiež jednoduchou a rýchlou správou úloh služby Cron, správou účtov a aliasov elektronickej pošty služieb Dovecot a Postfix.

1. Operačný systém

Ubuntu je jeden z mnohých operačných systémov rodiny Linux s otvoreným zdrojom a vyvíjaný komunitou. Ide o distribúciu založenú na architektúre Debian. Ubuntu je vhodný pre stolné počítače, notebooky, servery aj inteligentné telefóny. Tento projekt sa venuje konkrétne verzii 16.04 LTS (Long Term Support) pre servery. Zo skratky LTS vyplýva, že ide o verziu s dlhodobou podporou, ktorá je poskytovaná až po dobu piatich rokov. LTS verzie vychádzajú každé dva roky.

2. Výber služieb pre správu

Vyberal som z najčastejšie vyskytovaných serverových služieb, ktorých automatizovaná administrácia by mohla výrazne zjednodušiť a zrýchliť proces správy serveru.

2.1. Stránky WWW

Služba WWW (World Wide Web) umožňuje stiahnutie stránok klientom (webovým prehliadačom). Pri výbere konkrétneho softvéru som sa rozhodoval medzi Apache a Nginx. Po zvážení výhod a nevýhod týchto dvoch softvérov som sa rozhodol pre Nginx vo verzii 1.10. V porovnaní s Apache, Nginx disponuje rýchlejším spracovaním požiadaviek, ľahkou škálovateľnosťou a nízkymi nárokmi na výpočtové zdroje.

2.2. Databáza

Neodmysliteľnou službou dynamických stránok je databáza, do ktorej sa ukladajú rôzne údaje od dát návštevníkov po obsah stránky a nastavenia. Softvér databázy som vybral MariaDB 10.2, ktorý je plne kompatibilný či už protokolom, alebo konfiguráciou MySQL. Ako pri službe Nginx, MariaDB som vybral z dôvodu nižších nárokov na výpočtové zdroje.

2.3. DNS

Aby si návštevník stránok nemusel pamätať IP adresy stránok, resp. služieb, potrebujeme službu DNS (Domain Name System). Umožňuje preklad webových názvov na IP adresy. Taktiež odpovedá na iné dôležité operácie, ako je napr. určenie serveru elektronickej pošty (záznam MX), alebo lokalizovanie služby na serveri (záznam SRV). Software, ktorý implementuje DNS protokol som vybral najznámejší Bind 9 od ISC (Internet Systems Consortium).

2.4. Elektronická pošta

E-Mail (Electronic Mail) umožňuje výmenu správ medzi užívateľmi. Aby sme mohli správy odosielať, potrebujeme službu implementujúcu SMTP (Simple Mail Transfer Protocol). Jeden z najznámejších software, ktorý tento protokol implementuje je Postfix. Pre čítanie pošty zo serveru existuje protokol IMAP (Internet Message Access Protocol) a POP (Post Office Protocol), ktoré implementuje služba Dovecot.

2.5. Firewall

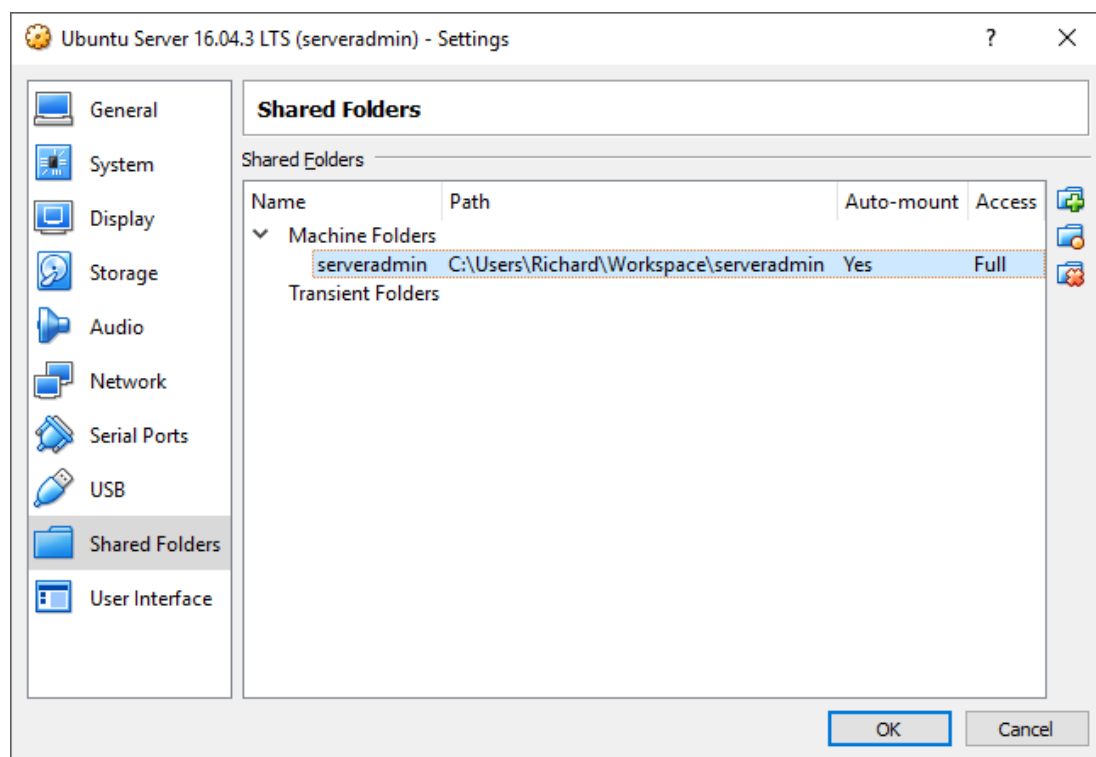
Firewall kontroluje, či sú prichádzajúce a odchádzajúce pakety povolené, zakázané, alebo obmedzené na základe portov a protokolov tretej a štvrtej vrstvy. Každý server by mal byť vybavený aspoň interným firewallom, resp. softvérom, ktorý vykonáva kontrolu priamo na chránenom zariadení. Softvér UFW (Uncomplicated Firewall), ktorý som pre túto úlohu vybral nahrádza existujúci softvér iptables, ktorý konfiguráciu zjednodušuje.

2.6. Úlohy

V určitých intervaloch je potrebné na serveri vykonávať úlohy, ktoré automatizujú správu serveru. Tieto úlohy môžu napríklad odstrániť dočasné súbory, odstrániť staré záznamové súbory, alebo vykonať iný definovaný príkaz. Operačný systém Ubuntu 16.04 disponuje softwarom Cron.

3. Výber vývojového prostredia

Keďže cieľom projektu je administrácia serveru s operačným systémom Ubuntu, bolo potrebné pripraviť ďalší vývojový počítač, na ktorom by tento operačný systém pre testovacie účely bežal. Pre nedostatok hardware som operačný systém Ubuntu nainštaloval v programe VirtualBox na virtuálny počítač. Týmto sa taktiež vyriešil problém s prenosom súborov medzi zariadeniami počas vývoja. Niektoré časti projektu sú generované automaticky a manuálny prenos súborov medzi zariadeniami by bol nepraktický a zdĺhavý. VirtualBox umožňuje vytvoriť tzv. zdieľaný priečinok, ktorý umožňuje zdieľať súbory počítača vývojára s virtuálnym strojom. Súbory sú ihneď dostupné virtuálnemu stroju a v prípade výpadku virtuálneho stroja sú súbory projektu v bezpečí.

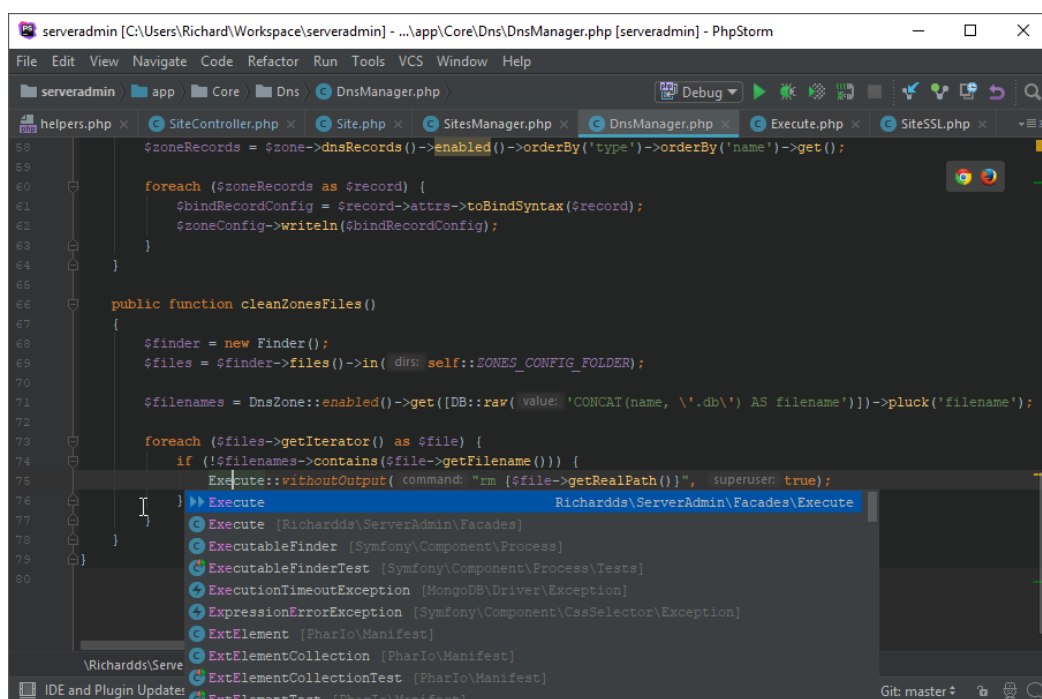


Obr. č. 1: Nastavenie zdieľaných priečinkov v softvéri VirtualBox

Aby zdieľanie súborov bolo možné uskutočniť, bolo potrebné na virtuálnom stroji nainštalovať softvér Guest Additions, ktorý sprostredkuje komunikáciu medzi hosťovským počítačom a virtuálnym strojom. Po každom zapnutí virtuálneho stroja je na ňom potrebné spustiť príkaz, ktorým sa zdieľaný priečinok primontuje do priečinku **/opt/serveradmin**.

3.1. PhpStorm

Ako primárny nástroj pri tvorbe projektu som použil PhpStorm od spoločnosti JetBrains. Jedná sa o cross platform IDE (Integrated Development Environment), ktorý poskytuje editor kódu pre PHP, HTML a JavaScript s prevenciou chýb, automatickým dopĺňaním, generátor kódu a komentárov a mnoho ďalších nástrojov pre zrýchlenie a uľahčenie vývojového procesu. PhpStorm taktiež disponuje doplnkami pre podporu externých nástrojov a pomocníkmi pre vývoj s konkrétnymi frameworkami ako sú napr. Laravel, Bootstrap, Vue.js a pod.



Obr. č. 2: Automatické dopĺňanie v PhpStorm

3.2. VirtualBox

Pre virtualizáciu som použil softvér VirtualBox vyvinutý spoločnosťou Oracle Corporation, vďaka ktorému je možné rozbehať cudzí operačný systém priamo na počítači vývojára. Keďže sa jedná o cross platform softvér s otvoreným zdrojom, ktorý je podporovaný na viacerých operačných systémoch, virtuálny stroj je možné jednoducho prenášať medzi počítačmi vývojára, ktoré nemajú ten istý druh operačného systému.

4. Výber technológií

Aby som sa mohol primárne venovať cieľu tejto práce, rozhodol som sa projekt napísať ako nadstavbu PHP frameworku Laravel, ktorý základnú funkcionality, to jest smerovanie požiadaviek, autentifikáciu, komunikáciu s databázou, zabezpečenie a mnoho iného už obsahuje.

Back-end časť tvoria prevažne programovacie jazyky PHP a BASH. Na strane klienta (front-end) tvoria užívateľské rozhranie a dodávajú funkcionality pre komunikáciu s back-endom technológie Vue.js, Bootstrap a Axios.

4.1. PHP

PHP je jeden z najstarších a zároveň najviac používaných programovacích jazykov navrhnutý pre vývoj webových aplikácií. Zabezpečuje väčšinu súčastí projektu, ako je generovanie konfiguračných súborov, spúšťanie príkazov na pozadí a podobne.

4.1.1. Laravel

Laravel patri v dnešnej dobe medzi najviac používané PHP frameworky pre tvorbu webových aplikácií používajúce MVC (Model View Controller) architektúru. Disponuje bohatou funkcionality, ktorá mi umožnila venovať sa viac cieľom tejto práce ako základným procesom spracovania požiadavky, smerovania žiadostí, komunikácie s databázou, atď...

4.2. JavaScript

JavaScript je programovací jazyk, ktorý sa po stiahnutí zo serveru spustí priamo v prehliadači návštevníka stránky. Táto technológia mi umožnila spolu s frameworkom Vue vytvoriť užívateľské rozhrania omnoho dynamickejšie, čo užívateľovi skráti dobu potrebnú na vykonanie rôznych akcií.

4.2.1. Vue

Vue je jeden zo známejších frameworkov pre tvorbu dynamických užívateľských rozhraní. Umožňuje vytvoriť vlastné HTML elementy, ktoré sú po načítaní frameworku nahradené komponentmi obsahujúce šablónu skladajúcu sa z rôznych iných HTML elementov. Týmto spôsobom sa v zdroji aplikácie eliminovali opakujúce sa bloky HTML elementov.

Logika napísaná v programovacom jazyku JavaScript je spúšťaná na každom komponente nezávisle. To nám umožňuje rozhranie spestriť rôznymi dynamickými elementami.

```

<template>
  <i :class="['fa', 'fa-' + icon, {'fa-spin' : spin}]" aria-hidden="true"></i>
</template>

<script>
  export default {
    props: {
      icon: {default: 'question'},
      spin: {default: false},
    }
  }
</script>

```

Obr. č. 3: Jednoduchý Vue komponent – Ikona

Obrázok č. 3 zobrazuje komponent reprezentovaný novým elementom `<sa-icon />` akceptujúci dva nepovinné parametre. Tento komponent skracuje syntax pre napísanie elementu zobrazujúci ikonu z knižnice font-awesome.

4.2.2. Axios

Axios je knižnica, ktorá poskytuje grafickému rozhraniu (front-endu) funkcionality odosielať rôzne HTTP žiadosti na back-end a tým získať údaje o serveri, alebo vykonávať vzdialené procedúry. Viac o volaní API sa môžete dočítať v kapitole 6.

```
grantPrivileges() {  
  this.grantPrivilegesForm.start();  
  axios.patch('/api/database/grant', this.grantPrivilegesForm.attributes).then(() => {  
    this.grantPrivilegesForm.finish();  
    this.loadPrivileges();  
  }).catch(error => {  
    this.grantPrivilegesForm.crash(error);  
  });  
},
```

Obr. č. 4: Metóda volajúca vzdialenú procedúru pre zmenu parametrov

5. Webová aplikácia pre server s IS Ubuntu

Prístup do administratívneho rozhrania je chránený heslom, ktoré si administrátor zvolí pri inštalácii. Zmenu hesla je možné vykonať nástrojom artisan a to zadáním príkazu **php artisan admin:password**.

```
C:\Users\Richard\Workspace\serveradmin>php artisan admin:password  
  
New password:  
>  
  
Confirm password:  
>  
  
Password successfully changed.
```

Obr. č. 4: Zmena hesla cez nástroj artisan

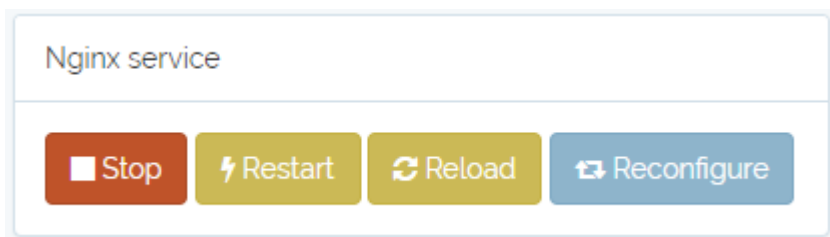
Po prihlásení sa nám zobrazí úvodná stránka s prehľadom základných informácií o serveri. V hornom menu sa môžeme preklikať medzi jednotlivými sekciami administračného rozhrania. Každá sekcia predstavuje správu jednej služby.

5.1. Všeobecná funkcionlita

Všetky sekcie fungujú na spoločnom princípe. Po načítaní stránky sa v prehliadači načíta Vue framework, ktorý zobrazí časti stránky pomocou naprogramovaných Vue komponentov. Každý komponent je reprezentovaný novým rezervovaným HTML elementom, ktorý je pri štarte aplikácie na strane prehliadača zachytený a následne nahradený šablónou z komponentu, na ktorom sa spustí jeho logika. V prípade načítania, alebo odosielania údajov z back-endu sa používa knižnica Axios, ktorá zabezpečuje komunikáciu s API.

5.1.1. Ovládanie služby

Každá sekcia obsluhujúca službu zahŕňa v hornej časti komponent pre ovládanie služby. Väčšinu služieb je možné zapnúť, vypnúť a načítať. V prípade služieb, ktorým sú konfiguračné súbory generované aplikáciou, je možné konfiguračné súbory znovu vygenerovať.










Obr. č 5: Komponent pre ovládanie služby





V triede **ServiceController** sú v premennej **\$services** definovaný manažéri služieb. Manažér služieb je trieda rozširujúca podtriedu **Service**, ktorá obsahuje základné metódy pre ovládanie služby. V prípade, že je trieda implementovaná interfacom **ConfigurableService**, ovládanie služby

poskytuje tlačidlo „**Reconfigure**“, ktoré spúšťa implementovanú metódu **configure**.

5.1.2. Ovládacie prvky

Každá sekcia umožňuje rôznym spôsobom službu ovplyvňovať. Pri tvorbe grafického rozhrania som sa snažil použiť prvky, ktoré intuitívne napovedajú o ich funkcionalite. V nasledujúcej tabuľke sa nachádzajú vysvetlenia k rôznym prvkom, s ktorými sa administrátor môže stretnúť.

Ikona	Akcia	Popis
	Odstrániť	Vyskytuje sa pri prvkoch, ktoré je možné týmto tlačidlom odstrániť.
	Pridať	Vyskytuje sa pod zoznamom prvkov, do ktorého je možné týmto tlačidlom pridať nový prvok.
	Deaktivovať	Vyskytuje sa pri prvkoch, ktoré je možné týmto tlačidlom deaktivovať a tým dočasne a nedeštruktívne prvok vyradiť z pomedzi aktívnych prvkov.
	Aktivovať	Vyskytuje sa pri prvkoch, ktoré je možné týmto tlačidlom aktivovať a tým prvok zaradiť naspäť medzi aktívne prvky.
	Upraviť	Vyskytuje sa pri prvkoch, ktoré je možné týmto tlačidlom upraviť, resp. upraviť jeho parametre.
	Zrušiť	Vyskytuje sa pri prvkoch, na ktorých práve prebieha akcia, ktorú je možné týmto tlačidlom zrušiť.
	Potvrdiť	Vyskytuje sa pri prvkoch, na ktorých práve prebieha akcia, ktorú je možné týmto tlačidlom potvrdiť.

	Kopírovať	Vyskytuje sa pri poliach pre zadávanie textu. Kliknutím na tlačidlo sa text z polia skopíruje.
	Generovať	Vyskytuje sa pri poliach pre zadávanie textu. Kliknutím na tlačidlo sa text podľa vygeneruje, alebo automaticky doplní na základe spätne vyplnených údajov, alebo iných aktuálnych údajov, ako je napr. čas.
 	Ostatné	Ostatné tlačidlá, ktorých vysvetlenie je možné nájsť v konkrétnych kapitolách opisujúce sekciu, ktorá tieto tlačidlá používajú.

Tab. č. 1: Ovládacie prvky grafického rozhrania

5.2. Dashboard

Úvodná sekcia umožňuje server vypnúť, reštartovať a zobrazíť základné informácie o serveri vrátane stavu procesora, pamäte RAM a SWAP. Tieto údaje sa získavajú prevažne spúšťaním príkazov na pozadí a následnou analýzou sa z výsledku príkazu extrahujú údaje, ktoré potrebujeme.

ServerAdmin v0.0.2-dev Dashboard Sites Database DNS Mail Firewall Cron Logout

Dashboard

Shutdown Restart

Info

OS	Ubuntu 16.04.3 LTS
Uptime	7 hours 30 minutes 14 seconds
Last offline	10. April 2018 13:00:17
Back-end user	serveradmin (110)

Processor

Model	Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
Cache	8192 KB
Current clock	4007 Mhz
Number of cores	2

Memory

Total	992 MB (58.2% used)
Used	394 MB
Free	136 MB
Shared	28 MB
Cache	460 MB
Available	415 MB

Swap

Total	1021 MB (0% used)
Used	0 MB
Free	1021 MB

Obr. č. 6: Základné informácie o serveri

V nasledujúcich tabuľkách sú vysvetlené údaje zobrazené na úvodnej stránke „Dashboard“.

OS	Úplný názov operačného systému.
Uptime	Zobrazuje koľko je server v nepretržitej prevádzke.
Last offline	Dátum a čas posledného výpadku, resp. kedy bol server naposledy vypnutý.
Back-end user	Názov a id systémového užívateľa pod ktorým aplikácia beží a vykonávajú sa príkazy. (Okrem príkazov, ktoré je nevyhnutné spustiť ako užívateľ root.)

Tab. č. 2: Zoznam prvkov tabuľky „Info“

Model	Názov značky a modelu procesoru @ frekvenciu procesoru.
Cache	Veľkosť pamäte cache procesoru.
Current clock	Aktuálna frekvencia procesoru.
Number of cores	Počet jadier procesoru.

Tab. č. 3: Zoznam prvkov tabuľky „Processor“

Total	Veľkosť nainštalovanej operačnej pamäte.
Used	Veľkosť použitej operačnej pamäte.
Free	Veľkosť nepoužitej operačnej pamäte.
Shared	Veľkosť použitia zdieľanou pamäťou tmpfs.
Cache	Veľkosť použitia vyrovnávajúcou pamäťou a cache.
Available	Veľkosť dostupnej pamäte, pri ktorej je možné spustiť program bez využitia SWAP.

Tab. č. 4: Zoznam prvkov tabuľky „Pamäť“

Total	Veľkosť nastavenej pamäte SWAP.
Used	Veľkosť použitej pamäte SWAP.
Free	Veľkosť nepoužitej pamäte SWAP.

Tab. č. 5: Zoznam prvkov tabuľky „Swap“

5.2.1. Spôsob získavania údajov

Aby aplikácia dokázala získať informácie o serveri, bolo potrebné implementovať funkcionality, ktorá by umožňovala komunikáciu priamo s terminálom serveru. Pre komunikáciu s terminálom slúži trieda **CommandExecuter**, ktorá sa volá cez fasádu **Execute**. Táto trieda umožňuje spustiť príkaz spôsobom vykonaním príkazu a vrátením výsledku, alebo iba vykonaním príkazu. Príkazy sú východiskovo spúšťané pod back-end užívateľom, avšak v prípade privilegovaných príkazov je možné príkaz spustiť ako root.

Metódy pre získavanie informácií o serveri využívajú spôsob vykonania príkazu s vrátením výsledku. Po navrátení hodnoty sa výsledok analyzuje a normalizuje na očakávaný formát. V prípade výsledku s viacerými hodnotami sú údaje zabalené do servisného objektu.

```

/**
 * @return \Richardds\ServerAdmin\Core\Server\Memory
 * @throws InvalidCommandOutputException
 */
public static function memory(): Memory
{
    $raw_output = Execute::output( command: 'free -m');
    $ex = explode( delimiter: ' ', regex_single($raw_output, regex: '/Mem:(.*)\n/m'));
    $values = [];

    foreach ($ex as $i => $value) {
        if ('' != $value && is_numeric($value)) {
            $values[] = intval(trim($value));
        }
    }

    if (6 != count($values)) {
        throw new InvalidCommandOutputException( message: 'Invalid command output.', $raw_output);
    }

    return new Memory($values[0], $values[1], $values[2], $values[3], $values[4], $values[5]);
}

```

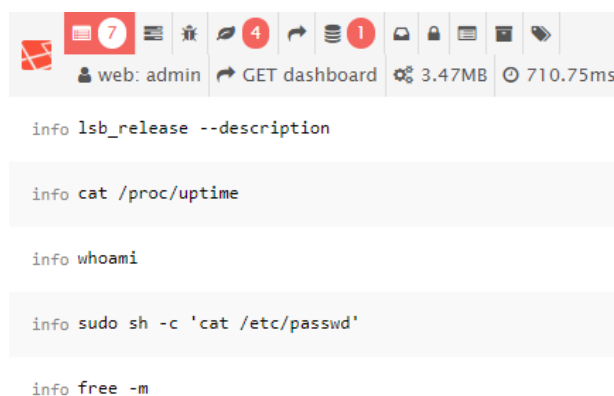
Obr. č. 7: Ukážka metódy pre získanie údajov o pamäti RAM

Na obrázku č. 7 vidíme spôsob analýzy výstupných údajov získané pomocou fasády **Execute**. Riadok, ktorý určuje údaje pamäte RAM vytiahneme pomocou regex metódy. Ten následne rozdelíme na stĺpce a číselný údaj z každého stĺpca prefiltrujeme funkciami **trim** a **intval**, ktoré odstránia redundantné medzery a text prekonvertujú na číslo. V poslednom kroku sa pre prípad nesprávneho výstupu príkazu skontroluje počet údajov, ktoré sme spomenutým postupom získali. V prípade neočakávaného počtu údajov sa vyhodí výnimka. Na podobnom princípe fungujú aj ostatné metódy pre získavanie údajov.

```
root@server:~# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	992	394	136	28	460	415
Swap:	1021	0	1021			

Obr. č. 8: Výstup príkazu „free -m“



Obr. č. 9: Zaznamenané príkazy pri generovaní sekcie

5.3. Stránky WWW

Táto sekcia umožňuje spravovať stránky bežiace pod službou Nginx. Stránkam je možné prideliť certifikát, resp. zapnúť protokol HTTPS a povoliť spúšťanie PHP skriptov. Údaje zobrazené v tejto sekcii sú získané z databázy, kde je konfigurácia stránok uložená.

Sites			
Name	PHP	SSL	
BRADTKE.ORG	PHP Disabled	SSL Enabled	<input checked="" type="checkbox"/>
HOWE.NET	PHP Enabled	SSL Disabled	<input type="checkbox"/>

Obr. č. 10: Správa stránok WWW

5.3.1. Konfiguračné súbory

Každá stránka má konfiguráciu uloženú v priečinku **/etc/nginx/sites-available**. Tieto konfiguračné súbory sú generované na základe parametrov stránok.

```
server {
    listen 443 ssl http2;

    server_name test.local;
    root /var/www/test.local/public;

    ssl_certificate /etc/nginx/ssl/certs/test.local.pem
    ssl_certificate_key /etc/nginx/ssl/keys/test.local.pem

    location / {
        try_files $uri =404;
    }

    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_pass php-handler;
        fastcgi_split_path_info ^(.+\.php) (/.+)$;
    }

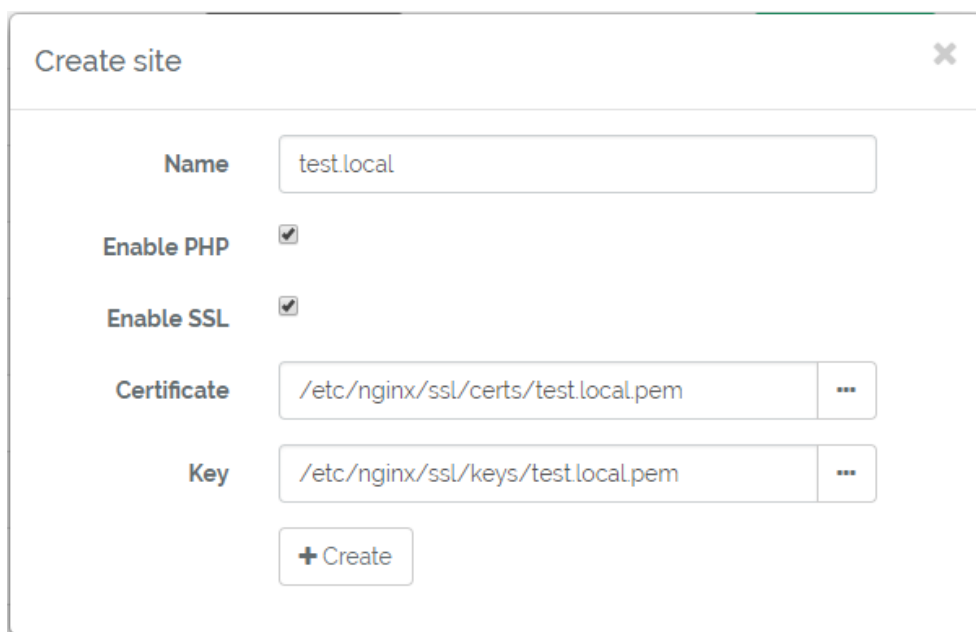
    access_log /var/www/test.local/logs/nginx_access.log;
    error_log /var/www/test.local/logs/nginx_error.log;
}
```

Obr. č. 11: Konfiguračný súbor stránky

Na obrázku č. 11 vidíme vygenerovaný konfiguračný súbor pre stránku s názvom **test.local**, kde je povolené spúšťanie skriptov PHP (zelená farba) aj zapnutý protokol HTTPS (modrá farba).

5.3.2. Aktivácia HTTPS a PHP

Šifrovanie prenosu údajov stránky a spúšťanie PHP skriptov môžeme povoliť pri vytváraní novej stránky, alebo kliknutím na tlačidlá zobrazujúce stav SSL (HTTPS) a PHP.



The image shows a 'Create site' dialog box. It has a title bar with the text 'Create site' and a close button (X). The dialog contains the following fields and controls:







- Name:** A text input field containing 'test.local'.
- Enable PHP:** A checkbox that is checked.
- Enable SSL:** A checkbox that is checked.
- Certificate:** A text input field containing '/etc/nginx/ssl/certs/test.local.pem' and a dropdown arrow (three dots).
- Key:** A text input field containing '/etc/nginx/ssl/keys/test.local.pem' and a dropdown arrow (three dots).
- + Create:** A button at the bottom of the dialog.

Obr. č. 12: Vytvorenie novej stránky

5.4. Databáza

Cieľom tejto sekcie je umožniť vytvoriť, alebo odstrániť databázu, spravovať databázových užívateľov a urýchlene im priradiť práva pre vykonávanie akcií s databázou. Tabuľka databáz navyše obsahuje informácie ako počet tabuliek a veľkosť, ktorú databáza zaberá na úložisku.

Údaje zobrazené v tejto sekcii sú získané spustením SQL príkazov voči databáze. Databáza je modifikovaná taktiež spúšťaním SQL príkazov, takže generovanie konfiguračných súborov nie je v tomto prípade potrebné. Aby aplikácia dokázala modifikovať globálne nastavenia a ostatné databázy, bolo nutné užívateľovi, pod ktorým aplikácia príkazy vykonáva povoliť globálne práva.

Databases					
Name	Character set	Collation	Tables	Size	
abc	utf8mb4	utf8mb4_slovak_ci	0	0 B	 
information_schema	utf8	utf8_general_ci	75	176 KB	 
mysql	utf8mb4	utf8mb4_general_ci	30	685.74 KB	 

Obr. č. 13: Správa databáz

5.4.1. Správa užívateľov

Aplikácia poskytuje správu databázových užívateľov. Na obrazovku správy užívateľov sa dostaneme kliknutím na tlačidlo „**Manage users**“ v pravej časti okna pre ovládanie služby. Na tejto obrazovke je možné vytvoriť nových užívateľov, alebo ich zmazať. Existujúcim užívateľom je možné zmeniť heslo kliknutím na im prislúchajúce tlačidlo s ikonou kľúča.

Create user

User


wordpress


Host

localhost

Password

.....





+ Create

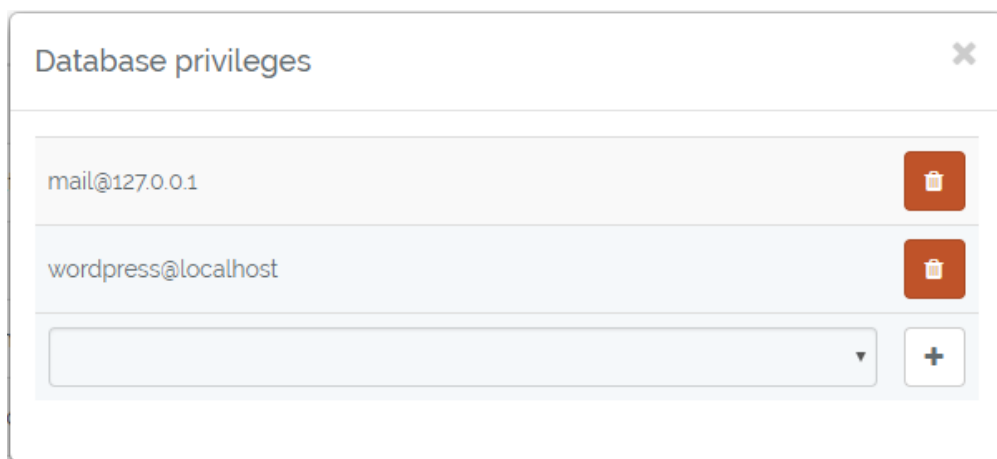
Obr. č. 14: Formulár pre vytvorenie nového užívateľa

5.4.2. Pridelovanie práv

Aby novovytvorený užívatelia mohli manipulovať s údajmi v databázach, je im nutné priradiť práva. Kliknutím na kľúč vedľa databázy sa zobrazí zoznam, v ktorom sa nachádzajú užívatelia s povolením vykonávať akcie s databázou. Po pridaní povolenia sa vybranému užívateľovi automaticky

pridelia práva **SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLE, LOCK TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, EVENT a TRIGGER**. Z dôvodu zrýchlenia procesu pridávania práv sa užívateľovi vždy priradia pravé vyššie spomenuté práva. Pri odstránení užívateľa zo zoznamu sa mu odstránia úplne všetky práva týkajúce sa konkrétnej databázy.

Pri zobrazení zoznamu užívateľov s povoleniami vykonávania akcii pre konkrétnu databázu sa postupne prechádzajú všetci užívatelia a kontroluje sa, či majú priradené všetky zo spomenutých práv. Pokiaľ užívateľ nemá priradené všetky práva, nie je zobrazený v zozname užívateľov s povolením vykonávania akcii.



Obr. č. 15: Zoznam užívateľov s povolením vykonávať akcie

5.5. DNS

Sekcia správy DNS poskytuje urýchlenú editáciu súborov zón a ich záznamov. Údaje sú najprv uložené do databázy a potom sú z nich vygenerované konfiguračné súbory. Každá zóna je reprezentovaná zónovým súborom. Tieto súbory sa nachádzajú v priečinku **/etc/bind/zones**. Každú zónu je potrebné zaregistrovať v konfiguračnom súbore **/etc/bind/named.conf.local**, kde je uvedený názov zóny, typ zóny (v našom prípade master) a cesta ku súboru zóny.

```
@ IN SOA mosciski.com admin.mosciski.com (19840910 64800 12600 4320000 43200)

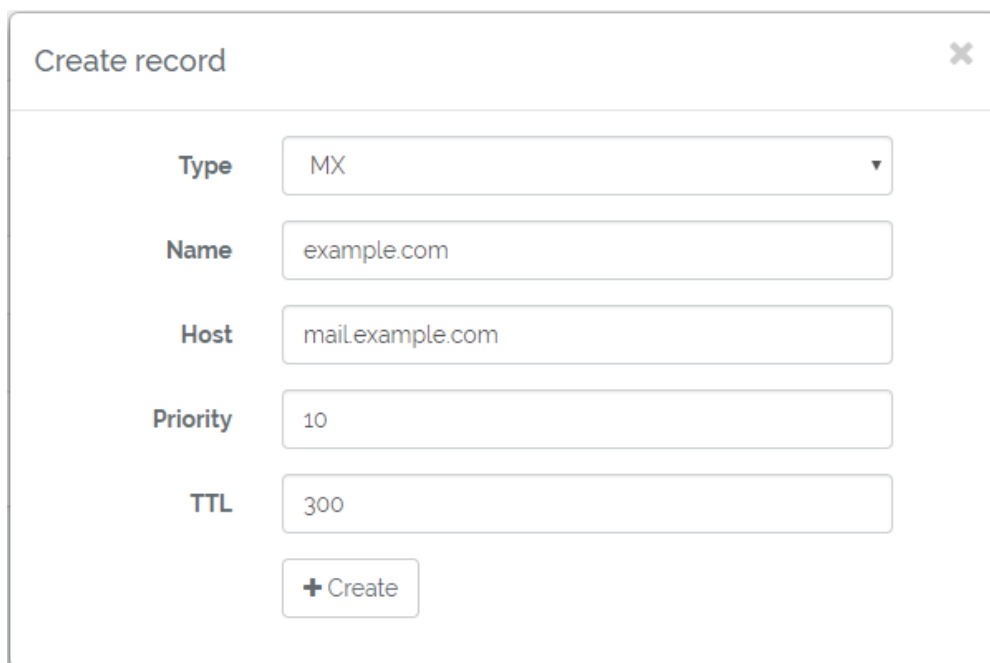
@ IN NS server.

lakin 216000 IN AAAA f797:acd6:fce7:852b:1a3e:79d6:e97f:feec
hills 270000 IN CNAME boyle.biz.
mckenzie 162000 IN CNAME becker.com.
_ldap._tcp.pagac 243000 IN SRV 50 20 47311 boyer.com.
kertzmann 297000 IN NS batz.biz.
```

Obr. č. 16: Ukážka generovaného súboru zóny

5.5.1. Správa zón a ich záznamov

Po pridaní zóny je možné kliknutím na názov upravovať záznamy. Aplikácia podporuje sedem najčastejšie vyskytovaných typov záznamov **A**, **AAAA**, **CNAME**, **MX**, **TXT**, **SRV** a **NS**. Pri vytváraní nového záznamu sa po výbere typu formulár automaticky prispôsobí.



Obr. č. 17: Formulár pre vytvorenie nového záznamu

5.5.2. Ukladanie záznamov

Každý záznam je reprezentovaný PHP triedou implementujúcou interface **DnsRecordAttributes**, ktorá obsahuje tri dôležité metódy.

fromArray – Načíta záznam z reťazca. Pri načítaní prebehne validácia údajov na základe typu záznamu.

toArray – Premení objekt reprezentujúci záznam na reťazec, ktorý je potom uložený do databázy vo formáte JSON.

toBindSyntax – Vráti reprezentáciu záznamu vo formáte podporujúci službou Bind. Pri generovaní konfiguračných súborov sa táto metóda používa na generovanie zónového súboru.

Každá z týchto metód je implementovaná na základe typu záznamu. Pri vytváraní nového záznamu je pomocou metódy **createDnsRecordAttributes** v traite **DnsRecordAssistance** vytvorený vyššie spomenutý objekt reprezentujúci záznam. Údaje o zázname sú následne exportované do reťazca a uložené do databázy vo formáte JSON. Pri načítaní záznamu sa proces vykonáva opačne. Počas preklápania z reťazca na objekt sú údaje validované.













5.5.3. Export & Import

V prípade migrácie zón aplikácia umožňuje zónový súbor exportovať do formátu podporujúci službou Bind. Proces exportovania prebieha na rovnakom princípe ako pri generovaní konfiguračných súborov s rozdielom, že zónový súbor sa ponúkne administrátorovi na stiahnutie.

Naopak je zónový súbor taktiež možné v prípade nutnosti migrácie od iného poskytovateľa importovať a uložiť do databázy.

5.6. Elektronická pošta

V tejto sekcii je možné spravovať virtuálne domény elektronickej pošty, užívateľov a ich aliasov služieb Postfix a Dovecot. Služby sú nakonfigurované automaticky pri inštalácii aplikácie tak, aby čítali všetky údaje z databázy. To znamená, že generovanie konfiguračných súborov nie je potrebné. Po vytvorení virtuálnej domény je možné spravovať jej užívateľov.


Mail users of domain turcotte.com			
Name	Login / E-Mail		
Braxton Bartell	braxton.bartell@turcotte.com		  
Clarabelle Parisian	clarabelle.parisian@turcotte.com		  
Connor Weber	connor.weber@turcotte.com		  

Obr. č. 18: Správa užívateľov


Každému užívateľovi je možné kliknutím na tlačidlo kľúča zmeniť heslo. Vedľajším tlačidlom otvoríme správu užívateľových aliasov.

User aliases


contact@example.com




info@example.com



sales@example.com



@example.com















Obr. č. 19: Správa užívateľových aliasov

5.7. Firewall

V sekcii firewall je možné spravovať pravidlá pre filtrovanie paketov. Aplikácia umožňuje zadať protokol, zdrojovú a cieľovú adresu, port, prípadne rozsah portov, pre ktoré pravidlo platí. Každé pravidlo určuje akciu, ktorá sa vykoná po splnení vyššie definovaných podmienok. Paket môže byť povolený, zahodený, odmietnutý, alebo obmedzený.

Na rozdiel od ostatných služieb, v tomto prípade sa pravidlá aplikujú cez externý softvér UFW a to spúšťaním jeho príkazov na pozadí na základe parametrov pravidiel uložených v databáze. Softvér UFW za nás vygeneruje pravidlá vo formáte iptables v súbore `/etc/ufw/user.rules`.

Rules					
Type	Protocol	Destination	Source	Port	
ALLOW	TCP	192.168.100.2	192.168.100.3	22	  
ALLOW	UDP	192.168.100.2	0.0.0.0/0	9987	  
DENY	TCP	0.0.0.0/0	10.0.0.0/8	5560 - 5570	  
REJECT	UDP	::	ff00:0042:8329::1	7777 - 8888	  

Obr. č. 18: Správa pravidiel firewallu

Pravidlá sú aplikované pomocou procesu opísaného v kapitole 5.2.1 s rozdielom, že príkazy sú spúšťané pod systémovým užívateľom root.

```

root@server:~# ufw status
Status: active

To Action From
--
192.168.100.2 22/tcp ALLOW 192.168.100.3
192.168.100.2 9987/udp ALLOW Anywhere
Anywhere DENY 10.0.0.0/8 5560:5570/tcp
:: 7777:8888/udp REJECT ff00:42:8329::1

```

Obr. č. 19: Zobrazenie konfigurácie cez UFW

5.8. Úlohy

V sekcii pre správu úloh je možné definovať vykonávanie úloh v definovaných intervaloch. Každý užívateľ má svoj konfiguračný súbor pre spúšťanie úloh nachádzajúci sa v priečinku **/var/spool/cron/crontabs**. V každom konfiguračnom súbore, ktorý nesie názov systémového užívateľa sa nachádzajú úlohy oddelené novým riadkom. Každý riadok obsahuje interval vykonania príkazu a samotný príkaz.

```
* 0,19,4-11 * 5/5 * groups root
* * * * * id
* * * 1,3-9,9-10 * ping -c 4 127.0.0.1
```

Obr. č. 20: Konfiguračný súbor služby Cron

6. API

Správa serveru je možná aj prostredníctvom externých nástrojov bez využitia grafického rozhrania dodávaného s týmto projektom. Príkazom **php artisan route:list** je možné zobrazíť všetky existujúce volania.

Aby bolo možné API volania úspešne vykonať, klient sa musí aplikácii autorizovať pomocou súboru cookie s aktívnou reláciou.

7. Inštalácia

Inštalácia sa vykonáva spustením BASH skriptu s názvom `install.sh`. Skript postupne nainštaluje všetky potrebné balíčky služieb. Následne skopíruje predkonfigurované konfiguračné súbory na príslušné miesta. Databáza a jej tabuľky sú vytvorené migračným príkazom **php artisan migrate**. Ako posledné sa nás inštalačný skript spýta na heslo pre administrátora, ktorý sa bude môcť do aplikácie prihlásiť.

8. Cenová kalkulácia nákladov

Kompletné vyhotovenie projektu mi bez teoretickej časti zabralo približne 120 hodín. Príprava projektu, resp. návrh mi zabral približne 30 hodín. Pri zvažovaní hodinovej mzdy som bral do úvahy nevyhnutnosť pokročilejších znalostí s niekoľkými spravovanými službami a technológiami. Hodinovú mzdu za vyhotovenie projektu som ohodnotil na 15 EUR. Keďže projekt bol navrhnutý pre správu čisto softvéru s otvoreným zdrojom, resp. bezplatného softvéru, licencie na spravované služby nie sú potrebné.

Objednávka

DODÁVATEĽ

Richard Boldiš

Martina Granca 9
841 02 Bratislava

Telefón: +421 944 559 704

E-mail: richard.boldis@protonmail.com

ODBERATEĽ

Stredná priemyselná škola elektrotechnická

Karola Adlera 5
841 04 Bratislava

IČO: 17319161

Dátum vystavenia

12.04.2018

Dátum platnosti

31.04.2018

Názov položky	Počet	MJ	Jedn. cena	Cena Spolu
Návrh a realizácia	140	hod.	14,00	1 960,00
Inštalácia	3	ks.	40,00	120,00
Školenie	1	hod.	150,00	150,00
Celkom				2 230,00 €

Navrhnete a zrealizujete webové administračné rozhranie pre správu serveru s funkcionalitou spravovať webové stránky, databázu, systém doménových názvov, pravidiel firewallu a úloh. Hotovú aplikáciu je potrebné nainštalovať na tri servery s OS Ubuntu 16.04. Je potrebné zaškoliť skupinu našich administrátorov.

Faktúra **2018041301**

DODÁVATEĽ

Richard Boldiš

Martina Granca 9

841 02 Bratislava

Telefón: +421 944 559 704

E-mail: richard.boldis@protonmail.com

ODBERATEĽ

Stredná priemyselná škola elektrotechnická

Karola Adlera 5

841 04 Bratislava

IČO: 17319161

Dátum vystavenia

13/04/2018

Dátum dodania

13/04/2018

Dátum splatnosti

27/04/2018

Číslo účtu **2935145247/1100**

SWIFT

VS

Suma k úhrade

IBAN **SK80 1100 0000 0029 3514 5247**

TATRSKBX

2018041301

2,440.00 €

Názov položky	Počet	MJ	Jedn. cena	Cena Spolu
Návrh	30	hod.	15.00	450.00
Realizácia	120	hod.	15.00	1,800.00
Inštalácia	3	ks.	50.00	150.00
Školenie	2	hod.	20.00	40.00

Celkom 2,440.00 €



Podpis

Záver

Cieľom tejto práce bolo vysvetliť čitateľovi zámer projektu a oboznámiť ho s funkcionalitou jednotlivých sekcií pre správu serveru. Aplikácia je pripravená na prevádzku v produkčnom prostredí na domácich alebo stredne menších firemných serveroch.

Počas programovania aplikácie som sa stretol s niekoľkými problémami rôzneho druhu od konfiguračných po nastavenia operačného systému, ktorých vyriešením som nadobudol nové skúsenosti z oblasti správy operačného systému Ubuntu a programovania všeobecne.

Zoznam použitej internetovej dokumentácie

<https://laravel.com/docs/5.6>

<https://vuejs.org/v2/guide/>

<https://github.com/axios/axios/blob/master/README.md>

<https://nginx.org/en/docs/>

<https://www.isc.org/downloads/bind/doc/>

<http://www.postfix.org/documentation.html>

<https://wiki.dovecot.org/ConfigFile>

<https://help.ubuntu.com/community/UFW>

<https://help.ubuntu.com/community/CronHowto>

<https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>

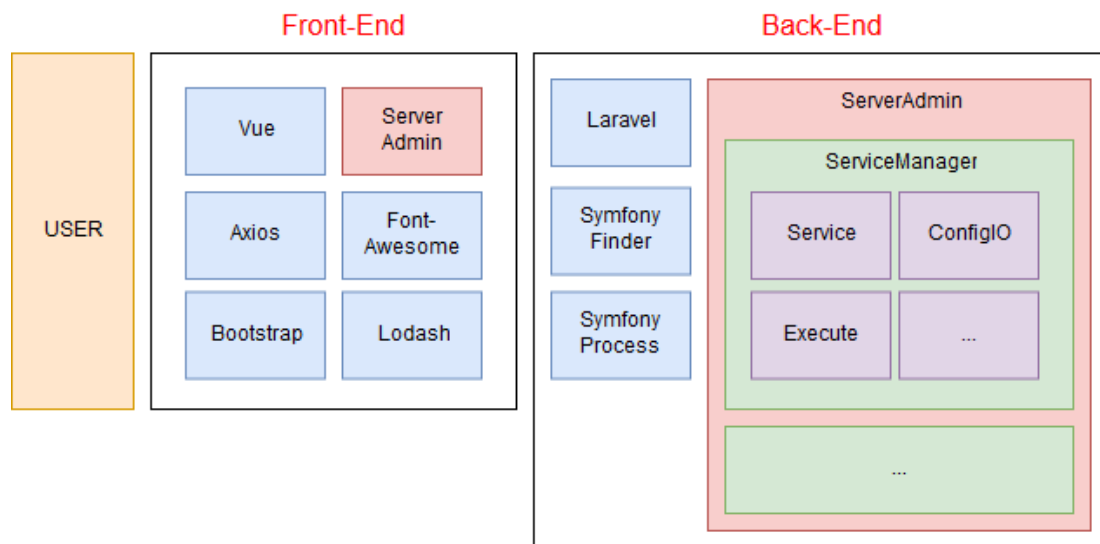
<http://www.admin-magazine.com/Articles/MariaDB-vs.-MySQL>

ZOZNAM PRÍLOH

A: Schéma aplikácie

B: Schéma databázy

PRÍLOHA A



PRÍLOHA B

