Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Informační systém pro školící středisko

2018

Libor Machálek

Vedoucí práce: Mgr. Martin Trnečka, Ph.D.

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Libor Machálek
Název práce:	Informační systém pro školící středisko
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2018
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	Mgr. Martin Trnečka, Ph.D.
Počet stran:	30
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliograhic info

Author:	Libor Machálek
Title:	Information system for training center
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2018
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Martin Trnečka, Ph.D.
Page count:	30
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Cílem této bakalářské práce je vytvoření informačního systému pro podporu a evidenci prací zaměstnanců školícího střediska. Každý zaměstnanec bude mít svůj vlastní účet. Po přihlášení bude moci zadávat nová či již proběhlá školení, popřípadě podpůrné materiály (např. prezenční listina). Každý zaměstnanec bude přiřazen do konkrétních sekcí, které budou odpovídat jejich působnosti. Součástí aplikace bude i upozornění na opakování školení, hromadná e-mailová korespondence, exporty, evidence školených zařízení. Informační systém bude realizován formou webové (responzivní) aplikace za použití PHP a MySQL databáze.

Synopsis

The aim of this bachelor thesis is to create an information system for the support and registration of the work of the employees. Every employee will have his/her own account. Upon logging in, it will be possible to enter new or already undergoing training, if appropriate supporting materials (e.g. attendance list). Each employee will be assigned to specific sections that will respond their scope. The application will include alerts for repeat training, email correspondence, exports, tuition facilities. The information system will be implemented in the form of web (responsive) applications using PHP and MySQL database.

Klíčová slova: databáze; CodeIgniter; informační systém; webová aplikace

Keywords: database; Codeigniter; information system; web application

Chtěl bych poděkovat vedoucímu práce Mgr. Martinu Trnečkovi, Ph.D. za jeho cenné rady a návrhy, které pomohly tuto práci zlepšit.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam zdrojových kódů

1 Úvod

K tématu mé bakalářské práce jsem se dostal tak, že jsem byl požádán majitelem (dále jen zadavatel) firmy BT servis[1] o vytvoření webové aplikace pro jeho společnost. Tato společnost se zabývá školením firem v oblasti bezpečnosti a ochrany zdraví při práci, ekologie a požární ochrany. Provádí také mnoho dalších, ale tato bakalářská práce se věnuje informačnímu systému, ne školením.

Každý zaměstnanec společnosti (dále jen školitel) si vedl o provedených školeních záznamy, které měly pokaždé jinou formu. Nápad zadavatele zajišťující zvýšení efektivity práce byl jednoduchý. Sjednotit formu, ve které zaměstnanci odevzdávají svou práci. Jeden z nápadů byl vytvořit webovou aplikaci, ke které by měl každý zaměstnanec přístup a ukládal zde výsledky své práce. Daná aplikace by ulehčila práci jak majiteli, tak i samotným zaměstnancům.

Druhá část této práce pojednává o návrhu aplikace a věcmi s tím spojenými. V této kapitole se dočteme o designu celého informačního systému, uživatelských oprávněních a v neposlední řadě také o návrhu databáze. Třetí kapitola se zabývá implementací informačního systému. Budou zde představeny zvolené technologie a odůvodnění, které vedly zrovna k nim. Poslední část je uživatelská dokumentace. V ní popíšeme uživatelské rozhraní a ovládání celého systému.

1.1 Popis aplikace

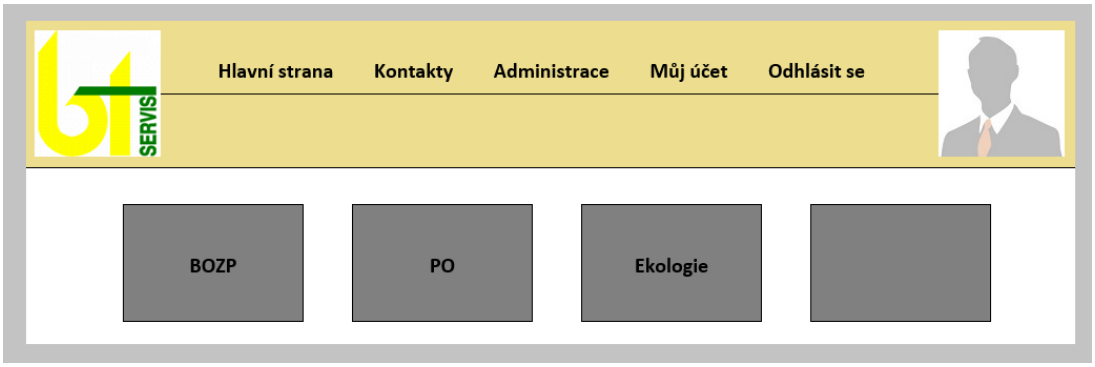
Jak již bylo řečeno aplikace má mít formu responzivní webové aplikace, kde bude možné evidovat práci zaměstnanců společnosti BT servis. Na aplikaci jsou kladeny následující požadavky:

- Každý zaměstnanec má svůj vlastní úče
- Přihlášený zaměstnanec bude moci zadávat, upravovat a mazat školení v sekci, kterou má přidělenou administrátorem
- Generování prezenčních listin
- Ke každému školení lze přidat prezenční listinu
- Upozornění na přeškolení již evidovaných školení
- Hromadná emailová korespondence
- Správa uživatelských oprávnění
- Evidence kontaktů na firmy

[1] http://www.ebtservis.cz/

2 Návrh aplikace

Při návrhu aplikace jsem se řídil požadavky zadavatele, z nichž jeden byl jednoduchý design viz obrázek 1. Na radu vedoucího práce bude vzhled částečně inspirován šablonou Metronic[2] a také z druhé části budeme brát v potaz přání zadavatele, jenž si přál jednotlivé sekce zobrazit jako dlaždice, které by měli být umístěny na hlavní stránce tak, aby nebylo možné si je s ničím jiným splést. Stránka kontaktů je zamýšlena se společnou uživatelskou a administrátorskou částí jak je zobrazeno na obrázku 2.



Obrázek 1: Návrh vzhledu hlavní strany

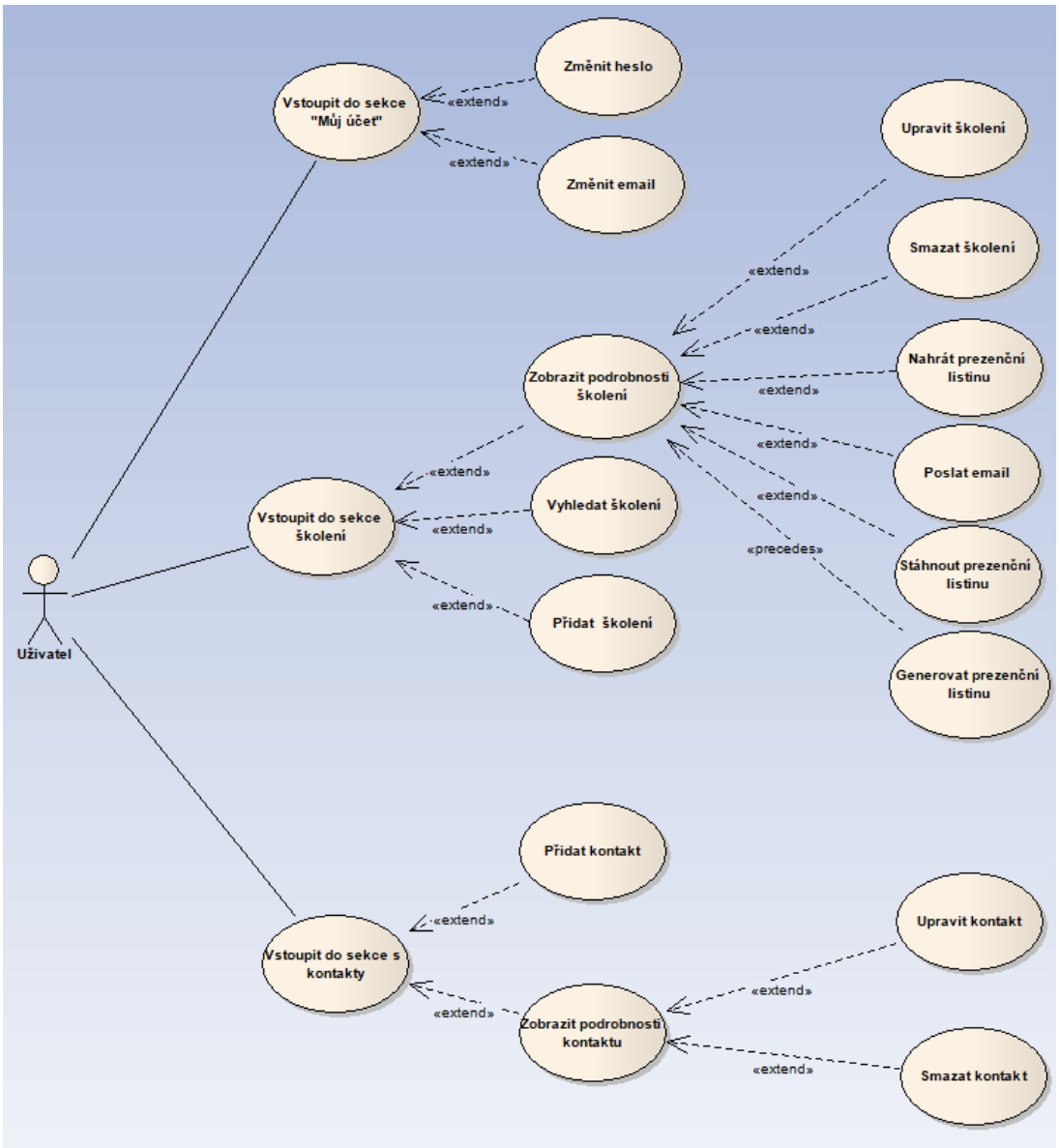
Obrázek 2: Návrh vzhledu kontaktů

[2] https://keenthemes.com/metronic/

2.1 Uživatelská a přístupová práva

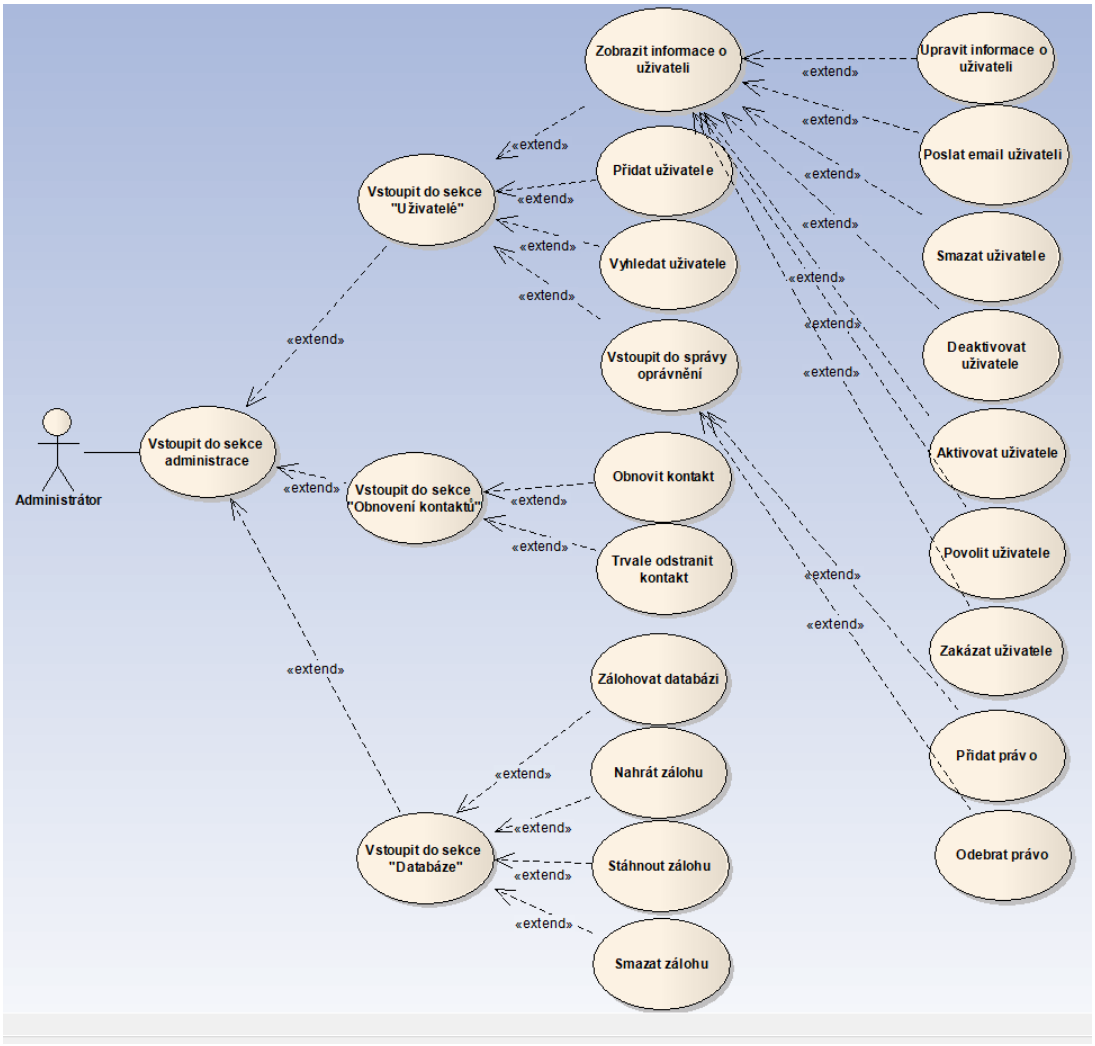
Systém byl navržen jako víceuživatelský. Na požadavek zadavatele umožňuje přidělení uživatelských práv, která jsou rozdělena na dva typy.

Prvním z nich je „uživatel“, jenž má omezené pravomoce. Toto oprávnění má přiděleno každý školitel. Přístupová práva jednotlivých uživatelů se liší, podle sekce, kterou jim nastaví administrátor ke správě. Speciální sekcí jsou „Kontakty“, ke kterým může přistupovat každý uživatel viz diagram případů užití na obrázku 3.



Obrázek 3: Přístupová oprávnění uživatele.

Dalším z uživatelských oprávnění je „administrátor“. Ten může být jen jeden. Administrátora představuje zadavatel, který může v aplikaci provádět veškeré činnosti a úkony, které informační systém jako celek poskytuje. Jinými slovy má administrátor přístup ke stejným sekcím jako běžný uživatel a navíc k sekci „Administrace.“ Diagram případů užití na obrázku 4 zobrazuje přístupová práva k této sekci.

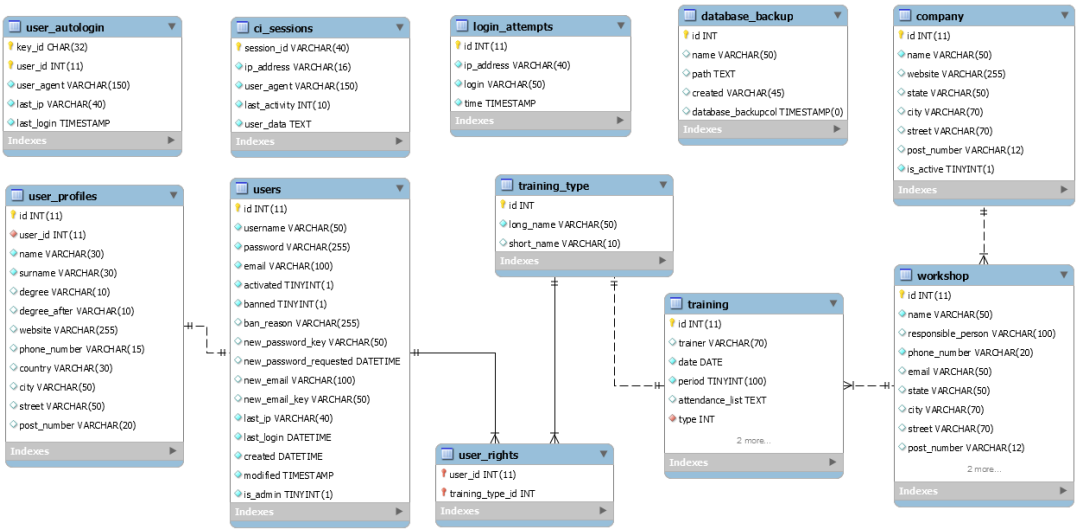


Obrázek 4: Přístupová oprávnění administrátora k sekci „Administrace“.

2.2 Databáze

Databáze popisovaného informačního systému není nijak komplikovaná. Obsahuje jedenáct tabulek. Z nichž pět obsluhuje knihovna Tank Auth, o které ještě bude řeč později v podkapitole 3.1.2. Jsou to tabulky: „ci_sessions, login_attempts, user_autologin, users a user_profiles“. Zbývající tabulky jsou již navrženy auto-

rem práce pro nejefektivnější chod aplikace. Jedná se o tabulky „company, database_backup, training, training_type, user_rights a workshop.“ Názvy tabulek byly voleny tak, aby co nepřesněji popisovaly, jaká data uchovávají. ER-diagram na obrázku 5 ukazuje všechny tabulky s jejich sloupci a v neposlední řadě relace mezi těmito tabulkami.



Obrázek 5: ER-diagram databáze popisovaného informačního systému.

3 Implementační část

3.1 Volba technologií

Důležitou částí při tvorbě jakékoliv aplikace je volba technologií. Zvolíme-li nevhodný programovací jazyk, výsledný program nemusí mít požadované vlastnosti jako je například rychlost nebo plynulost běhu. Pro vytvoření této aplikace byly zvoleny následující technologie.

3.1.1 CodeIgniter v.3.1.8.

CodeIgniter [1] je open-source PHP framework založený na návrhovém vzoru MVC, ale framework nenutí vývojáře k jeho dodržování. CodeIgniter je šířen pod licencí MIT. Obecně řečeno, hlavní úkol frameworků je ušetřit práci vývojářům. Poskytují spoustu již hotových funkcí a knihoven, které by se mohly vývojářům hodit. Všechny funkce a knihovny lze najít v přehledné a dobře čitelné dokumentaci buď na adrese: https://codeigniter.com/user_guide/ nebo při stažení frameworku v adresáři „user_guide“.

Návrhový vzor MVC je často využívaná metoda při programování, kdy je kód aplikace rozdělen na tři části, které se nazývají model, view a controller. Model má na starost komunikaci s databází. View nebo jinak řečeno pohled obstarává zobrazení dat uživateli. Poslední částí je controller. Ten zajišťuje propojení view s daty, která jsou mu poskytnuta modelem.

Pro stažení archivu s frameworkem CodeIgniter z webu je potřeba 2 – 3MB volného místa. Tento archiv obsahuje i dokumentaci, která je velmi obsáhlá a srozumitelná. Další bezespornou výhodou je bezpečnost, na kterou CodeIgniter dbá. Obsahuje spoustu funkcionalit pro její zajištění. Výše vypsané výhody jsou také důvody, proč jsem si framework zvolil.

3.1.2 Knihovna Tank Auth v.1.0.9.

Jedná se o autentizační knihovnu založenou na DX Auth. Tank Auth [9] používá pro hashování hesel knihovnu phpass. Obsahuje též spoustu funkcí, které jsou defaultně zapnuty, ale lze je vypnout. Jako příklad takové funkce uvedeme registraci tzn. vytváření účtů samotnými uživateli. Dále pak ověření pomocí CAPTCHA při registraci, anebo při překročení daného počtu pokusů o přihlášení. Vybral jsem tuto knihovnu díky její jednoduché instalaci a množství funkcí, které nabízí.

3.1.3 TCPDF

Projekt open-source knihovny TCPDF [10] vznikl v roce 2002 a od té doby slouží ke generování PDF souborů. Pomocí jazyka PHP a funkcí z TCPDF vytvoříme šablonu. Z této šablony je následně vygenerován PDF dokument. Je zde i několik možností, jestli chceme dokument rovnou stáhnout nebo jen zobrazit a mnoho

dalších. TCPDF jsem si vybral, protože mi přijde lepší než dosud mnou využívaná knihovna mPDF.

3.1.4 MySQL

Je databázový systém švédského původu, ale nyní je pod záštitou Oracle. Jazyk použitý k dotazování na databázi, je založen na SQL (Structured Query Language) [2], což je dotazovací jazyk, používaný pro práci s daty v relačních databázích. MySQL byla jasná volba od začátku, protože bývá často součástí balíčku LAMP, je multiplatformní a mám s ním spoustu zkušeností. Další informace viz [3].

3.1.5 Jazyk HTML 5

HTML (Hypertext Markup Language) [4] je značkovací jazyk, který byl navržen pro tvorbu webových stránek. Syntaxe jazyka je velmi jednoduchá a rychle pochopitelná. V dnešní době je v něm napsána většina webů a považuje se za základ při tvorbě webových stránek a aplikací. Slouží pro definici významu (sémantiky) obsahu webové stránky. Aktuální verze HTML je 5.2, která je standardizována již 9 měsíců. V roce 2017 měla status recommendation[3].

3.1.6 CSS

Kaskádové styly (Cascading Style Sheets) jsou jazykem, který popisuje styl HTML dokumentu viz [5]. S jejich pomocí definujeme výsledný vzhled aplikace. Často se používají ve spojení s jazykem HTML. Syntaxe jazyka je velmi jednoduchá. CSS obsahuje pravidla, jejichž částí je selektor, pomocí kterého vybereme HTML element, dále pak sada definic. Když je element vybrán, nastavujeme mu předdefinované vlastnosti a tím upravíme jeho design, pozici a popř. mu nastavíme nějaké efekty. Bez této technologie bychom nebyli schopni definovat design popisovaného informačního systému, proto není třeba odůvodňovat jeho zvolení.

3.1.7 JavaScript a technologie s ním spojené

JavaScript [6] je objektově orientovaný skriptovací jazyk, který je dynamicky typovaný. Obsahuje však i funkcionální prvky. Jedná se o plnohodnotný jazyk, jenž je implementací ECMAScriptu. Na webu se používá pro „oživení“ stránky. Tento jazyk je často využíván jako nedílná součást plnohodnotných informačních systémů.

3.1.7.1 HTML DOM a DOM

Pro vývoj webových stránek nebo aplikací se často využívá HTML DOM (Document Object Model). Funguje v návaznosti na HTML stránku, která je po načtení dostupná přes objekt JavaScriptu

[3] https://www.w3.org/TR/html52/

„document“. DOM je standard konsorcia W3C[4], definující dokument (webovou stránku) jako strom.

3.1.7.2 Knihovna jQuery

jQuery [7] je JavaScriptová knihovna pod licencí MIT, která je hojně využívána pro tvorbu interaktivního webu. Výrazně zlepšuje práci a manipulaci s obsahem stránky a také mnoho dalšího, například použití technologie AJAX.

3.1.7.3 AJAX

Tato „technologie“ není tak úplně technologií, jde spíš jen o použití již existujících technologií viz [8]. Zkratka AJAX znamená asynchronous JavaScript and XML a jak již název vypovídá jde o použití technologií: JavaScript, XML a objekt „XMLHttpRequest“. Použití těchto technologií nám zaručí výměnu dat se serverem a změnu určité části stránky bez nutnosti načíst celou stránku.

3.2 Instalace zvolených technologií

Jelikož jedním z kritérií při volbě frameworku či knihovny byla obtížnost instalace, není těžké všechny technologie nainstalovat. Některé už nainstalovány máme, například JavaScript je implementován skoro v každém webovém prohlížeči. Nyní si vysvětlíme instalaci některých technologií.

3.2.1 CodeIgniter

Instalace frameworku CodeIgniter je triviální. Předpokládejme, že máme staženou stabilní verzi frameworku. Celý postup instalace[5] lze shrnout do tří následujících kroků:

1. Nahrajeme rozbalenou složku s frameworkem pod názvem „codeigniter“ do kořenového adresáře na webovém serveru.

2. Jako další krok otevřeme soubor „config.php“ umístěný v adresáři „codeigniter/application/config“ a změníme následující řádek: `$config['base_url'] = '';` za řádek `$config['base_url'] = 'https://example.com/codeigniter/';` a uložíme změny.

3. Pokud budeme chtít používat databázi, ve stejném adresáři tzn. „codeigniter/application/config“ se nachází soubor „database.php“, kde vyplníme informace pro připojení k databázi. Obvykle jsou to položky „username, password a database“ pro název databáze.

To je vše co potřebujeme ke zprovoznění. Nyní budeme pracovat s adresářem „application“ a jeho podsložkami „models, views a controllers“, do kterých budeme umísťovat soubory se zdrojovými kódy.

[4] https://www.w3.org/
[5] https://codeigniter.com/user_guide/installation/index.html

3.2.2 Databáze

Máme-li nainstalovaný některý z balíků LAMP, MySQL je zde již jednou z jeho součástí. Stačí se jen připojit s nějakému systému pro správu databáze (např. phpMyAdmin[6] nebo Adminer[7]). Vytvořit novou databázi a importovat SQL skript, který se postará o vytvoření tabulek, klíčů a dalších náležitostí.

3.2.3 Tank Auth

Po rozbalení archivu s knihovnou, dostaneme SQL skript s názvem „schema.sql“ a dva adresáře – „application“ a „captcha“. Složku „application“ sloučíme s adresářem „application“ ve frameworku, který už máme na webovém serveru. Adresář „captcha“ pak nahrajeme na webový server na úroveň adresáře „application“. SQL skript importujeme do databáze, která je připojená k frameworku.

3.3 Struktura aplikace

Předpokládejme, že máme správně nainstalovaný framework. V této podkapitole si vysvětlíme některé jeho důležité části. Z adresáře „application“ vyzdvihneme složky „config, controllers, language, libraries, models, third party a views“. Nejdůležitějšími částmi jsou nepatrně adresáře „controllers, models a views“. Zde se ukládají soubory se zdrojovými kódy, ale ostatní adresáře jsou neméně užitečné. Složka „config“ obsahuje konfigurační soubory. Do adresáře „language“ si můžeme nahrát soubory s překlady různých hlášek, jimiž nás CodeIgniter informuje o chybách. Popisovaný systém má plnou podporu hlášek v českém jazyce [11]. Zbývá nám už jen „libraries a third_party“ – tyto složky jsou obsahem celkem podobné, ale rozdíly tu jsou. Nejmarkantnějším rozdílem je fakt, že do „libraries“ bychom měli ukládat knihovny, které si sami napíšeme. Do složky „third_party“ mohou také přijít knihovny nebo jakékoliv soubory, ale pouze třetích stran. Nebývá to však striktně dodržováno.

3.3.1 Adresáře „models“ a „controllers“

Do adresáře „models“ ukládáme modely. Jsou to PHP třídy, které mají za úkol komunikovat s databází. To lze dělat buď přímo pomocí SQL, anebo pomocí Query Builder Class, která v některých případech dokáže zkracovat dotazy či další užitečné věci. Zmíněná třída byla použita při programování popisované aplikace. V adresáři „system/core“ je uložena třída „CI_Model“, kterou musí dědit každý model vytvořený vývojářem.

Další třída, jež je uložena v „system/core“ je „CI_Controller“, kterou musí pro změnu dědit každý controller. Controllery slouží k propojení dat získaných z databáze s pohledem neboli view. Název funkce, vytvořené v controlleru slouží jako část URL adresy. Uvažujme, že máme controller s názvem „Foo“ a jeho

[6] https://www.phpmyadmin.net/
[7] https://www.adminer.org/cs/

funkci „bar“, volaná adresa pro zobrazení výsledku bude „example.com/foo/bar“. Ze zdrojového kódu 1 funkce „select“ controlleru „Contacts“ vidíme syntaxi získávání dat z modelu, následné uložení do pole „$data“ a poslání získaných dat do view „selection.php“, které je umístěno v adresáři „views/contacts“

```
1 function select() {
2   $id_current_user = $this->tank_auth->get_user_id();
3
4   $data['user_id'] = $id_current_user;
5   $data['username'] = $this->tank_auth->get_username();
6   $data['is_admin'] = $this->administration_model->get_admin_right(
        $id_current_user);
7   $data['user_rights']  = $this->trainings_model->get_access_rights(
        $id_current_user);
8
9   $this->load->view('/contacts/selection', $data);
```

Zdrojový kód 1: Funkce „select()“ v controlleru „Contacts“.

3.3.2 Adresář „views“

Posledním z nejdůležitějších adresářů je „views“. V něm jsou uloženy šablony s HTML kódem, ze kterých se po doplnění dat přicházejících od controlleru stane finální stránka. Existuje zde možnost používat HTML helper. Ten dokáže v určitých situacích opět ulehčit práci, stejně jako v případě Query Builder Class. HTML helper jsem však při vývoji informačního systému nevyužil kvůli obavám, že výsledný vygenerovaný kód nebude validní.

3.3.3 Knihovna „Trainings“

Tato knihovna funguje v podstatě jako controller zajišťující propojení modelů a view jednotlivých sekcí se školeními. V adresáři „application/controllers“ jsou umístěny controllery, které volají funkce této knihovny.

Důvody tohoto řešení jsou prosté. Jak už bylo výše zmíněno, URL adresa se skládá z názvu controlleru a nějaké jeho funkce, aby bylo patrné v jaké sekci se uživatel nachází byly vytořeny controllery s názvy sekcí. Dalším důvodem je redundance kódu, ke které by docházelo, kdyby jednotlivé controllery nevolaly funkce zmiňované knihovny, ale každá by měla vlastní velmi podobný kód s drobnými změnami.

3.3.4 Vlastní obecný controller „AUTH_Controller“

CodeIgniter umožňuje definovat si třídy, které dědí systémovou třídu. Tyto třídy pak umísťujeme složky application/core. Daná funkcionalita je poměrně mocný nástroj. V případě popisované aplikace byla tato možnost využita pro vytvoření

obecného controlleru „AUTH_Controller“. Ten dědí systémový „CI_Controller“ a přidává k němu oveření, zda-li je uživatel přihlášen, jak je vidět na zdrojovému kódu 2. Toto řešení je elegantní z důvodu odstranění redundance kódu.

```
1 class AUTH_Controller extends CI\_Controller {
2   function __construct() {
3     parent::__construct();
4     if(!$this->tank_auth->is_logged_in()) {
5       redirect('auth/login');
6       exit;
7     }
8   }
9 }
```

Zdrojový kód 2: Vlastní třída obecného controlleru.

3.3.5 Knihovna TCPDF

Pro generování prezenčních listin byla použita knihovna TCPDF. Knihovna je umístěna v adresáři „application/libraries“. Zároveň s ní je tu také soubor s třídou, pomocí které se volají funkce knihovny.

Instalace knihovny je naprosto jednoduchá. V podstatě se jedná o rozbalení archivu, jeho umístění do adresáře „application/libraries/tcpdf“ a na stejné úrovní je třeba vytvořit třídu viz zdrojový kód 3, která je pak volána při vytváření nové šablony.

```
1 require_once dirname(__FILE__) . '/tcpdf/tcpdf.php';
2
3 class Pdf extends TCPDF {
4   function __construct(){
5     parent::__construct();
6   }
7 }
```

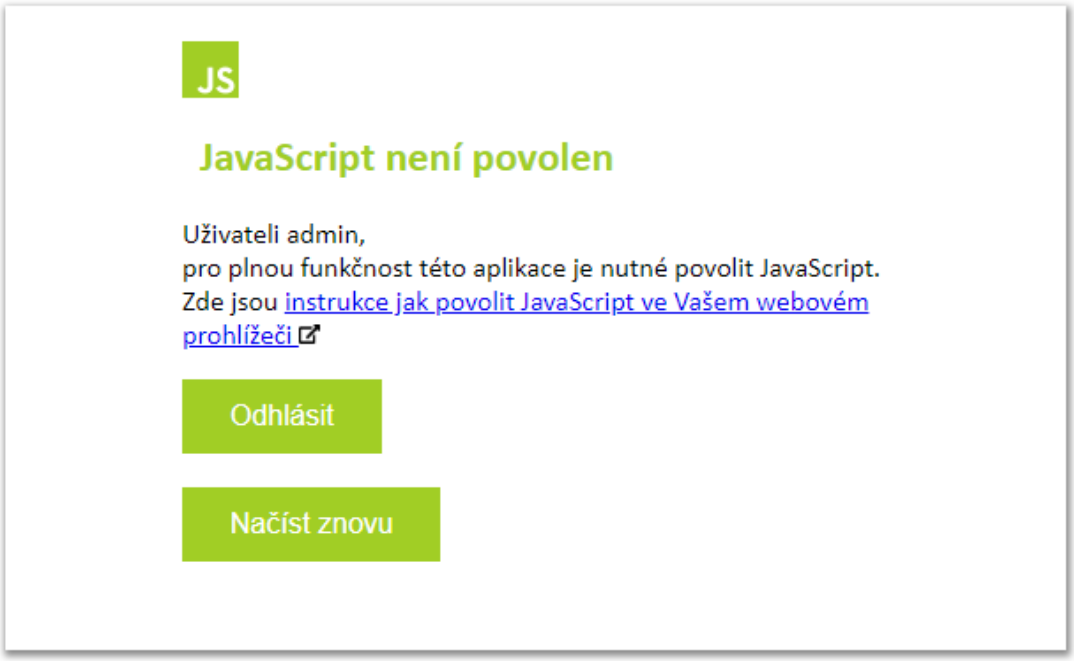
Zdrojový kód 3: Třída „Pdf“ potřebná pro volání funkcí knihovny.

Triviální použití knihovny spočívá v tom, že postupným voláním funkcí TCPDF vytváříme šablonu, z níž bude generován výsledný PDF dokument. Knihovna obsahuje spoustu funkcí, se kterými se dá dělat neskutečné množství šablon. Knihovna umožňuje nastavení autora, titulku, hlaviček, patiček a mnoho dalších. Jedna z možností TCPDF je generovat obsah z HTML kódu. S touto možností se také pojí stylovaní elementů pomocí kaskádových stylů. Pro vygenerování dokumentu knihovna nabízí funkci „Output“ se dvěma parametry. Prvním z nich je název dokumentu a druhým je obvykle přepínač, říkající co se má stát

s dokumentem po vygenerování. Zadáme-li parametr „D“ dokument bude stažen, což je použito v popisované aplikaci. Dalšími variantami parametru jsou například „I“ pro otevření přímo v prohlížeči, „F“ pro uložení na server, „S“ pro návrat dokumentu jako textového řetězce a další.

3.4 Detekce vypnutého JavaScriptu

Někteří opatrnější uživatelé mají pro jistotu vypnutý JavaScript. Argumentují mnoha důvody, ale ochuzují se tak o určitou funkcionalitu webu. Informační systémy obecně hodně využívají JavaScript. Proto je dobré na takové uživatele systém připravit. Popisovaná aplikace se brání před uživateli s vypnutým JavaScriptem tak, že jim vůbec nedovolí přistupovat k aplikaci, dokud si jazyk v prohlížeči nezapnou. Jak jde vyčíst z obrázku 6, aplikace uživatele varuje dialogem s informacemi jak zapnout JavaScript. Dále pak dialog obsahuje dvě tlačítka. První z nich slouží k odhlášení uživatele a druhé obnoví stránku.



Obrázek 6: Dialog při vypnutém JavaScriptu.

3.5 Bezpečnost

3.5.1 Validace formulářů

Každá webová aplikace potřebuje formuláře, aby byla nějakým způsobem schopna od uživatele získat data. Avšak ne z každého formuláře lze poznat jaká data do daného vstupního pole má uživatel vložit či napsat. Tento problém částečně řeší

HTML 5. Máme-li například ve formuláři vstupní pole, od kterého se očekává, že bude obsahovat pouze čísla můžeme použít vstupní pole „s datovým typem“ number.

Nejedná se však o dostatečnou validaci, protože uživatel, potažmo útočník, může lehce ve zdrojovém kódu načtené stránky s formulářem změnit „type“ na typ text, kde může vložit čísla, ale klidně i text, což povede k chybě. Vzniklou skutečnost už nijak neošetříme na straně uživatele. Budeme muset udělat validaci formuláře na straně serveru.

Při tvorbě frameworku s tímto předem počítali, a tak CodeIgniter integruje knihovnu Form Validation[8] pro validaci formulářů. Zvolíme si pole a povolíme mu jaké hodnoty vstupní pole může obsahovat, například že zadaná hodnota nesmí být prázdná a musí to být číslo. Tak ošetříme na serveru špatný uživatelský vstup.

3.5.2 XSS

XSS (Cross-site scripting) je typ útoku na webové stránky nebo aplikace, kdy se útočník snaží zneužít neošetřených vstupů ve formulářích a může tak na stránku umístit škodlivý kód. To může značně poškodit webovou stránku nebo aplikaci.

Obrázek 7: Pokus o Cross-site scripting útok na pole „název firmy“ ošetřené knihovnou Security.

V CodeIgniteru pro tento případ existuje knihovna Security Class[9], která ošetřuje vstupy před XSS útokem. Na obrázku 7 můžeme vidět knihovnou ošetřený vstup. Popisovaná webová aplikace aktivně využívá tuto knihovnu u veškerých formulářů, kde existuje poteciální možnost XSS útoku. Alternativa, jak se bránit tomuto útoku je použít rovnou PHP funkci htmlspecialchars[10], která

[8] https://www.codeigniter.com/userguide3/libraries/form_validation.html
[9] https://codeigniter.com/user_guide/libraries/security.html
[10] http://php.net/manual/en/function.htmlspecialchars.php

převede speciální znaky jako jsou lomené závorky na HTML entity a tím zamezí XSS útoku.

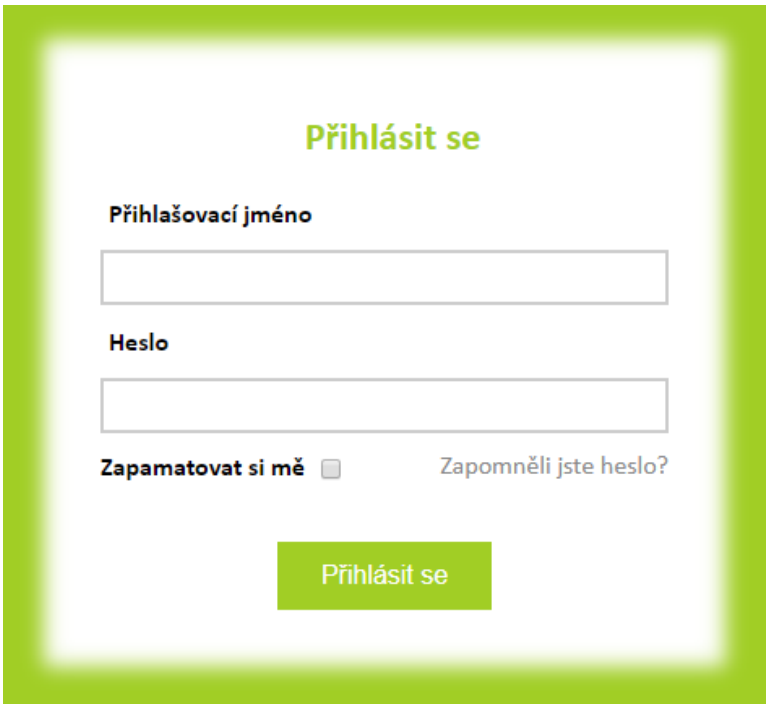
3.5.3 SQL Injection

Dalším častým typem útoku je SQL Injection. Postup je tentokrát takový, že útočník zadá vstup s SQL kódem, který může způsobit v databázi nevratné škody. Nemáme-li ošetřený vstup, aplikace spustí škodlivý kód. Řešení tohoto problému spočívá v „escapování“ nebezpečných znaků, typicky se jedná o apostrof. CodeIgniter tento problém řeší, používáme-li Query Builder Class. Tato knihovna, potažmo třída, při konstrukci dotazu automaticky „escapuje“ nebezpečné znaky.

4 Uživatelská část

4.1 Autentizace

Kdokoliv se dostane s popisované aplikaci, uvidí jako první přihlašovací formulář. Nepřihlášený uživatel se nemůže do aplikace dostat. Vzhled přihlašovacího okna je barevně sladěný se zbytkem aplikace. Z obrázku 8 lze vidět, že přihlašovací formulář obsahuje zaškrtávací pole pro zapamatování údajů k přihlášení, dvě pole – přihlašovací jméno a heslo. Dále se zde vyskytuje odkaz na formulář pro zapomenuté heslo.



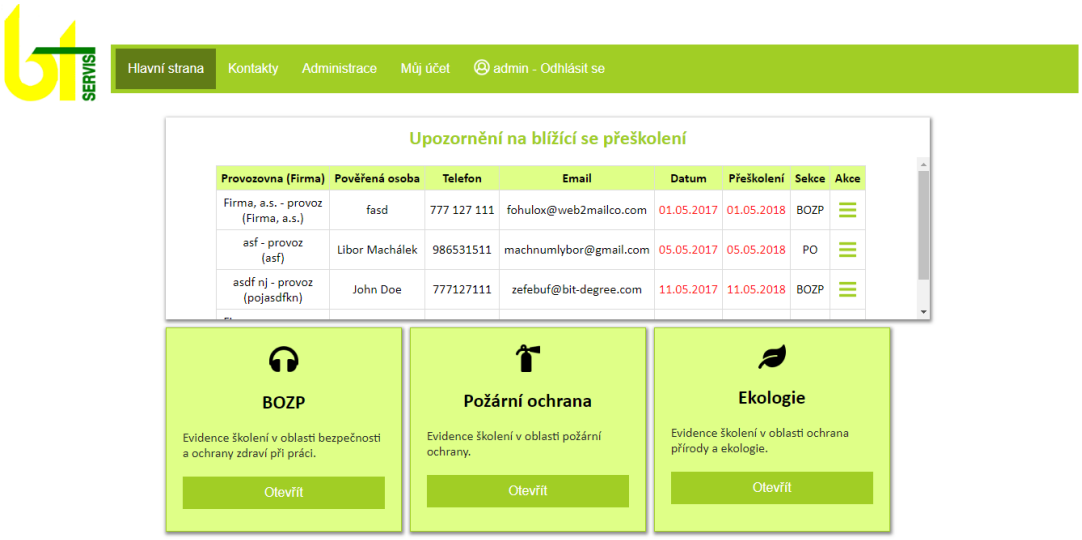
Obrázek 8: Přihlašovací fomulář.

Stejný design mají také formuláře pro změnu hesla nebo emailu. Při změně emailu je potřeba zadat heslo a novou emailovou adresu, na kterou přijde potvrzovací email. Po kliknutí na odkaz v emailu budeme přesměrování na přihlašovací formulář. Pokud se přihlásíme, email již bude změněn.

Při změně hesla je třeba do formuláře vložit staré heslo a dvakrát nové heslo. Po správném vyplnění a odeslání formuláře budete přesměrování na hlášku o správném provedení akce. Jediné, co potřebujete udělat je kliknout na šipku zpět na hlavní stranu. Budete přesměrování na stránku se školeními a po odhlášení už se budete přihlašovat novým heslem.

4.2 Vzhled aplikace

Pro design aplikace byla zvolena jako hlavní barva světle zelená na bílém pozadí. Sekce jsou prezentovány jako „dlaždice," což odpovídá jednomu z požadavků zadavatele. Text v celé aplikaci je zobrazen v bezpatkovém fontu Calibri, protože v aplikaci se nevyskytují texty většího rozsahu. Z toho jsem usoudil, že tento font přispívá k větší čitelnosti. Z obrázku 9 lze vidět, že výsledný vzhled je v základě velmi podobný s návrhem. Zůstaly zachovány ostré hrany a dlaždice, menu už neobsahuje logo společnosti, ale stojí samostatně. Administrátorská část kontaktů byla oddělena a přesunuta do sekce „Administrace."



Obrázek 9: Hlavní strana aplikace.

Primární určení aplikace je pro stolní počítače, laptopy nebo zařízení s vyšším rozlišením. Responzivní verze aplikace byla vytvořena jako určitá alternativa skutečnosti, že uživatel bude nucen jisté úkony udělat z mobilního telefonu nebo ze zařízení s nižším rozlišením. Tato verze je však ochuzena o některé „méně" podstatné údaje v tabulkách. Dalo by se říct, že webová verze je o nějakou část plnohodnotnější než responzivní. Tabulky, ve kterých nebylo možno nedůležité informace vypustit, protože žádné neobsahují, se proto adaptivně nezobrazují. Uživatel není nijak výrazněji omezen v užívání responzivní verze.

4.3 Funkcionalita

4.3.1 Menu

Hlavním navigačním prvkem je menu u horního okraje stránky. Tento prvek při své responzivní podobě není složen do „hamburger menu", protože obsahuje tak málo položek, že nemá cenu tuto variantu volit. Obsahuje pět položek. Poslední z nich je tlačítko pro odhlášení aktuálně přihlášeného uživatele. Zbývající

čtyři si nyní vysvětlíme.

4.3.1.1 Hlavní strana

Na hlavní straně se vyskytují dlaždice se sekcemi jednotlivých školení a nástěnka s upozorněním na opakující se školení. Zobrazení těchto sekcí se liší u každého uživatele, podle oprávnění, ke kterým sekcím má přístup. Administrátor může vstupovat do každé sekce a může zde provádět potřebné úkony jako je např. přidávání nového školení a další.

4.3.1.2 Kontakty

Jsou druhým tlačítkem v pořadí menu. Slouží k evidenci kontaktů firem a provozoven. Uživatel získá přístupové právo k této kategorii automaticky, když je přidán administrátorem do systému.

4.3.1.3 Administrace

Tlačítko je viditelné jen a pouze pro administrátora. Kategorie obsahuje tři sekce. První z nich je „Obnovení kontaktů“, další „Uživatelé“ a poslední „Databáze“.

4.3.1.4 Můj účet

Toto tlačítko slouží k sumarizaci dat uživatelů, která jsou uvedena v databázi společnosti. „Můj účet“ je přístupný pro každého uživatele, který má v systému vytvořený účet. Uživatel si zde může změnit email nebo heslo.

4.3.2 Sekce

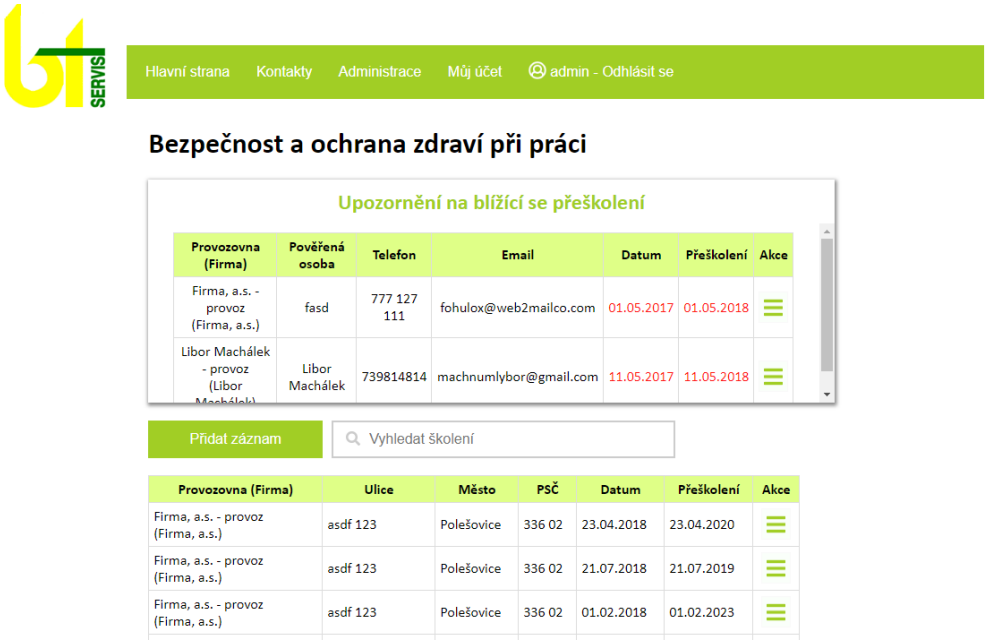
Kromě hlavního menu systém obsahuje sekce. Jsou to v podstatě kategorie, které shlukují obsah, který má nějaké společné téma. Sekcí je zde spousta. Dělíme je na ty, ke kterým může přistupovat běžný uživatel (školitel). A ostatní – k těm má přístup jen Administrátor. Obsah sekcí byl navržen intuitivně, aby bylo na první pohled poznat co, které tlačítko dělá. Pokud například v tabulkách uživatel neví, co znamená ikona se třemi čárkami pod sebou, může na ně najet kurzorem myši a objeví se mu jednoslovný nebo dvouslovný popis.

4.3.2.1 Sekce na hlavní straně a v kontaktech

Jak již víme hlavní strana obsahuje sekce s názvy školení. Po kliknutí na nějakou sekci. Se nám otevře stránka, kterou vidíme na obrázku 10.

Dané školení obsahuje box s upozorněními na blížící se přeškolení, tlačítko pro přidávání nového školení, vyhledávací okno a tabulku s již zadanými školeními, ta obsahuje tlačítko s ikonou třech svislých čar pod sebou, jež nás přesměruje na novou stránku s vypsanými podrobnostmi o zvoleném školení.

Při pohledu na načtenou stránku viz obrázek 11, vidíme detailní tabulku s informacemi a vpravo od tabulky se vyskytují tlačítka, pro editaci a mazání záznamů. Dále pak tlačítko poslat email, které pošle email uvedené kontaktní osobě. Poslední částí na této stránce je box s tlačítky pro práci s prezenčními listinami. Filozofie této části je taková, že uživatel si vygeneruje listinu a následně ji vytiskne. Po proběhnutém přeškolení listinu naskenuje a ve formátu PDF ji opět nahraje do systému.



Obrázek 10: Sekce školení

Obsah sekce v kontaktech se nijak zvlášť neliší od obsahu sekcí na hlavní straně. Jediným rozdílem je, že při přidávání firmy se objeví dialog, jestli uživatel chce zadat provozovnu na stejné adrese jako je adresa firmy.

4.3.2.2 Sekce v administraci Obsahy sekcí v administrátorské části jsou však již dost odlišné. Nachází se zde tři sekce a každá má jiný vzhled. Nám již známý design má stejnou podobu jen v případě sekce „Uživatelé“. Jediným rozšířením je tlačítko „Správa oprávnění.“ Po vstupu na stránku s právy viz obrázek 12 můžeme vidět seznam evidovaných školitelů a práva, která jsou jim přidělena. Práva může měnit jen administrátor. Jediná dvě práva, která nelze měnit jsou „Adminitrátor“ a „Kontakty“.

Další sekce nese název „Obnova kontaktů“. Jsou zde vypsány kontakty, které byly smazány ať už úmyslně nebo nedopatřením. Administrátor tak má dvě možnosti. Může kontakt obnovit i se záznamy o školeních dané firmy či provozovny, anebo kontakt trvale i se všemi školeními (dané firmy či provozovny) smazat trvale.

Poslední sekcí je „Databáze.“ V této poslední sekci, ke které má administrátor přístup jsou zobrazeny zálohy databáze, které administrátor vytváří. Vzhledem tato sekce připomíná sekci s obnovou kontaktů, je zde jen drobná změna. Existují tři možnosti, co lze s databází provádět. Můžeme si zálohu stáhnout ze serveru, můžeme ji nahrát anebo smazat. Nad tabulkou s databázemi je tlačítko „Zálohovat“, které slouží k vytvoření zálohy.

Obrázek 11: Stránka s podrobnostmi

Jméno a příjmení	Uživatelské jméno	Administrátor	KONT	BOZP	PO	EKO	Akce
Libor Machálek	admin	✓	✓	✓	✓	✓	Žádná akce
John Doe	machnum	✗	✓	✗	✓	✗	
Alan Doe	egeenquf	✗	✓	✗	✗	✓	

Obrázek 12: Tabulka s přístupovými oprávnění.

Závěr

Výsledkem této práce je aplikace pro firmu BT servis, která odpovídá požadavkům zadavatele a majitele společnosti. Aplikace bude používána k evidenci školení vykonávané zaměstnanci výše zmíněné školící agentury. Vytvořený systém naskytuje i několik možností k rozšíření např. vícenásobné přidávání záznamů, automatickou zálohu databáze, přidávání sekcí na hlavní stránce či rozšíření o další sekce nebo moduly, které by se mohly hodit ve firemním prostředí například sekce pro účetnictví. Framework CodeIgniter znám již z dřívější doby. Měl jsem základní znalosti, jak funguje a jaké možnosti poskytuje. Naskytly se však situace, kdy jsem dělal úplně novou věc a bylo potřeba si tyto věci dostudovat. Díky této bakalářské práci jsem nabyl spoustu cenných skutečností, které se budou hodit v budoucnu.

Conclusions

The result of this work is an application for BT servis, which corresponds to the requirements of the client and the owner of the company. The application will be used to record trainings provided by employees of this training center. Created system can be extended by a lot of expansions such as adding more records in one form, automaticly database backup, adding sections or modules that may be appropriate for business such as accounting. I knew CodeIgniter since the earlier time. I had basic knowledge of how it works and what options it provides. But there were situations when I was doing new things and it was necessary to learn these things. Thanks to this bachelor thesis I have gained a lot of valuable expirience that will be useful in the future.

A Zprovoznění aplikace

Předpokladem pro zprovoznění aplikace je webový server s podporou jazyka PHP verze 5.3.1 nebo vyšší. Dále pak databázový server MySQL. Nejprve zprovozníme databázi. Na přiloženém CD/DVD v adresáři „data/“ nalezneme SQL skript s názvem „intranet.sql“. Tento skript importujeme pomocí systému pro správu databází viz kapitola 3.2.2.

Nyní přichází na řadu aplikace. Na přiloženém CD/DVD je archiv intranet. Rozbalený obsah archivu (tzn. složku intranet) nahrajeme do kořenového adresáře na serveru. Otevřeme soubor „config.php“ umístěný v adresáři „intranet/application/config“ a změníme následující řádek: `$config['base_url'] = '';` za řádek `$config['base_url'] = 'https://example.com/intranet/';` a uložíme změny. Místo adresy „example.com“ lze použít IP adresu, například 127.0.0.1 pro localhost. Variantu „https://“ používáme pouze za předpokladu, používáme-li protokol HTTPS. Jinak do „base_url“ napíšeme „http://“. Poté ve stejném adresáři, tedy „intranet/application/config“, otevřeme soubor „database.php“ a v poli „$db“ vyplníme přihlašovací údaje uživatele, jenž má přístup k databázi, kterou jsme vytvořili postupem z předchozího odstavce. Jedná se o položky „username a password“.

Tento postup však můžeme přeskočit, protože aplikace je již umístěna na webu na adrese: https://www.libormachalek.cz/intranet. Přihlašovací jméno na administrátorský účet je: admin. Heslo k tomuto účtu je také: admin.

B Obsah přiloženého CD/DVD

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého CD/DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

doc/
: Text práce ve formátu PDF včetně všech příloh a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu.

src/
: Kompletní zdrojové texty webové aplikace včetně knihoven použitých k zajištění správně funkce aplikace.

data/
: Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

readme.txt
: Instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Literatura

[1] CodeIgniter Web Framework. Dostupný z: https://www.codeigniter.com/

[2] ŠIMŮNEK, Milan. *SQL kompletní kapesní průvodce*. Vydala Grada Publishing, spol. s.r.o. U Průhonu 22, Praha 7. Vydání 1. rok 1999. ISBN 80-7169-692-7.

[3] Databáze MySQL. Dostupné z: https://www.mysql.com/why-mysql/

[4] Jazyk HTML. Dostupné z: https://www.w3.org/standards/webdesign/htmlcss

[5] W3C, Jazyk CSS. Dostupné z: https://www.jakpsatweb.cz/css/

[6] JavaScript. Dostupné z: https://www.jakpsatweb.cz/javascript/

[7] jQuery. Dostupné z: https://jquery.com/

[8] AJAX. Dostupné z: https://www.itnetwork.cz/javascript/ajax

[9] Autentizační knihovna Tank Auth pro framework CodeIgniter. Dostupná z: https://konyukhov.com/soft/tank_auth/

[10] Knihovna TCPDF. Dostupná z: https://tcpdf.org/

[11] Překlady hlášek ve frameworku CodeIgniter do českého jazyka. Dostupné z: https://github.com/bcit-ci/codeigniter3-translations