

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Informační systém pro lyžařskou školu



2019

Vedoucí práce: Mgr. Tomáš Kühn,
Ph.D.

Martin Stojmenov

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Martin Stojmenov
Název práce: Informační systém pro lyžařskou školu
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2019
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.
Počet stran: 36
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Martin Stojmenov
Title: Information System for a Ski School
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2019
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Kühn, Ph.D.
Page count: 36
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Bakalářská práce se zabývá tvorbou informačního systému pro lyžařskou školu. V úvodu práce je krátce vysvětlena motivace a problematika. Dále jsou popsány užité technologie a jejich použití. K této bakalářské práci byla vytvořena aplikace, vyvíjena zejména v Nette Framework napojeným na databázový systém MySQL. Součástí práce je i uživatelská dokumentace.

Synopsis

This thesis deals with creation of information system for ski school. Motivation is briefly explained at the beginning. The thesis also describes used technologies. An application was developed for the thesis, especially in Nette Framework connected to MySQL database system. User documentation is also included.

Klíčová slova: lyžařská škola; informační systém; rezervační systém; webová aplikace; Nette Framework

Keywords: ski school; information system; booking system; web application; Nette Framework

Děkuji Mgr. Tomáši Kührovi, Ph.D. za cenné rady a věcné připomínky při tvorbě této bakalářské práce.

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

datum odevzdání práce

podpis autora

Obsah

1	Úvod a motivace	8
2	Programátorská dokumentace	9
2.1	Softwarová architektura MVC	9
2.2	Nette Framework	9
2.2.1	Latte	10
2.2.2	Tracy	11
2.2.3	NetteDatabase	12
2.2.4	Kdyby/Translation	12
2.2.5	Forms	13
2.2.6	Passwords	13
2.2.7	Nette a AJAX	13
2.3	Ostatní použité technologie	14
2.3.1	HTML, CSS, JavaScript	14
2.3.2	AJAX	14
2.3.3	MySQL [22]	14
2.3.4	Bootstrap a knihovna JQuery	15
2.3.5	NetBeans a Git	15
2.4	Struktura aplikace	15
2.4.1	App	15
2.4.2	Classes	16
2.4.3	Config	16
2.4.4	Forms	16
2.4.5	Lang	17
2.4.6	Model, Presenters, Templates	17
2.4.7	Www	17
2.5	Struktura uživatelů v aplikaci	17
2.6	Struktura databáze	17
2.6.1	Tabulka accounts	20
2.6.2	Tabulka instructors	21
2.6.3	Tabulka lessons	21
2.6.4	Tabulka available lessons	22
2.6.5	Tabulka gdpr	22
3	Uživatelská dokumentace	23
3.1	Všeobecná funkcionality	23
3.1.1	Zprávy	23
3.1.2	Emailová notifikace	23
3.1.3	Jazyky	24
3.1.4	Registrace	24
3.1.5	Aktivace účtu	24
3.1.6	Přihlášení	24
3.1.7	Zapomenuté heslo	25

3.2	Uživatelská funkcionalita	25
3.2.1	Zobrazení rozvrhu hodin	25
3.2.2	Objednávka lekce pro daný čas	26
3.2.3	Přehled lekcí	27
3.2.4	Správa účtu	27
3.3	Instruktorská funkcionalita	28
3.3.1	Zápis výuky	28
3.3.2	Přehled všech hodin	28
3.4	Funkcionalita moderátora	28
3.4.1	Správa hodin všeobecně	29
3.5	Funkcionalita Administrátora	29
3.5.1	Správa účtů	30
3.5.2	Přehled hodin	30
	Závěr	31
	Conclusions	32
	A Příloha: Instalace a spuštění aplikace	33
	A.1 Hosting a HTTPS certifikát	33
	A.2 Požadavky	33
	A.3 Instalace	33
	A.4 Spuštění	33
	B Obsah přiloženého CD/DVD	34
	Literatura	35

Seznam obrázků

1	Graficky znázorněná MVC architektura	10
2	Adresářová struktura v Nette [3]	11
3	Adresářová struktura v aplikaci	16
4	Diagram případu užití uživatele	18
5	Diagram případu užití instruktora	18
6	Diagram případu užití moderátora	19
7	Diagram případu užití administrátora	19
8	Schéma databáze	20
9	Reakce systému zprávami	23
10	Přepnutí jazyka v informačním systému	24
11	Rozvrh hodin na počítači	25
12	Rozvrh hodin na mobilním zařízení Honor 8	26

Seznam zdrojových kódů

1	Databázový dotaz v Nette	12
2	Stejný dotaz přeložený do MySQL	12
3	Vlastní validační pravidlo	13

1 Úvod a motivace

Ačkoliv je to v dnešní době poněkud zarážející, mnoho lyžařských škol v České republice nedisponuje online rezervačními systémy. Klienti se zpravidla objednávali přes telefon, či kontaktní formulář, s následně probíhající komunikací skrze email. Příčinou absence oněch systémů je zpravidla finanční náročnost. Nechat si vytvořit a následně spravovat vlastní rezervační systém není nejlevnější, jelikož neexistují nějak stanovené hranice, které by určovaly pohyb cen za zpracování. Uvážíme-li že spousta lyžařských škol má pouze jednu pobočku, přičemž ta je většinou aktivní pouze v zimních měsících (zpravidla 2–4 měsíce), pak je toto řešení nevhodné. Další možností je nechat si zřídit univerzální rezervační systémy od firem, které tuto službu nabízejí. V takovém případě je pořízení většinou zdarma, ale platí se měsíční nebo jinak paušalovaný pronájem. Avšak podstatnější je, že se jedná o univerzální systémy a jen těžko se dosahuje na funkcionalitu a přizpůsobení, kterým mohou disponovat vlastní rezervační systémy.

Toto jsou jedny z důvodů, proč jsem se rozhodl pro vytvoření vlastního informačního systému. Tento systém bude používán v nové lyžařské škole poblíž Olomouce, kterou budu spoluvlastnit. V praxi to znamená, že budu správcem a mohu systém nadále vylepšovat podle potřeb, které se vyskytnou. Velkou výhodou je nepotřebnost platit externímu vývojáři, správci či firmě, čímž se systém stane pro lyžařskou školu téměř beznákladovým. V dnešní době je výhodné takové systémy tvořit jako webové aplikace, a to hlavně z důvodu minimálních požadavků kladených na uživatele. V tomto případě se jedná pouze o moderní webový prohlížeč. Problematika webů mě zajímá už delší dobu a spolu s mým blízkým vztahem k lyžování a samotné výuce, je pro mě téma této bakalářské práce přímo skvělé. Při tvorbě systému jsem se navíc seznámil s technologiemi, které jsem doposud nepoužil.

Cílem této bakalářské práce je tedy vytvoření vlastního informačního systému pro lyžařskou školu, který bude uživatelský přívětivý. Systém bude schopný evidovat časové úseky pro instruktory, které si sami vypíší, přičemž tyto časové úseky budou sloužit klientům pro objednání výuky podle jimi zvolených kritérií. Vedoucí instruktoři a správce systému budou mít možnost manipulovat s objednanými hodinami a vypsányi časovými úseky dle libosti. Samozřejmostí je správa uživatelských účtů, tím pádem i instruktorů, a získávání přehledu o proběhlých hodinách všech uživatelů. S ohledem na rozhodnutí EU o zpracování osobních údajů bude k systému vypracováno i základní GDPR [1] [2].

2 Programátorská dokumentace

Aplikace byla vyvíjena objektovým přístupem v Nette frameworku. Na základně použité technologie byla určena i návrhová architektura, včetně samotného objektového přístupu. Pro ukládání dat slouží databázový systém MySQL.

2.1 Softwarová architektura MVC

Jako softwarová architektura je použita MVC [3], která se uchytila zejména ve webových aplikacích. Její hlavní podstatou je oddělení logiky aplikace od prezentace obsahu. Její přínos je samozřejmě vidět z pohledu programátora, jelikož drasticky zvyšuje přehlednost a pořádek ve zdrojových kódech, jsou-li její principy dodrženy.

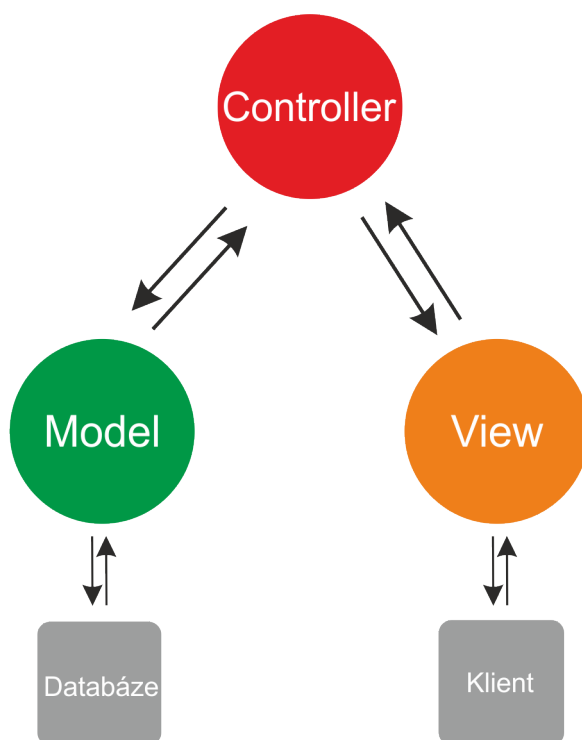
MVC architektura [4] se skládá ze tří základních vrstev, Model, View a Controller. **Model** se stará o datovou část celé aplikace, v našem případě to znamená napojení a komunikaci s databází. Na druhé straně je vrstva **View**, která představuje to, co vidí samotný uživatel, tedy zodpovídá za samotné zobrazení dat. Někdy se můžeme setkat s označením šablona, a to z důvodu častého užití různých šablonovacích systémů. Komunikaci mezi těmito dvěma vrstvami zajišťuje **Controller**, který se dá chápat jako prostředník. Přijímá požadavky od View (tzn. od uživatele) a následně tyto požadavky zpracovává, nezřídka s dotazy na model. S toho samozřejmě plyne napojení Controlleru na Model. Výše popsané můžeme pěkně vidět na obrázku 1.

2.2 Nette Framework

Nette framework [5] je český framework, jehož autorem je David Grudl. Jedná se o balíček vyspělých a samostatně použitelných komponent pro PHP [6]. Nette se prezentuje jako použitelný a velmi vstřícný vůči programátorovi s přednostmi jako je [7]:

- bezpečnost,
- efektivní práce s databází,
- skvělý šablonovací systém,
- podpora moderních technologií (AJAX, HTML5...),
- kvalitní dokumentace s aktivní českou komunitou.

Paradoxně jako slabší stránku programátoři často uvádějí slabší dokumentaci (obsahově), kterou ovšem nahrazuje aktivita komunity (zejména české) a školení, která pořádá i sám autor Nette. Zde se ale sluší říci, že tento názor je čím dál méně relevantní, jelikož se dokumentace stále rozrůstá a aktualizuje.



Obrázek 1: Graficky znázorněná MVC architektura

Nette je založeno na MVC architektuře, kterou jsem již popisoval. Dodržuje její principy, avšak setkáváme se zde s trochu jiným pojmenováním vrstev [3]. Místo tradiční Model-View-Controller zde máme Model-View(Template)-Presenter, funkce zde zastávají stejné. Adresářová struktura je vidět na obrázku 2, avšak je potřeba zmínit, že se nejedná o pevně danou věc. Po upravení souborů `index.php` a `bootstrap.php`, které v daném pořadí zajišťují správné předání řízení do místa v aplikaci, lze tuto strukturu změnit.

Jak jsem zmínil, Nette se skládá zejména z různých balíčků a knihoven. V následujících sekcích budu popisovat zejména ty, které byly při tvorbě aplikace hojně využívány.

2.2.1 Latte

Jako šablonovací systém zde máme Latte [8]. Jeho obrovskou výhodou je velké zpřehlednění kódu a zabezpečení proti XSS útokům¹. Již zde narážíme na zmíněnou přednost Nette, bezpečnost. Latte totiž automaticky escapuje proměnné, které vypisujeme. Samozřejmě chceme-li vypsát proměnnou bez escapování, Latte to umožňuje, avšak je potřeba o to explicitně požádat modifikátorem `noescape` za výpisem proměnné (`$var|noescape`). Další užitečnou pomůckou jsou tzv. makra a `n:makra`, která jednak zapouzdřují PHP funkcionalitu (jako např. cykly) a záro-

¹Cross Site Scripting - jedná se útok, který zneužívá neošetřené vstupy webových stránek, při kterém útočník podvrhne stránce svůj vlastní kód [9]



Obrázek 2: Adresářová struktura v Nette [3]

veň přidávají i vlastní (například makro pro překlad). Klasická makra se zapisují do `{}` a každé takové makro lze přepsat do podoby `n:makra`, která se zapisují přímo do HTML tagů. Výsledkem je mnohem přehlednější a strukturovanější kód.

2.2.2 Tracy

Obrovskou pomocí je ladící nástroj Tracy [10], často označován jako „laděnka“. Každý, kdo programoval v PHP, se určitě setkal s chybovými hláškami, které PHP vypisuje při některých chybách. Častokrát jsou tyto hlášky zmatečné a hledání chyby může být velmi obtížné. Tracy dělá pravý opak, chyby vyznačí přesně v místě, kde se vyskytly, je schopna navrhnout řešení, popř. pomoci s nápravou. Dále je dostupný tzv. debugger bar, který si lze přizpůsobit dle libosti. Zobrazuje např. dotazy na databázi, identitu uživatele, přesměrování a mnoho dalšího. Tyto vychytávky jsou samozřejmě dostupné pouze ve vývojářském režimu, jelikož by tato citlivá data mohl uživatel zneužít. Po přepnutí do produkčního režimu ovšem tracy své kouzlo neztrácí, jelikož poměrně jednoduše dokáže logovat chyby, které nastanou za běhu aplikace. Těto funkcionality jsem pochopitelně využil, jelikož logování je velmi užitečné a dnes snad i standardem.

2.2.3 NetteDatabase

Práci s databází zajišťuje Nette Database [11], kterou lze rozdělit do dvou podkategorií. Database Core a Database Explorer. Database Core je ve své podstatě obalené PDO [12] známé z PHP, psaní dotazů je tedy velice podobné. Na druhou stranu Database Explorer je mnohem komplexnější. Upouští se zde od psaní tradičních SQL dotazů, které jsou nahrazeny skrze objektový přístup nad objektem Context, a to s přívětivou syntaxí (viz zdrojový kód 1). Navíc si automaticky vybere pouze ty sloupce, které jsou opravdu potřeba (viz výsledný dotaz v SQL 2). Samozřejmostí je automatické escapování², čímž opět narážím na bezpečnost frameworku, jelikož se tím předchází známému SQL injection útoku³.

```
1 $this->database->table('lessons')->where([
2     'status' => 0,
3     'date >' => '2019-1-1 00:00:00'
4 ])
5         ->order('instructor_ID DESC')
6         ->limit(10, 5)
7         ->fetchAll();
```

Zdrojový kód 1: Databázový dotaz v Nette

```
1 SELECT `date`, `instructor_ID`
2 FROM `lessons`
3 WHERE (`status` = 0) AND (`date` > '2019-1-1 00:00:00')
4 ORDER BY `instructor_ID` DESC
5 LIMIT 10
6 OFFSET 5
```

Zdrojový kód 2: Stejný dotaz přeložený do MySQL

2.2.4 Kdyby/Translation

Bylo vhodné, aby systém uměl nejen Češtinu, ale i Angličtinu. K tomuto účelu poslouží rozšíření zvané Kdyby/Translation [13], implementující ITranslator z Nette, za použití Translation z frameworku Symfony. Díky tomuto rozšíření je možné napsat tzv. slovníky, které slouží pro překlad, a to nejen v šablonách, ale v celé aplikaci. Slovníky se zapisují do souborů ve formátu NEON [14], což zajišťuje poměrně čistou strukturu souboru. Jelikož je Kdyby/Translation kompatibilní s Latte, pak pro překlad v šablonách můžeme použít makro `{_}Výraz{/_}`. Pro překlad v presenterech stačí přistoupit k objektu translator.

²Escapování je nahrazení znaků, které mohou mít určitý význam za jiné, odpovídající znaky proměnných

³SQL injection je typ útoku, při kterém útočník podvrhne stránce vlastní SQL dotaz a může narušit strukturu databáze nebo se dostat k citlivým informacím

2.2.5 Forms

Nedílnou součástí Nette, resp. každé webové stránky, jsou formuláře. Práce s nimi je v Nette velmi příjemná. Nejdůležitější je samotná bezpečnost, jelikož formuláře na webových stránkách představují jedno z největších potencionálních rizik. Jeden špatně ošetřený vstup může vyvolat nežádoucí stavy, zadá-li uživatel chybná data (ať záměrně, či omylem). Z toho důvodu Nette poskytuje velké množství validačních pravidel, které je možné rozšířit i o uživatelsky definovaná pravidla, a to jak na straně serveru, tak na straně klienta. Této funkcionality jsem využil v podobě validačního pravidla ošetřující vstup pro datum narození, jak je vidět ve zdrojovém kódu 3. Vykreslování formulářů by mohla být kapitola sama o sobě, jelikož Nette programátorovi nabízí několik způsobů, jak formuláře vykreslovat (zcela manuálně, jako komponentu, manuálně s kontrolou nad vykreslením, ...) [15].

```
1 public static function validateBirthday(FormControl $control)
2 {
3     $between[0] = (new Nette\Utils\DateTime("now"))
4         ->modify('-100 year');
5     $between[1] = (new Nette\Utils\DateTime("now"))
6         ->modify('-2 year');
7     $date = DateTime::createFromFormat('Y-n-j', $control->getValue());
8     return (Validators::isInRange($date, $between));
9 }
```

Zdrojový kód 3: Vlastní validační pravidlo

2.2.6 Passwords

Jen těžko by bylo možné pracovat s uživatelskými účty bez kvalitního zahashování⁴ hesel. Ukládání hesel v čitelné podobě představuje obrovské bezpečnostní riziko, kterému by neměl být nikdo vystaven. K tomuto účelu v Nette slouží třída „Passwords“ [16], která při zavolání funkce hash generuje otisky pomocí algoritmu bcrypt. Při volání této funkce je možné určit počet iterací algoritmu, já jsem ponechal defaultní hodnotu 10. Samozřejmě je i funkce pro zjištění, zda daný vstup odpovídá otisku.

2.2.7 Nette a AJAX

Bohužel Nette nepodporuje AJAX požadavky na straně klienta, a proto je doporučeno rozšíření `nette.ajax.js` od Vojtěcha Dobeše [17]. Toto rozšíření, spolu se `snippets`⁵ tvoří mocný nástroj, kterým můžeme pohodlně překreslit jen určitou část stránky [18]. Více o AJAXu v sekci 2.3.2

⁴Pomocí Hashovací funkce se vytvoří otisk vstupních dat, který je také označován jako Hash

⁵Nette snippets definuje jako výřezy ze šablony

2.3 Ostatní použité technologie

2.3.1 HTML, CSS, JavaScript

HyperText Markup Language [19], zkráceně HTML, je velmi populární a jednoduchý značkovací jazyk s počátky v 90. letech, který je podporován napříč velkou škálou zařízení. Je důležité zmínit, že jazyk HTML slouží k definici významu obsahu stránek, a to pomocí tagů⁶. Nejnovější verze je označována jako HTML5(.3) a přinesla řadu vylepšení, přičemž nejznámějšími jsou strukturální elementy (header, nav, section, footer, ...). Při vývoji této verze byl zároveň kladen důraz na zpětnou kompatibilitu.

Cascading Style Sheets [19], zkrácené CSS, je jazyk popisující zobrazení elementů. Je přímo vztažen k jazyku HTML, avšak lze jej použít např. i pro XML aj. Na rozdíl od HTML slouží k definici způsobu zobrazení obsahu pomocí selektorů. Aktuální verze se označuje jako CSS3 a stejně jako HTML5 přinesla řadu nové funkcionality, mezi kterou patří vizuální efekty aj.

JavaScript [20] se také řadí mezi známé technologie, jedná se o skriptovací jazyk vykonávaný na straně klienta. Slouží zejména pro přidání dynamiky na webové stránky, jako např. změna obsahu bez znovu načtení, reagování na události a další. Může posloužit i pro validaci dat, avšak zde je potřeba dát velký pozor. Jelikož se jedná o jazyk interpretovaný na straně klienta, pak jej tedy uživatel může svévolně změnit. Proto je vždy potřeba data validovat i na straně serveru.

2.3.2 AJAX

Před příchodem této technologie se musela webová stránka načíst pokaždé znovu, jestliže byla potřeba získat nějaká data ze serveru. Velkým přínosem proto byla technologie zvaná Asynchronous JavaScript and XML, zkráceně AJAX [21], která definovala způsob, jakým si může klient vyměňovat data se serverem, a to bez nutnosti znovu načtení stránky. Na straně klienta se AJAX používá skrze JavaScript (konkrétně objekt XMLHttpRequest), přes který se odešle požadavek a následně přijme data (odpověď) od serveru.

2.3.3 MySQL [22]

Jedná se o open source⁷ databázový systém, momentálně pod záštitou Oracle, který je určen primárně pro weby. Za jeho úspěchem stojí jednak jeho flexibilita (multiplatformovost⁸) a rychlost, která je na webu dosti důležitým aspektem. Právě spojení s webovými technologiemi, resp. s jazykem PHP, zapříčinilo jeho obrovskou oblibu a úspěch. MySQL je založeno na relačním modelu a ke komunikaci se serverem se využívá jazyk SQL (resp. jeho dialekt). Pro správu MySQL

⁶Tagem se zde rozumí značka jazyka HTML

⁷Mající volně dostupný zdrojový kód

⁸Multiplatformní software - běh není závislý na zařízení či prostředí

databází slouží aplikace phpMyAdmin nebo **Adminer**, kterou jsem při tvorbě aplikace použil.

2.3.4 Bootstrap a knihovna JQuery

Bootstrap [23] je sada nástrojů pro HTML a CSS, jejíž největším přínosem je zajištění responzivity a stejného zobrazení prvků v různých prohlížečích (včetně starších). Obsahuje řadu předpřipravených komponent (grid systém, navbary, celé šablony, ...), které lze snadno zakomponovat do vlastního projektu. V aplikaci byl použit zejména navbar, modální okna a form-controls.

JavaScriptová knihovna JQuery [24] usnadňuje práci s JavaScriptem. Její přínos je vidět zejména při práci s DOMem⁹ nebo AJAXem, jelikož velmi zpřehledňuje a usnadňuje programování. Nabízí i řadu nových funkcí, které se v nativní JavaScriptu nevyskytují. Zde bych rád zmínil, že knihovny JQuery není v aplikaci nijak hojně využíváno, spíše slouží jako předpoklad pro použití Bootstrapu a Nette Ajaxu.

2.3.5 NetBeans a Git

Jen těžko se vyvíjí aplikace bez řádného vývojového prostředí. Samozřejmě to lze, snad každý začínal v poznámkovém bloku, ale je to těžkopádné a programátorovi se nedostane žádné větší pomoci. Já jsem využil vývojové prostředí NetBeans [26], které nabízí rozšiřující pluginy pro podporu PHP, Nette a Latte. Jedná se o volně dostupný open source projekt spadající pod Apache Software Foundation. Další výhodou je napojení z vývojového prostředí přímo na hosting (skrze FTP¹⁰), kde běží server Apache, na kterém běží i informační systém.

Spolu s NetBeans jsem využil verzovacího nástroje Git, který velmi usnadňuje správu zdrojového kódu a celého projektu při jeho úpravě.

2.4 Struktura aplikace

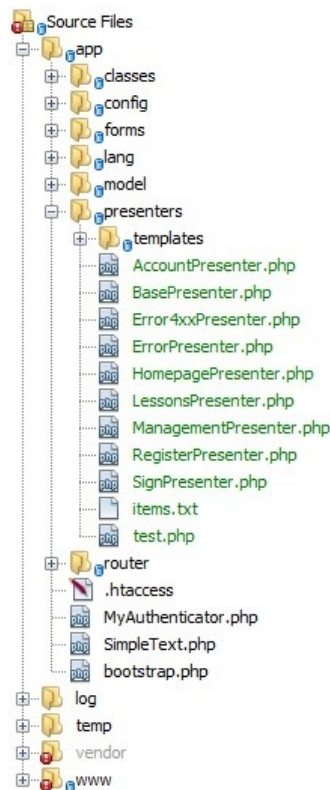
Aplikace dodržuje princip MVC architektury a od toho se odvíjí i adresářová struktura (viz obrázek 3). V této sekci blíže popíši obsah složek, kde se nachází nejvíce obsahu z pohledu programování. Obsah zbylých složek je zřejmý z obrázku 2.

2.4.1 App

Složka App obsahuje celou logiku aplikace. Struktura jejího obsahu je stejná, jako v čisté verzi Nette. Zajímavá zde může být třída `MyAuthenticator`, která zajišťuje autentizaci uživatelů (komunikuje s příslušným modelem).

⁹Document Object Model [25] je standard, který říká jak přistupovat k obsahu stránky

¹⁰File Transfer Protocol slouží pro přenos souborů mezi počítači



Obrázek 3: Adresářová struktura v aplikaci

2.4.2 Classes

V podsložce `classes` nalezneme pomocné třídy, které slouží pro uchování dat, která nebylo vhodné jinak udržet. Jsou používány zejména pro změnu údajů. Jedná se o třídy `Account` a `Instructor`, které jsou schopné uchovat informace z příslušných databázových tabulek. Třída `Lesson` slouží jako pomoc při přípravě hodin pro šablony.

2.4.3 Config

Tato složka obsahuje konfigurační soubory ve formátu NEON¹¹, zejména `config.neon`, který uchovává údaje potřebné k připojení databázi, registruje služby uvnitř aplikace apod. Soubor `config.local.neon` má stejnou funkci, avšak slouží pro lokální verzi, kdežto první zmíněný pro produkční verzi.

2.4.4 Forms

Podsložka `forms` obsahuje továrny na komponenty, jmenovitě na formuláře. Ve složce jsou i třídy `UserFormRules`, která slouží pro definici dalších validačních pravidel a `UserAccountRestrictions`, kterou všechny továrny imple-

¹¹NEON [14] je snadno čitelný formát pro serializaci dat

mentují. Tyto komponenty je možné vytvářet i přímo v presenterech, ale snížila by se tím přehlednost kódu.

2.4.5 Lang

Méně obvyklá je složka `lang`, která obsahuje tzv. slovníky, které jsou ve formátu NEON, stejně jako konfigurační soubory. Ty slouží pro překlad mezi různými jazyky, kterých je možno mít libovolně mnoho, přičemž informační systém umí česky a anglicky.

2.4.6 Model, Presenters, Templates

Složky jsou ekvivalenty vrstev MVC architektury, jejich obsah je tedy zřejmý.

2.4.7 Www

Jedná se o jediný adresář, do kterého může přistoupit kdokoliv, kdo navštíví danou webovou stránku. V tomto adresáři je tedy umístěn soubor `index.php`, který předává požadavky dál do aplikace. Při přístupu do ostatních adresářů je přístup zamítnut, což nám zajistí správně nastavené soubory `.htaccess`.

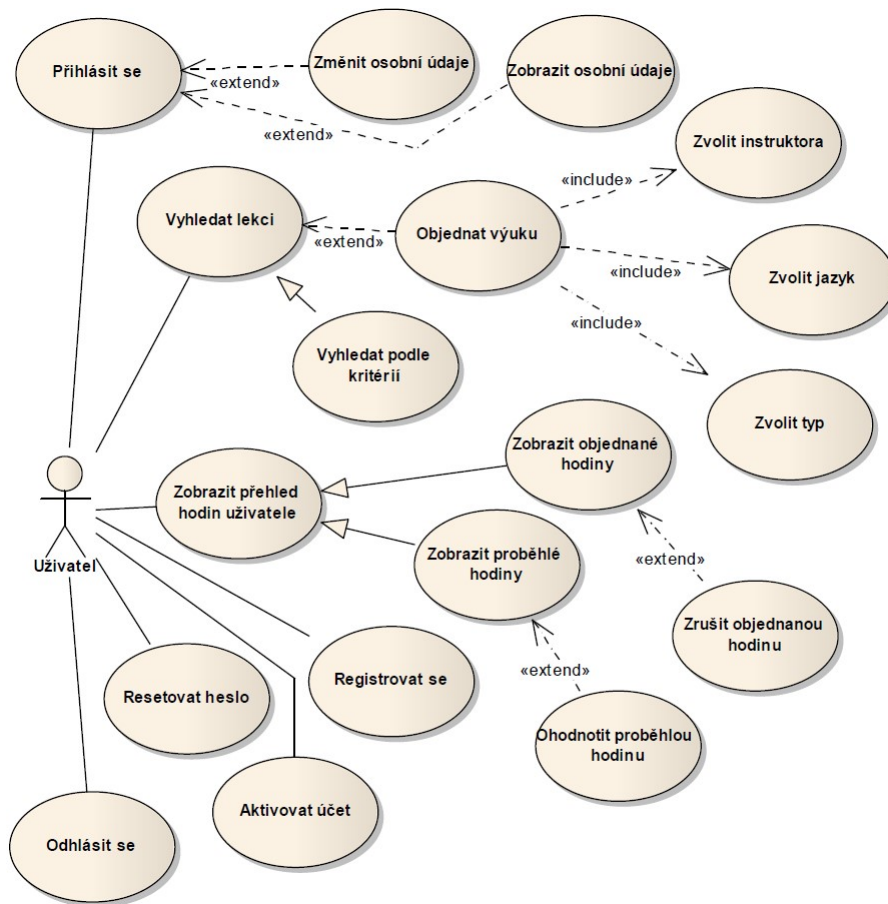
2.5 Struktura uživatelů v aplikaci

Bez správných uživatelských oprávnění by systém fungoval stěží. Proto je každý účet brán jako uživatel a další práva jsou pouze přidávána či odebírána, případně je asociována instruktorská část účtu. V praxi to znamená, že bez ohledu na práva má i administrátor možnost dělat základní úkony, jako uživatel, čímž se myslí např. objednávka lekcí. Možností uživatelů, instruktorů a uživatelů s vyšším oprávněním můžeme vidět na diagramech případu užití 4, 5, 6 a 7.

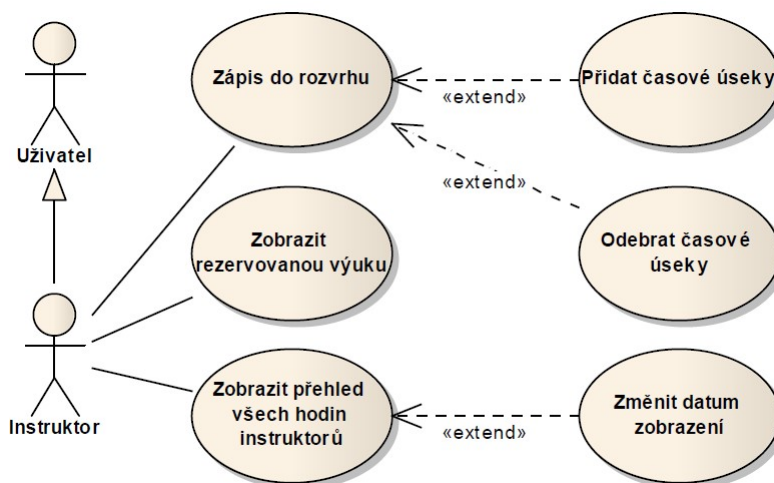
2.6 Struktura databáze

Databázová struktura se skládá z pěti tabulek, jak můžeme vidět na obrázku 8. Integrita je zajištěna formátováním úložiště dat jako InnoDB. Uživatelé se ukládají do tabulky „accounts“, přičemž pokud je uživatel instruktor, tak má v tabulce nastaven cizí klíč „instructor_ID“, který ukazuje na odpovídající záznam v tabulce „instructors“. V tabulce „instructors“ je opět cizí klíč na uživatele, pro jednodušší přístup k jeho osobním informacím. Pokud je uživateli zrušen instruktorský účet, pak je jeho příslušný cizí klíč v tabulce „accounts“ nastaven na NULL¹², přičemž v tabulce „instructors“ záznam zůstane, aby nebyla akce destruktivní. Tím pádem je stále možné zjistit, kdo byl dříve instruktorem a také tím zůstanou úplné údaje u lekcí.

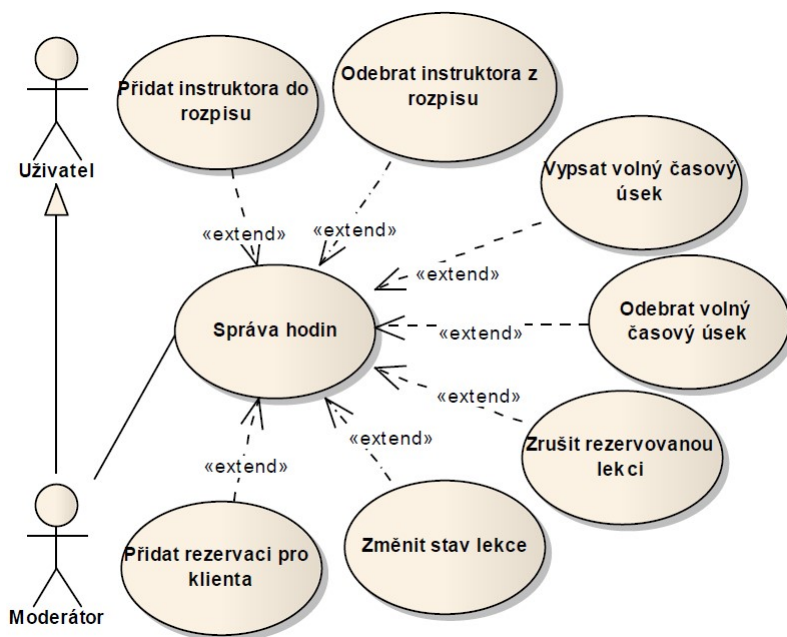
¹²NULL = žádná hodnota



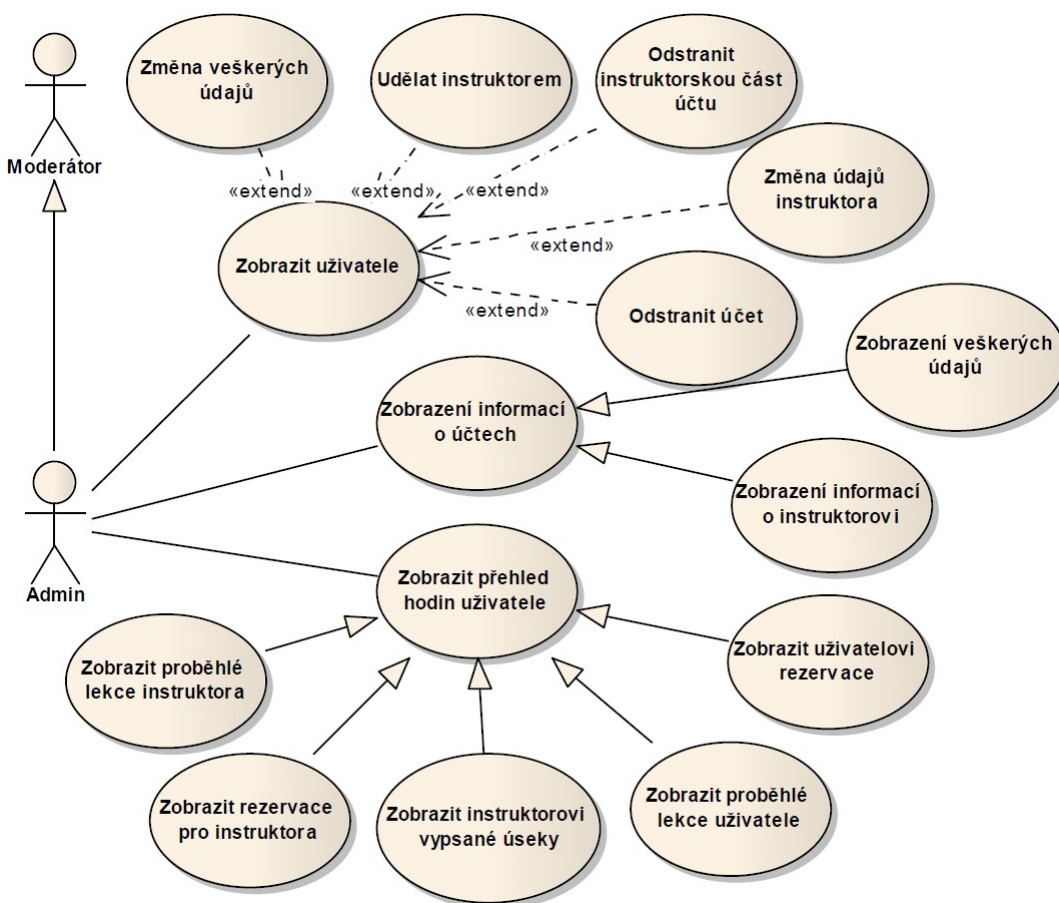
Obrázek 4: Diagram případu užití uživatele



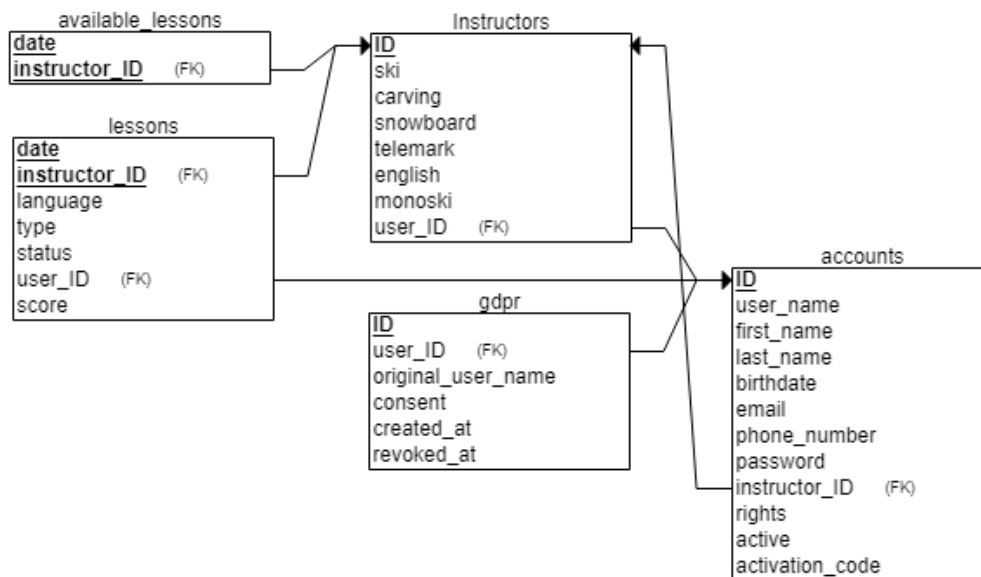
Obrázek 5: Diagram případu užití instruktora



Obrázek 6: Diagram případu užití moderátora



Obrázek 7: Diagram případu užití administrátora



Obrázek 8: Schéma databáze

Tabulka „available_lessons“ obsahuje vypsané časové úseky instruktorů, které si mohou uživatelé rezervovat. Naopak v tabulce „lessons“ jsou zarezervované a proběhlé lekce.

2.6.1 Tabulka accounts

Uchovává údaje o registrovaných uživateli. Při odebrání instruktorských oprávnění se nastaví cizí klíč na NULL, díky vykonání akce „SET NULL“. Podle GDPR musí mít uživatel právo na odebrání svých osobních informací ze systému, z toho důvodu je tolik sloupců tabulky nastaveno na akceptování hodnoty NULL.

Při registraci je účet neaktivní a je jej potřeba aktivovat. Lze provést i deaktivaci účtu. Do neaktivního účtu se není možné přihlásit.

Tabulka má následující atributy:

- ID – PRIMARY KEY, AUTO INCREMENT;
- user_name – UNIQUE – uživatelské jméno, které slouží pro jednodušší výběr uživatele a také jako jediný identifikační údaj po odebrání osobních údajů podle GDPR;
- first_name – NULL – křestní jméno uživatele;
- last_name – NULL – příjmení uživatele;
- birthdate – NULL – datum narození uživatele;
- email – NULL, UNIQUE – email uživatele;

- phone_number – NULL – telefonní číslo uživatele;
- password – zahashované heslo;
- instructor_ID – FOREIGN KEY, UNIQUE, NULL - odkaz do tabulky instruktorů;
- rights – 0/1/2, práva uživatele, kde 0 = user, 1 = moderátor a 2 = administrátor;
- active – 0/1, značí, zda je uživatelský účet aktivní;
- active_code – NULL – náhodná posloupnost znaků, která slouží jako integrita při obnovení hesla, či aktivaci účtu;

2.6.2 Tabulka instructors

Obsahuje schopnosti instruktorů, asociovaných k registrovaným účtům. Při odebrání instruktorských oprávnění zůstává cizí klíč nedotčen, aby bylo možné asociovat bývalého instruktora. Při úplném smazání uživatelského účtu (z tabulky „accounts“) se cizí klíč nastaví na NULL, díky vykonání akce „SET NULL“.

Tabulka má následující atributy:

- ID – PRIMARY KEY, AUTO INCREMENT;
- ski – 0/1, značí, zda může instruktor vést výuku lyžování;
- carving – 0/1, značí, zda může instruktor vést pokročilou výuku lyžování, tedy carving;
- snowboard – 0/1, značí, zda může instruktor vést výuku na snowboardu;
- telemark – 0/1, značí, zda může instruktor vést výuku na telemarkových lyžích;
- english – 0/1, značí, zda může instruktor vést výuku v anglickém jazyce;
- monoski – 0/1, značí, zda může instruktor vést výuku na monoski;
- user_ID – FOREIGN KEY, NULL - odkaz do tabulky účtů;

2.6.3 Tabulka lessons

Uchovává všechny objednané a proběhlé hodiny. Při pokusu o odstranění záznamu v tabulce instructors je zavolána akce „RESTRICT“, která zabráni odstranění, jestliže v této tabulce existují hodiny s cizím klíčem odkazujícím na odstraňovaný záznam. Při smazání uživatelského účtu se nastaví user_ID na NULL, zavoláním akce „SET NULL“.

Tabulka má následující atributy:

- date – PRIMARY KEY – datum, kdy výuka proběhne/proběhla (vč. hodiny);
- language – czech/english, v kterém jazyce výuka proběhně/proběhla;
- type – jakého je výuka typu (ski, carving, snowboard, telemark, monoski);
- status – 0/1, značí, zda je hodina nezaplacená/zaplacena;
- instructor_ID – PRIMARY KEY, FOREIGN KEY - instruktor, který provede výuku;
- user_ID – FOREIGN KEY - uživatel, který má rezervovanou výuku;
- score – hodnoty 0–5, hodnocení výuky, které zadává uživatel po tom, co výuka proběhla;

2.6.4 Tabulka available lessons

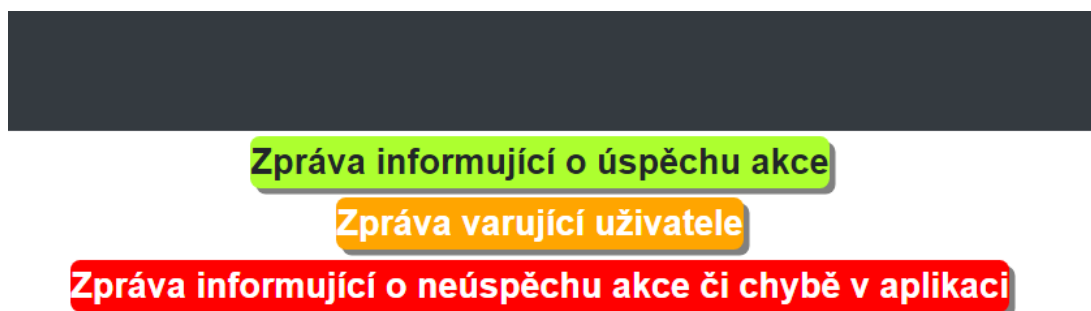
Obsahuje všechny volné hodiny. Při odstranění instruktorského účtu je zavolána akce „CASCADE“, která smaže všechny záznamy v této tabulce, které mají cizí klíč odkazující na odstraňovaný záznam.

- date – PRIMARY KEY – datum, kdy výuka proběhne (vč. hodiny);
- instructor_ID – PRIMARY KEY, FOREIGN KEY - instruktor, který má úsek vypsaný;

2.6.5 Tabulka gdpr

Slouží pro uchovávání souhlasů se zpracováním osobních údajů. Při smazání uživatelského účtu se user_ID nastaví na NULL, ale vše ostatní zůstane nedotčeno, a to díky akci „SET NULL“.

- ID – PRIMARY KEY;
- user_ID – FOREIGN KEY, NULL - uživatel, který udělil souhlas;
- original_user_name – originální přezdívka uživatele, pod kterou se registroval;
- consent – přesné znění souhlasu;
- created_at – datum, kdy byl souhlas udělen;
- revoked_at – NULL – datum, kdy byl souhlas odebrán;



Obrázek 9: Reakce systému zprávami

3 Uživatelská dokumentace

V této sekci popíšeme samotnou aplikaci, její funkce a návod, jak je správně použít. Abych to nemusel vždy explicitně uvádět, **systém reaguje zprávami** na akce uživatele, více v sekci 3.1.1. Pro navigaci v aplikaci slouží intuitivní menu v horní části stránky. Akce v menu se liší podle uživatelských práv a přihlášení uživatele.

3.1 Všeobecná funkcionalita

3.1.1 Zprávy

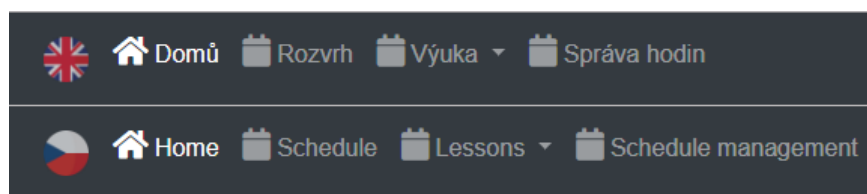
Jelikož je hojně využíváno AJAXu, tak na první pohled nemusí být zřejmé, že se akce opravdu provedla. Z toho důvodu systém dokáže reagovat na většinu uživatelských akcí zprávami. Tyto zprávy jsou zobrazovány hned pod menu v horní části obrazovky. Dělí se do tří kategorií, a to success, warning a error, s příslušným vzhledem, jak můžeme vidět na obrázku 9.

3.1.2 Emailová notifikace

Systém je schopen rozesílat emaily uživatelům, přičemž aktuálně posílá uživateli email pokud:

- uživatel se zaregistruje,
- uživatel zažádá o aktivaci účtu,
- uživatel podá žádost o obnovu hesla,
- uživatel rezervuje výuku.

Obsah emailů a jejich účel budou vždy popsány v následujících sekcích, dochází-li k emailové notifikaci.



Obrázek 10: Přepnutí jazyka v informačním systému

3.1.3 Jazyky

Informační systém umí dva jazyky, češtinu a angličtinu. Všechn text, zprávy, emaily apod. jsou v aplikaci přeloženy a k přepnutí mezi jazyky slouží ikona s vlajkou reprezentující daný jazyk, která se nachází v menu (viz obrázek 10). Systém automaticky překládá i rozesílané emaily. V tomto textu je systém popisován s jazykem nastaveným na češtinu, avšak vše popsáno je funkční i v angličtině.

3.1.4 Registrace

K registraci se lze dostat kliknutím na příslušný odkaz v menu. Stránka se zobrazí pouze nepřihlášeným uživatelům, jelikož funkcionality, kdy by se přihlášený uživatel mohl registrovat, postrádá smysl. Registrace probíhá tradičně uživatel musí vyplnit přihlašovací jméno, email, heslo, osobní údaje (celé své jméno, datum narození, telefonní číslo) a zároveň souhlasit se zpracováním osobních údajů, na které vede odkaz přímo z formuláře. Je nutno dodržet pravidla, která jsou pro jednotlivé vstupy nastavena a srozumitelně sdělena. Po úspěšné registraci je uživateli odeslán aktivační email na emailovou adresu, kterou zadal při registraci.

3.1.5 Aktivace účtu

Registrovaný uživatel musí před přihlášením svůj účet nejdříve aktivovat. K aktivaci dochází po kliknutí na příslušný odkaz v emailu, jež uživatel obdržel při registraci. Vyskytl-li se problém a uživateli email nepřišel, popř. chce-li z nějakého důvodu zaslat aktivační email znovu, pak na stránce s přihlášením stačí kliknout na „aktivovat účet“, vyplnit email a aktivační email bude znovu odeslán.

3.1.6 Přihlášení

Registrovaný uživatel s aktivovaným účtem se může přihlásit na stránce s přihlášením. Po vyplnění přihlašovacího jména a hesla, je uživatel přihlášen, pakliže účet existuje, je aktivní a heslo je správné. V opačném případě je uživatel informován chybovou zprávou o opaku. Po úspěšném přihlášení je přesměrován na domovskou stránku systému.

Rezervace výuky

04.05.2019

Datum	4. 5. 2019	5. 5. 2019	6. 5. 2019	7. 5. 2019	8. 5. 2019	9. 5. 2019	10. 5. 2019
09:00						VOLNO	
10:00							
11:00							
12:00				VOLNO	VOLNO		
13:00				VOLNO	VOLNO		
14:00		OBSAZENO	VOLNO				
15:00			VOLNO				
16:00							
17:00							
18:00							
19:00							
20:00					VOLNO		

Obrázek 11: Rozvrh hodin na počítači

3.1.7 Zapomenuté heslo

Nežřídká se stává, že uživatel zapomene heslo. V takovém případě stačí na stránce s přihlášením kliknout na „obnovit heslo“. Po vyplnění emailu je na zadanou emailovou adresu odeslán email, který uživatele odkáže na stránku, kde může své heslo obnovit. Jestliže email nebyl v databázi nalezen, je uživatel informován chybovou hláškou. Na stránce s obnovou zapomenutého hesla stačí vyplnit nové heslo a potvrdit jej opětovným napsáním do dalšího pole.

3.2 Uživatelská funkcionality

V této sekci budu popisovat funkcionality přihlášeného uživatele s minimálním oprávněním (vyjma sekce 3.2.1).

3.2.1 Zobrazení rozvrhu hodin

Do rozvrhu hodin (viz obrázek 11) se lze dostat kliknutím na příslušný odkaz v menu. V horní části rozvrhu nalezneme změnu data vyhledávání spolu s dalšími filtry pro užší výběr lekcí. Základním filtrem jsou položky „Jednohodinová lekce“ a „Dvouhodinová lekce“, které se vzájemně vylučují (buď je vybráno jedno, nebo na druhé). Uživatel si samozřejmě může objednat dvě hodinové lekce přímo



Obrázek 12: Rozvrh hodin na mobilním zařízení Honor 8

po sobě, avšak druhý zmíněný filtr zobrazí pouze ty hodiny, kde je možné objednat stejného instruktora pro obě po sobě jdoucí lekce (což je žádoucí). Dále si klient filtruje jazyk, a to skrze položky „Čeština“ a „Angličtina“, přičemž tyto položky jsou spolu ve skupině a opět se vzájemně vylučují. Zbývajících 5 filtrů tvoří poslední skupinu, která určuje typ výuky. Vybraný a aktivní filtr je označen modrým pozadím a filtry se aplikují okamžitě po kliknutí. Není to nejšetrnější řešení, co se týče dotazů na databázi, jelikož se při každém kliknutí na filtr musí vyžádat překreslení tabulky s rozpisem. Avšak k tomuto řešení jsem přistoupil z důvodu uživatelské přívětivosti. Při testování, zejména na mobilních zařízeních, se nejednou vyskytl problém, kdy si uživatel neuvědomil nutnost potvrzení filtrů explicitně kliknutím na tlačítko.

Rozvrh se přizpůsobuje různým velikostem zařízení (stejně jako celá stránka), jak můžeme vidět na obrázku 12. Zde přizpůsobení spočívá v počtu zobrazených sloupců (tedy dnů), přičemž posloupnost podle velikosti zobrazovacího zařízení je následující – 7, 5, 3, 2 a 1. Je záhodno zmínit, že tuto funkcionalitu může využít i nepřihlášený uživatel.

3.2.2 Objednávka lekce pro daný čas

Poté, co si uživatel zobrazil rozvrh podle zvolených filtrů, může kliknout na zeleně vyznačené položky, s textem „VOLNO“, značící dostupnost vypsany termínů instruktory. Po kliknutí se pod rozvrhem zobrazí dostupní instruktoři, přičemž

uživatel je na ně ihned přesunut. Vyobrazení instruktoři mají vypsané své jazykové dovednosti a jaký typ výuky mohou vést. Vybrané filtry z rozvrhu zaručují, že instruktor bude schopen vést danou výuku a zvolený typ a jazyk výuky je již předvyplněn před objednávkou. Uživatel stále může změnit jazyk a typ výuky bez toho, aniž by načítal celý rozvrh znovu. Výběr a rezervace je dokončena kliknutím na tlačítko „objednat“. Klient je notifikován úspěšností či případnou neúspěšností rezervace zprávou nad rozvrhem. Neúspěch může nastat, pokusí-li se klient objednat více jak dvě lekce pro stejný datum (den i hodina).

Při úspěšné rezervaci je uživateli odeslán email upozorňující na vytvořenou rezervaci. Email obsahuje detaily výuky, jako je datum, instruktor, typ a jazyk výuky. Zároveň obsahuje odkaz pro zrušení rezervace a opětovný souhlas se zpracováním osobních údajů (a samozřejmě opět odkaz na jeho znění).

3.2.3 Přehled lekcí

Skrze „Účet“ → „Můj přehled hodin“, je možno si prohlédnout lekce, které má uživatel objednané, stejně tak jako lekce proběhlé. Objednané hodiny jsou uživateli zobrazeny při načtení stránky. Pro přepínání mezi objednanými a proběhlými hodinami slouží dvě dobře viditelná tlačítka, která mohou připomínat „záložky“. Obě kategorie jsou stránkovány, přičemž se zobrazuje 10 hodin na stránku. Listovat je možné šipkami ihned pod tabulkou. Pro uživatele menších zařízení, zpravidla mobilních, je zde responzivita¹³ vyřešena přidáním posuvníku pro celou tabulku hodin, nevlze-li se na stránku. Toto řešení bylo zvoleno, jelikož se mi nezdálo moudré skládat hodiny pod sebe, nebo nechat uživatele listovat po jedné hodině. Při testování na menších zařízeních se objevil problém, kdy uživatelé zprvu nevěděli, že lze tabulku takto posouvat. Z toho důvodu jsem při každém načtení tabulky přidal animaci, která uživateli ukáže možnost pohybu tabulky do stran.

Při zobrazení objednaných hodin má uživatel možnost jednotlivě hodiny stornovat kliknutím na tlačítko „storno“. Po kliknutí je dotázán, zda si je jist, jelikož se může stát, že se uživatel překlikne.

Má-li uživatel zobrazeny proběhlé hodiny, nabízí se mu možnost každou z nich ohodnotit. Defaultně¹⁴ hodina nemá žádné hodnocení (hodnocení 0), přičemž kliknutím na hvězdičky uživatel hodnotí danou lekci. Počet hvězdiček včetně té, na kterou uživatel klikne, značí hodnocení jako hodnotu.

3.2.4 Správa účtu

Každý uživatel se v sekci „Můj účet“ může podívat na své osobní údaje. Samozřejmostí je i změna těchto údajů. Pro změnu údajů stačí vyplnit daný vstup (možnost vyplnit i všechny naráz), zadat heslo pro potvrzení a kliknout na tlačítko změnit. Pokud je heslo nesprávné, změna se neprovede a je zobrazena chy-

¹³Responzivita - jak se stránka přizpůsobuje různým zařízením

¹⁴Defaultní hodnota = původní hodnota

bová hláška, avšak vyplněné údaje zůstanou, aby uživatel nemusel vše vyplňovat znovu.

Dle GDPR musí mít uživatel možnost odebrat souhlas se zpracováním osobních údajů. Při odebrání souhlasu je uživateli zneaktivněn účet a jeho osobní údaje jsou vymazány (konkrétně jméno, příjmení, datum narození, email a telefon). K takovému účtu se uživatel znovu nepřihlásí.

3.3 Instruktorská funkcionalita

Od této sekce se bude jednat vždy o funkcionalitu přihlášeného uživatele (v této sekci navíc s asociovaným instruktorským účtem). Instruktoři jsou stále uživatelé, tzn. že pro ně stejně tak platí i uživatelská funkcionalita.

3.3.1 Zápis výuky

Každý instruktor má možnost zapsat si své časové úseky v sekci „Výuka“ → „Mé hodiny“. Zapisování probíhá klikáním na příslušná políčka (den a hodina). Políčko bez podbarvení značí, že pro tento úsek nemá instruktor žádné lekce ani vypsané hodiny. Vypsaná volná hodina, která je viditelná klientům v rozvrhu, je označena modrým podbarvením. Takovou hodinu si instruktor může kliknutím odepsat. Zeleně podbarvené políčko je políčko, na které instruktor klikl a tím má v úmyslu ho zařadit jako volnou hodinu do rozpisu. Takto zbarvené políčko není synchronizováno s databází, tedy ho klienti nevidí. Pro uložení rozpisu je potřeba kliknout na tlačítko „Uložit rozpis“, jinak se změny neprovedou.

Instruktor nemá právo přidávat a odebírat hodiny, které jsou starší, než je aktuální datum a čas. Takové hodiny mu nejsou ani zobrazeny.

Je-li pro nějakou volnou hodinu objednan klient, pak je tato hodina vyznačena červeně s bílým textem, informujícím instruktora o příjmení klienta, jazyku a typu výuky. Takovou hodinu instruktor nemá právo zrušit.

3.3.2 Přehled všech hodin

Instruktoři mají právo nahlédnout do rozpisu hodin všech instruktorů. Jsou zde vyobrazeny volné (modré), obsazené nezaplacené (oranžové) a obsazené zaplacené hodiny (zelené). Slouží pouze jako informace pro instruktory o tom, kteří kolegové budou učit ve stejný den, jaký typ výuky apod. Vzájemně se mohou domluvit na prohození výuky a požádat o to moderátora.

3.4 Funkcionalita moderátora

Moderátor je uživatel, který má právo spravovat všechny hodiny v lyžařské škole. Pro tento účel slouží stránka „Správa hodin“. Takových uživatelů bude v lyžařské škole malé množství, jedná se o vedoucí instruktory, kteří jsou prostředníky mezi klienty a instruktory.

3.4.1 Správa hodin všeobecně

Ve správě hodin jsou zobrazeni všichni instruktoři pro zvolené datum (které lze libovolně měnit). Pod datem se nachází tlačítko pro přidání instruktora do rozpisu, které vyvolá modální okno se vstupem obsahujícím nabídku instruktorů jako našeptávač. Pokud je vybrán instruktor, který už se v rozpisu nachází, pak je zobrazena chybová hláška a nedojde ke změně. Přidání instruktora, který v rozpisu pro daný den ještě není, způsobí jeho přidání se všemi hodinami označenými, jako „VOLNO“. Modální okno zůstane otevřené i po přidání instruktora, a to pro případ kdyby bylo potřeba přidat více instruktorů. Po přidání je tedy potřeba okno zavřít.

K odebrání instruktora z rozpisu slouží červené tlačítko ve sloupci „Odebrat“. Při kliknutí se zobrazí potvrzovací dialog, který když je odsouhlasen, tak odebere instruktora z rozpisu včetně jeho volných vypsáných časových úseků. Instruktora není možné odebrat, jestliže má rezervovanou nějakou hodinu. Takovou hodinu je potřeba nejdříve zrušit.

Po kliknutí na jméno instruktora v rozpisu, či klienta v detailu hodiny, budeme přesměrováni na stránku s informacemi o uživateli. Stránka zobrazuje a poskytuje jiné možnosti z pohledu moderátora a administrátora.

Políčko hodiny může být ve 4 stavech. „PŘIDAT“, „VOLNO“, „OBJEDNÁNO“ a „ZAPLACENO“. Pokud je políčko ve stavu „PŘIDAT“, pak není hodina vypsána v rozpisu, tedy klient si ji nemůže rezervovat. Kliknutím na něj se ovšem změní stav na „VOLNO“.

Políčko ve stavu „VOLNO“ značí hodinu, která je volně viditelná klientům pro objednání lekce. Při kliknutí se zobrazí část, kde je možno objednat klienta, přičemž uživatel je na ni ihned přesměrován. Klienta je potřeba zvolit (je k dispozici našeptávač), stejně tak jazyk a typ výuky. Jazyk a typ obsahují pouze možnosti instruktora. Tlačítkem „Objednat klienta“ se vytvoří rezervace, aktualizuje se celý rozpis a jsou ihned zobrazeny informace o rezervované hodině. Hodinu ve stavu „VOLNO“ lze odepsat tlačítkem „odepsat“, při zvolené volné hodině, nebo poklepáním¹⁵ na hodinu v rozpisu.

Stav „OBJEDNÁNO“ značí objednanou lekci. Při kliknutí na ni se zobrazí informace o lekci s tlačítky pro změnu stavu a stornování. Stornováním se lekce zruší a hodina se vrátí zpět do stavu „VOLNO“. Změnu stavu lze provést zmíněným tlačítkem, nebo poklepáním na hodinu.

Stav „ZAPLACENO“ je víceméně totožný s předchozím stavem. Pouze značí, že hodina byla zaplacená. Takto označená lekce znamená, že klient bude instruktorem akceptován pro výuku.

3.5 Funkcionalita Administrátora

Administrátor přebírá funkcionalitu od moderátora a je to celkový správce systému. Jedná se o jednoho, maximálně dva zaměstnance (resp. vedení) lyžařské

¹⁵dvojitým kliknutím

školy.

3.5.1 Správa účtů

Jako správce má takový uživatel právo libovolně prohlížet a měnit účty všech uživatelů, a to včetně asociovaných instruktorských účtů. Po zvolení uživatele (opět pomůže našeptávač) a potvrzení uživatele na stránce „Správa účtů“, jsou zobrazeny informace o vybraném uživateli, včetně jeho asociovaného instruktorského účtu, má-li nějaký. Změna probíhá obdobně jako na stránce „Můj účet“, tzn. vyplněním příslušných políček (v libovolném množství) a následným kliknutím na tlačítko „Změnit“. Účet je samozřejmě možné taky smazat, což ale není úplně doporučeno, každopádně je to logická funkcionalita.

Uživatelský účet, který nemá asociovaný instruktorskou část je možno o tuto část obohatit kliknutím na tlačítko „Udělat instruktorem“. Tímto a pouze tímto způsobem vznikají instruktoři. Vedení lyžařské školy musí mít licence instruktorů, které je opravňují k vedení daného typu výuky. Proto také pouze správce (jakožto vedení lyžařské školy) má právo instruktory vytvářet a měnit jejich dovednosti v systému. Zároveň je zde možnost odebrání instruktorského účtu, čímž se účet opět stane klasickým uživatelem.

3.5.2 Přehled hodin

Slouží pro celkový přehled nad hodinami daného uživatele. Stačí vybrat uživatele (na pomoc zde máme našeptávač se všemi uživateli v systému), data od kdy do kdy a pořadí, ve kterém se mají hodiny zobrazit. Defaultně jsou nastaveny hodnoty, které vypíše všechny položky pod danou kategorií, seřazené podle data.

Pro zobrazení nám slouží trojice tlačítek (vzhled je stejný, jako v přehledu hodin v „Můj přehled hodin“). Při nezadání uživatele, zadání neexistujícího uživatele či uživatele, jež nemá instruktorský účet, je přímo pod vstupem zobrazena chybová hláška, upozorňující na špatný vstup.

Tlačítko „Hodiny uživatele“ zobrazí všechny hodiny, které má uživatel jako klient, a to jak proběhlé, tak rezervované. Zde už záleží na uživateli, aby si zvolil data a tyto dva druhy rozlišil.

„Hodiny instruktora“ nám zobrazí všechny hodiny, které si klienti rezervovali pro uživatele, jako instruktora, či je odučil jako instruktor. Opět se zde implicitně nerozlišuje mezi proběhlými a rezervovanými.

Při kliknutí na „Instruktorovi vypsané“ se vypíše všechny hodiny, které má uživatel vypsané jako instruktor, tedy jsou volně viditelné pro klienty. Tato funkce slouží spíše pro přehled, kdy je daný instruktor vypsán. Lze to samozřejmě zjistit a upravit ve správě hodin, avšak tamější UI není přizpůsobeno na přehled, nýbrž zejména na funkcionalitu.

Závěr

Výsledkem práce je informační systém, který nyní disponuje uživatelskými účty s veškerou asociovanou funkcionalitou, instruktorskými účty a samozřejmě účty pro správu jak uživatelů, instruktorů, tak samozřejmě objednaných hodin, přičemž tato správa není nikterak náročná. Klienti si mohou objednávat výuku podle kritérií a sledovat své lekce. Samotná funkcionalita a obsluha systému bude ověřena praxí, jelikož systém bude nadcházející zimu používán v lyžařské škole. Existují vylepšení, která mohou být přidána, jako např. detailnější hodnocení výuky, věrnostní program (jedná se spíše o věc lyžařské školy), upozornění na nadcházející výuku (např. den předem) emailem nebo skrze SMS a další.

Na začátku textu je uvedena motivace pro tvorbu této bakalářské práce a lehké uvedení do problematiky. Následuje popis použitých technologií s nástinem jejich použití při tvorbě aplikace a samozřejmě nechybí ani uživatelská dokumentace.

Conclusions

As the result we have an information system, which dispose of user and instructor accounts with all associated functionality, as well as administration accounts. These administration accounts are also privileged to manage lessons in an easy way. Client may book lessons as well as track them. The system itself shall be proven worthy by practice, since it will be deployed in the upcoming winter season in a ski school. It can be improved by adding or extending functionality, such as more detailed evaluation of lessons, adding a loyalty program (depends on the ski school), making clients notified about the upcoming lesson (e. g. the day before) by email or SMS and more.

The motivation for creation of this bachelor's thesis is summed up at the beginning of the thesis together with a brief introduction into problematics. The following section is dealing with used technology and its usage. User friendly documentation is included at the end of the thesis.

A Příloha: Instalace a spuštění aplikace

A.1 Hosting a HTTPS certifikát

Již při psaní této práce jsem aplikaci umístil na hosting s placeným tarifem. Zároveň jsem nechal vystavit HTTPS certifikát, který zabezpečuje komunikaci a budí větší důvěryhodnost. Díky úpravě souboru `.htaccess` vždy proběhne přesměrování na protokol HTTPS [27] spolu s „www.“ Aplikace je dostupná na vlastní doméně <https://www.funschool.cz>.

A.2 Požadavky

Jelikož se jedná o webovou aplikaci, existují zde požadavky na server:

- Apache server,
- MySQL server,
- PHP ve verzi 5.6 a vyšší (s verzí 7.3 nemusí fungovat správně),
- splňovat požadavky na provoz Nette Frameworku ve verzi 2.4 (viz <https://doc.nette.org/cs/2.4/requirements>).

Tyto požadavky lze v lokálním prostředí splnit instalací správného LAMPP balíčku. Já jsem při vývoji na lokálním prostředí použil balíček XAMPP ve verzi 7.2.8.

Požadavky na klienta jsou minimální, a to moderní webový prohlížeč. Zejména se jedná o Google Chrome, Mozilla Firefox nebo Opera, ve verzích, které jsou stabilní a podporované.

A.3 Instalace

Instalace má následující kroky:

- vytvoření databáze (ideálně s porovnáváním „utf8_general_ci“),
- importování souboru `SQL.sql` – je potřeba vypnout kontrolu cizích klíčů (vytvoří tabulky a vloží předpřipravená data),
- upravit přístupové údaje do vytvořené databáze v souboru `config/config.local.neon`.

Do složky webového serveru přepokopírovat data ze složky `/bin`.

A.4 Spuštění

Pokud je vše správně provedeno, aplikace bude dostupná na URL `localhost/fun-school/www`.

B Obsah příloženého CD/DVD

bin/

Kompletní adresářová struktura webové aplikace (v ZIP archivu) pro zkopírování na webový server. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový provoz webové aplikace na webovém serveru.

doc/

Složka obsahuje text práce ve formátu PDF a ZIP archiv se všemi potřebnými soubory pro vygenerování tohoto PDF dokumentu.

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.

Navíc CD/DVD obsahuje:

data/

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

Literatura

- [1] KOUBA, Tomáš. *NET MAGNET: Jak na GDPR na webu – praktický návod* [online]. [cit. 2019-5-4]. Dostupný z: <https://www.netmagnet.cz/blog/gdpr/>.
- [2] SHOPTET. *Univerzální podmínky ochrany osobních údajů* [online]. [cit. 2019-5-4]. Dostupný z: <https://www.shoptet.cz/univerzalni-podminky-ochrany-osobnich-udaju/>.
- [3] NETTE, Foundation. *Nette Framework: Presenters* [online]. [cit. 2019-4-30]. Dostupný z: <https://doc.nette.org/cs/2.4/presenters>.
- [4] TRNEČKA, Martin. *KMI/INFS Webové technologie: Jazyk PHP - Odbočka: Model, view, controller (MVC) v PHP* [online]. [Cit. 2019-4-30]. Dostupný z: <https://www.dropbox.com/s/kzcnuo4rh9m5b4n/infs05.pdf?dl=0>.
- [5] NETTE, Foundation. *Nette Framework* [online]. [cit. 2019-4-30]. Dostupný z: <https://nette.org/cs/>.
- [6] THE PHP, Group. *PHP* [online]. [cit. 2019-5-12]. Dostupný z: <https://php.net/>.
- [7] NETTE, Foundation. *Nette Framework: Seznámení s Nette Frameworkem* [online]. [cit. 2019-4-30]. Dostupný z: <https://doc.nette.org/cs/2.4/getting-started>.
- [8] NETTE, Foundation. *Nette Framework: Latte* [online]. [cit. 2019-5-1]. Dostupný z: <https://latte.nette.org/>.
- [9] NETTE, Foundation. *Nette Framework: Zabezpečení před zranitelnostmi* [online]. [cit. 2019-5-1]. Dostupný z: <https://doc.nette.org/cs/2.4/vulnerability-protection>.
- [10] NETTE, Foundation. *Nette Framework: Tracy* [online]. [cit. 2019-5-1]. Dostupný z: <https://tracy.nette.org/>.
- [11] NETTE, Foundation. *Nette Framework: Database* [online]. [cit. 2019-4-30]. Dostupný z: <https://doc.nette.org/cs/2.4/database>.
- [12] THE PHP, Group. *PHP: The PDO class* [online]. [cit. 2019-4-30]. Dostupný z: <https://www.php.net/manual/en/class.pdo.php>.
- [13] KDYBY, CMS. *KDYBY: Translation* [online]. [cit. 2019-5-1]. Dostupný z: <https://github.com/Kdyby/Translation/blob/master/docs/en/index.md>.
- [14] NETTE, Foundation. *Nette Framework: Práce s formátem NEON* [online]. [cit. 2019-5-1]. Dostupný z: <https://doc.nette.org/cs/2.4/neon>.
- [15] NETTE, Foundation. *Nette Framework: Formuláře* [online]. [cit. 2019-5-2]. Dostupný z: <https://doc.nette.org/cs/2.4/forms>.
- [16] NETTE, Foundation. *Nette Framework: Hashování hesel* [online]. [cit. 2019-5-2]. Dostupný z: <https://doc.nette.org/cs/2.4/passwords>.

- [17] DOBEŠ, Vojtěch. *nette.ajax.js* [online]. [cit. 2019-5-2].
Dostupný z: <https://componette.com/vojtech-dobes/nette.ajax.js/>.
- [18] NETTE, Foundation. *Nette Framework: AJAX & Snippetty* [online]. [cit. 2019-5-2]. Dostupný z: <https://doc.nette.org/cs/2.4/ajax>.
- [19] CASTRO, Elizabeth; HYSLOP, Bruce. *HTML5 a CSS3: Názorný průvodce tvorbou WWW stránek*. First. Brno: Computer Press, 2012. 441 s.
- [20] TRNEČKA, Martin. *KMI/INFS Webové technologie: JavaScript* [online]. [Cit. 2019-4-30].
Dostupný z: <https://www.dropbox.com/s/6oxxakn2qpe2egh/infos03.pdf?dl=0>.
- [21] TRNEČKA, Martin. *KMI/INFS Webové technologie: AJAX* [online]. [Cit. 2019-4-30].
Dostupný z: <https://www.dropbox.com/s/wo8oxn2cner4llq/infos07.pdf?dl=0>.
- [22] GILMORE W., Jason. *Beginning PHP and MySQL: From Novice to Professional*. Third. 2018. 1080 s.
- [23] BOOTSTRAP, team. *Bootstrap* [online]. [cit. 2019-5-1].
Dostupný z: <https://getbootstrap.com/>.
- [24] JQUERY, Foundation. *Jquery* [online]. [cit. 2019-5-1].
Dostupný z: <https://jquery.com/>.
- [25] W3SCHOOLS. *JavaScript: HTML DOM* [online]. [cit. 2019-5-1]. Dostupný z: https://www.w3schools.com/js/js_htmldom.asp.
- [26] APACHE, Software Foundation. *Netbeans* [online]. [cit. 2019-5-1]. Dostupný z: <https://netbeans.org/>.
- [27] ENDORA, Freehosting. *Jak mám spustit svůj web jen na HTTPS?* [online]. [cit. 2019-5-6]. Dostupný z: <https://www.endora.cz/podpora/napoveda/3-technicke-ftp-mysql/157-jak-mam-spustit-svuj-web-jen-na-https>.
- [28] ERDPLUS. [online]. [cit. 2019-5-6]. Dostupný z: <https://erdplus.com/>.
- [29] NETTE, Foundation. *Nette Framework: Documentation* [online]. [cit. 2019-5-6].
Dostupný z: <https://doc.nette.org/cs/2.4/>.