

# **Návrh nízkonákladové alternativy k RFID řešením pro řízení přístupu, identifikaci, tvorbu rezervací a elektronické platby v budově fitness centra**

Bc. Martin Sedlář



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2013/2014

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin SEDLÁŘ**  
Osobní číslo: **A11461**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Integrované systémy v budovách**  
Forma studia: **prezenční**

Téma práce: **Návrh nízkonákladové alternativy k RFID řešením pro řízení přístupu, identifikaci, tvorbu rezervací a elektronické platby v budově fitness centra**

Zásady pro vypracování:

1. Prostudujte stávající komerční systémy vhodné pro řízení přístupu z hlediska identifikace, tvorby rezervací a elektronických plateb v budovách fitness center.
2. Uvažujte aplikaci systému na konkrétní provozovnu Spinning centra ve Zlíně a shrňte požadavky na systém.
3. Srovnajte komerční systémy z hlediska nákladů a zhodnoťte náklady na realizaci vlastního systému dle získaných požadavků.
4. Navrhněte klient-server informační systém pro řízení agendy provozovny fitness centra, zohledňující výše uvedené požadavky.
5. Proveďte testovací instalaci v reálném provozu a ověřte funkčnost systému.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KOFLER, Michael a Tomáš SUCHÁNEK. Mistrovství v MySQL 5: vztahy a pomůcky. Vyd. 1. Překlad Jan Svoboda, Ondřej Baše, Jaroslav Černý. Brno: Computer Press, 2007, 805 s. ISBN 978-80-251-1502-2.
2. GUTMANS, Andi a Tomáš SUCHÁNEK. Mistrovství v PHP 5: vztahy a pomůcky. Vyd. 2. Brno: Computer Press, 2007, 655 s. ISBN 978-80-251-1519-0.
3. LAVIN, Peter. PHP – objektově orientované: koncepty, techniky a kód. 1. vyd. Praha: Grada, 2009, 211 s. ISBN 978-80-247-2137-8.
4. PROCHÁZKA, David. CSS a XHTML: tvorba dokonalých WWW stránek krok za krokem. 2., aktualiz. vyd. Praha: Grada, 2011, 175 s. Průvodce (Grada). ISBN 978-80-247-3897-0.
5. ŠKULTÉTY, Rastislav. JavaScript: programujeme internetové aplikace. Vyd. 1. Praha: Computer Press, 2004, 224 s. ISBN 80-251-0144-4.

Vedoucí diplomové práce:

**Ing. Radek Vala**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**7. března 2014**

Termín odevzdání diplomové práce:

**11. června 2014**

Ve Zlíně dne 28. února 2014

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.  
~~že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.~~

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Cílem této diplomové práce bylo vytvořit webovou aplikaci, která by byla využita jako rezervační systém pro spinning centrum Mad Jack ve Zlíně. Ta bude mít za úkol vkládání a zobrazování spinning lekcí, identifikaci klientů a možnost přihlášení a rezervací klientů na vypsání lekce. Doplnkem pak mají být elektronické platby za tyto rezervace a jejich přehled pro obsluhu. Aplikace je tvořena v jazyce PHP s prezentací v HTML s použitím Frameworku Nette. Pro ukládání dat byla použita databáze MySQL.

Klíčové slova: Webová aplikace, PHP, Nette, MySQL, elektronické platby, rezervace

## **ABSTRACT**

The main goal of this diploma was to create web application, which will be used as reservation system for Spinning Center Mad Jack in Zlin. This application should be using for Adding and showing lections, klient identifikation and klient reservation at lections. Additional tool should be possibility of electronic payments for reservations, and time sorting of these informations. Application is created in PHP language in HTML language presentation with using Nette framework and MySQL database.

Keywords: Web application , PHP, Nette, MySQL, electronic payments, reservations

Motto:

*„Život je těžká zkouška a jen ten, kdo jeho výzvu přijme, opravdu ví, co znamená žít“*

# OBSAH

<b>ÚVOD.....</b>	<b>10</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>11</b>
<b>1 POŽADAVKY NA SYSTÉM .....</b>	<b>12</b>
1.1 POŽADAVKY Z POHLEDU KLIENTA .....	12
1.1.1 Přístupnost systému .....	12
1.1.2 Jednoduché intuitivní rozhraní .....	12
1.1.3 Možnosti elektronických plateb .....	12
1.1.4 Možnosti hotovostních plateb .....	12
1.1.5 Jednoduchý přehled o probíhajících lekcích .....	13
1.1.6 Rezervace na lekci a její storno .....	13
1.1.7 Přehled o výši zůstatku a pohybech na účtu .....	13
1.1.8 Zabezpečení uživatelských účtů .....	13
1.1.9 Identifikace a autorizace uživatele .....	13
1.1.10 Zabezpečení .....	14
1.2 POŽADAVKY Z POHLEDU PROVOZOVATELE .....	14
1.2.1 Přístupnost systému .....	14
1.2.2 Jednoduché intuitivní rozhraní .....	14
1.2.3 Možnosti elektronických plateb .....	14
1.2.4 Možnosti hotovostních plateb .....	15
1.2.5 Ruční rezervace klientů a jejich storno .....	15
1.2.6 Náhled do uživatelských účtů a manipulace s nimi .....	15
1.2.7 Ruční přidávání klientů, jejich zrušení a zamčení .....	15
1.2.8 Vkládání lekcí a úprava jejich parametrů .....	15
1.2.9 Zabezpečení .....	15
1.2.10 Automatická údržba systému .....	16
<b>2 STÁVAJÍCÍ KOMERČNÍ REZERVAČNÍ SYSTÉMY .....</b>	<b>17</b>
2.1 KONKRÉTNÍ KOMERČNÍ SYSTÉMY A JEJICH FUNKCE .....	17
<b>3 IDENTIFIKACE KLIENTA .....</b>	<b>20</b>
3.1 MOŽNOSTI IDENTIFIKACE .....	20
3.1.1 Biometrické metody identifikace .....	20
3.1.2 Nebiometrické metody identifikace .....	21
<b>4 DATABÁZE .....</b>	<b>25</b>
4.1 DATABÁZE A ZPŮSOB ULOŽENÍ DAT .....	25
4.2 MYSQL .....	25
4.3 NÁVRH DATABÁZE .....	25
4.3.1 Zásady pro návrh databází .....	25
4.3.2 Ukládání dat pod správnými datovými typy .....	26
4.3.3 Každý řádek by měl obsahovat jednoznačný identifikátor .....	26
4.3.4 Nežádoucí duplicita dat .....	26
4.3.5 Relace .....	27
4.3.6 Pevně daná struktura databáze .....	28

4.3.7	Kompaktní struktura.....	28
4.4	DATOVÉ TYPY .....	28
<b>5</b>	<b>WEBOVÁ APLIKACE.....</b>	<b>30</b>
5.1	ARCHITEKTURA MCV .....	30
5.1.1	Princip .....	30
5.2	PHP.....	31
5.3	NETTE .....	31
5.3.1	Latte.....	32
5.3.2	Laděňka .....	32
5.3.3	Firebug .....	32
5.3.4	Logování chyb .....	33
5.3.5	Zabezpečení aplikace .....	33
5.4	JAVASCRIP T .....	33
5.4.1	jQuery.....	33
<b>6</b>	<b>BEZPEČNOST WEBOVÝCH APLIKACÍ.....</b>	<b>35</b>
6.1	CROSS-SITE-SCRIPTING .....	35
6.2	CROSS-SITE REQUEST FORGERY (CSRF) .....	35
6.3	SQL INJECTION .....	36
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>37</b>
<b>7</b>	<b>DATABÁZOVÁ ČÁST .....</b>	<b>38</b>
7.1	STRUKTURA DATABÁZE .....	38
<b>8</b>	<b>PRAKTICKÁ IMPLEMENTACE .....</b>	<b>40</b>
8.1	MODUL ADMIN.....	40
8.1.1	Přihlášení.....	40
8.1.2	Přehled.....	40
8.1.3	Uživatelé .....	41
8.1.4	Lekce .....	43
8.1.5	Platby.....	45
8.1.6	Ceny .....	47
8.1.7	Kredity.....	47
8.1.8	Ruční CRON .....	48
8.2	MODUL USER .....	49
8.2.1	Přehled.....	51
8.2.2	Moje Lekce.....	51
8.2.3	Můj profil .....	53
8.2.4	Účet .....	54
8.3	IDENTIFIKACE UŽIVATELE .....	56
8.4	AUTOMATICKÁ ÚDRŽBA APLIKACE .....	56
8.4.1	Překlopení rezervací do plateb .....	56
8.4.2	Lekce již proběhlé .....	57
8.4.3	Neaktivní uživatelé.....	57



8.5	ROZPOČET NAŠÍ APLIKACE.....	57
8.6	TESTOVACÍ PROVOZ .....	58
<b>ZÁVĚR .....</b>		<b>59</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>		<b>61</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>		<b>63</b>
<b>SEZNAM OBRÁZKŮ .....</b>		<b>64</b>
<b>SEZNAM TABULEK.....</b>		<b>65</b>

## ÚVOD

Žijeme v době, kdy jsou moderní technologie naším každodenním chlebem. Často je to důsledkem snahy člověka zjednodušit si každodenní činnosti, a to i tak jednoduché, jakým je sport. Tento trend vnořování se technologií do všech našich činností nabývá velkých rozměrů.

Také internet a připojení na web je dnes naprostou samozřejmostí, a to doslova všude, doma i na cestách. Proto se pokusíme spojit výhody internetového připojení a nutnost rezervace a elektronických plateb.

Práce se zaměří na teoretický, ale i praktický vývoj webové aplikace typu klient-sever s jednoduchými funkcemi, jež by měli lidem pomoci si rychle a jednoduše zamluvit své místo na dané lekci u určitého lektora za danou cenu.

V teoretické části si rozebereme požadavky na aplikaci na základě konzultace s provozovatelem spinning centra. Dále si uděláme průzkum stávajících komerčních aplikací s podobným zaměřením z hlediska funkcí i nákladů. Dále pak se podíváme možnosti identifikace klienta pro práci s aplikací a zvolíme ideální variantu pro naši aplikaci. Nakonec si nastíníme použité technologie pro vývoj a běh aplikace. Kapitola je věnována i bezpečnostním rizikům v oblasti webových aplikací a jejich řešení.

Praktická část popisuje samotný vývoj aplikace. Popisuje její běh, funkce a možnosti, a to jak v aplikační části, tak v části databázové. Zde se věnujeme především struktuře aplikace a databáze. Byl proveden i testovací provoz, aby byla ověřena funkcionality a odladěny případné chyby. Nakonec jsou shrnuty náklady na vývoj aplikace a srovnány s komerčními řešeními.

## **I. TEORETICKÁ ČÁST**

## **1 POŽADAVKY NA SYSTÉM**

Tato práce pojednává o rezervačním systému spinning centra Mad Jack ve Zlíně, a tak si v této kapitole rozebereme, jaké jsou požadavky na takový systém, a to jak z hlediska klienta spinning centra, tak z hlediska jeho provozovatele. Budeme vycházet především z informací na základě konzultace s provozovatelem tohoto spinning centra.

### **1.1 Požadavky z pohledu klienta**

V této kapitole budou shrnuty požadavky na aplikaci z pohledu klienta.

#### **1.1.1 Přístupnost systému**

Vzhledem k dnešní moderní době, kdy se využívá informační technologie ve velkém měřítku, měl by být přístup rezervačního systému odkudkoliv a kdykoliv. Mnoho z nás vlastní chytré mobilní telefony s permanentním připojením na internet a tudíž, není žádný problém se přihlásit do prohlížeče, kde lze jednoduše najít vše snadno a rychle. Z tohoto důvodu, je ideální volbou webová aplikace.

#### **1.1.2 Jednoduché intuitivní rozhraní**

Aby byl rezervační systém jednoduchý a na telefonu i obrazovce počítače dobře zobrazitelný, bylo zvoleno jednoduché rozložení. Díky intuitivnosti celého systému, každý uživatel vše zvládne jednoduše a rychle, jak na svém počítači, tak i na svém mobilním telefonu, přestože ten má omezené možnosti zobrazení.

#### **1.1.3 Možnosti elektronických plateb**

Pro usnadnění možnosti zaplacení jednotlivých lekcí je nabídnuta možnost elektronických plateb. Tato možnost má zvýšit rychlost odbavování jednotlivých cvičících, neboť platbu provedou předem z pohodlí svého domova. Tím pádem před danou lekcí pouze prokáží svou totožnost a mohou se jít nerušeně připravovat na své cvičení.

#### **1.1.4 Možnosti hotovostních plateb**

Možnosti hotovostní platby má klient možnost využít přímo na pokladně ve spinning centru a to buď ve formě dobítí kreditu na svůj účet, anebo přímou jednorázovou platbou za

služby, jako například lekce. Proto by měla být kromě automatizované platby v systému možnost i zvolit si tento druh úhrady za lekce.

### **1.1.5 Jednoduchý přehled o probíhajících lekcích**

Hlavním úkolem tohoto rezervačního systému by mělo být zpřístupnění kalendáře akcí jednotlivým klientům. Systém by měl uchovávat všechny probíhající lekce spolu s ukazatelem obsazenosti, díky které má klient rychlý přehled o nabízených lekcích, jejich ceně, době trvání a jména lektora.

### **1.1.6 Rezervace na lekci a její storno**

Samozřejmostí je přihlášení do volných pozic na jednotlivé lekce a automatické provedení platby. V případě, že bude klient chtít provést storno své rezervace, měl by mít možnost využít storno systému a navrácení platby na kreditový účet.

### **1.1.7 Přehled o výši zůstatku a pohybech na účtu**

Protože si provozovatel přál použití koncepce předplacených prostředků, ze kterých jsou mu odečteny poplatky při rezervacích za jednotlivé lekce, musí mít klient přehled o těchto poplatcích a jejich historii.

V případě, že se klient chce přihlásit na lekci a nemá na svém kreditovém účtu dostatečnou výši pro uhrazení lekce, musí být mu nabídnuta možnost dobít potřebnou finanční částku a to okamžitě pomocí elektronického platebního portálu.

### **1.1.8 Zabezpečení uživatelských účtů**

Protože by měla aplikace obsahovat osobní informace jednotlivých klientů a zároveň jim bude dávat možnost operovat se zůstatky na účtu, musí být aplikace zabezpečena proti cizímu vniknutí. Zde je nutný využít některý ze standartních přihlašovacích formulářů webových aplikací pomocí uživatelského jména a hesla. Aplikaci samu pak je nutné zabezpečit proti narušení.

### **1.1.9 Identifikace a autorizace uživatele**

Každá lekce má pouze omezenou kapacitu, proto mohou nastat nesrovnalosti v obsazení míst na hodině spinningu. Zde je nutné jasně odlišit klienty, kteří jsou na lekci přihlášení a

kteří ne. Proto je nutné zavést identifikaci klienta například pomocí některého zvoleného unikátního identifikátoru, nebo sady identifikátorů. Dále je nutné samotné ověření identity klienta, tzv. autorizace. Autorizace se provádí některou doplňkovou informací, známou pouze uživateli samotnému.

#### **1.1.10 Zabezpečení**

Díky uchovávání citlivějších dat v aplikaci a práce s platebními prostředky je vyžadována určitá úroveň zabezpečení, aby nebyly osobní informace klienta přístupné široké veřejnosti.

### **1.2 Požadavky z pohledu provozovatele**

Tato kapitola nám přiblíží, co by měla aplikace obsahovat z pohledu provozovatele.

#### **1.2.1 Přístupnost systému**

Pro provozovatele spinning centra je důležitý přehled o dění v jeho spinning centru, a to odkudkoliv a kdykoliv. Tento požadavek naprosto splňuje využití webové aplikace, která je dostupná odkudkoliv, kde je přístup k internetu. Možnosti webového rozhraní můžeme využít jak na notebooku, stolním počítači a mnoho dalších elektronických zařízeních.

#### **1.2.2 Jednoduché intuitivní rozhraní**

Pro jednoduché zvládnutí obsluhy a orientace v této aplikaci bylo využito intuitivní rozhraní. Díky tomuto rozhraní se uživatel rychle zorientuje a není třeba žádného speciálního školení pro jeho obsluhu, pouze základní instruktáž.

#### **1.2.3 Možnosti elektronických plateb**

Jedním z největších kladů této aplikace je elektronická verze plateb, která probíhá rychle a okamžitě z domova, bez nutnosti přičinění se lidského faktoru. Tímto se především zabráňuje chybám při manipulaci obsluhy, která tím pádem nemusí zasahovat do úprav platebních procesů, nebo do něj zasahuje jen minimálně. Dalším požadavkem, který se nabízí ve spojení s elektronickými platbami je rychlý a přesný přehled o výtěžnosti jednotlivých lekci, ale i celkových příjmů spinning centra za určené období.

#### **1.2.4 Možnosti hotovostních plateb**

V případě, že si klient nepřeje využívat možností svého kreditového účtu, musí mít možnost uhradit lekci přímo na místě. Pokud si přeje využívat klientského systému, avšak nedůvěřuje elektronickým platebním portálům, může si i svůj kreditový účet dobít přímo na pokladně u příslušného pracovníka, proto jsou tyto funkce v administrační části nutností.

#### **1.2.5 Ruční rezervace klientů a jejich storno**

Požadavkem je též manuální rezervace libovolného klienta operátorem bez jeho přičinění a storno jeho rezervace, v případě telefonické rezervace.

#### **1.2.6 Náhled do uživatelských účtů a manipulace s nimi**

Správce systému by měl mít možnost nahlédnout do informací o jednotlivých klientech a stavech jejich kreditových účtů. V případě nutnosti tyto klienty upozornit na nesrovnalosti a zrušení lekcí například z důvodu nemoci lektora. Možnost změny osobních údajů jednotlivých klientů by měla být také k dispozici, tak jako jejich povýšení do statusu lektora/administrátora.

#### **1.2.7 Ruční přidávání klientů, jejich zrušení a zamčení**

Obsluha by měla mít možnost založení uživatelského účtu přímo v administraci, tak jako i jeho smazání či pozastavení.

#### **1.2.8 Vkládání lekcí a úprava jejich parametrů**

Celý provoz spinning centra se točí především kolem lekcí v určitém termínu a jejich kalendářním výpise. Zde je jasná nutnost vkládání, úprav těchto údajů a jejich mazání, chceme-li, jejich zrušení. Vkládání lekcí a úpravy jejich údajů zde budou na denním pořádku, tak jako bude nutné informovat klienty v případě změn, pokud budou na lekci přihlášení. Všechny tyto funkce jsou v požadavcích provozovatele a musí být prioritně splněny.

#### **1.2.9 Zabezpečení**

Stejně jako v části uživatelské, je nutné zabezpečit přístup do aplikace. V případě administrátorské sekce je tato potřeba o něco žádanější, protože administrátorský modul dává uživateli nejen k dispozici data všech klientů, ale také možnost jejich úpravy.

### 1.2.10 Automatická údržba systému

Aby se v aplikaci nehromadily zbytečné data, je potřeba, aby se systém sám zbavoval nepotřebných údajů. Zde by se měly rušit dlouhodobě neaktivní klientské účty, proběhlé lekce a rezervace, nebo správa plateb.



## 2 STÁVAJÍCÍ KOMERČNÍ REZERVAČNÍ SYSTÉMY

Dnes najdeme na trhu velkou spoustu komerčních rezervačních systémů, které poskytují řadu funkcí. Nicméně častokrát příliš robustních a cenově nedostupných parametrů. Není výjimkou startovací poplatek a dále navíc měsíční sazba za licenci softwaru, což může znamenat pro menší provozovatele, jako je právě naše sportovní zařízení, náklad i v desítkách procent celkového zisku.

### 2.1 Konkrétní komerční systémy a jejich funkce

Pro srovnání autor práce vybral tři nejběžnější komerční online rezervační systémy, které jsou dostupné na webu. Každý ze systémů je šířen v několika verzích a to dle použití od základních, až pro nejvyšší verze s pokročilými funkcemi, jako je například zasílání sms zpráv, nebo vytváření permanentek.

#### **Fitness-rezervace.cz**

Tento systém je ze všech třech pravděpodobně nejpoužívanější. Je přímo určen pro rezervace ve sféře fitness center, posiloven a pohybových aktivit. Jak předesílá výrobce na svých propagačních stránkách, tak jeho výhodami jsou, kromě klasických výhod online rezervačních systémů, také zasílání sms o změnách lekcí, tisk permanentek, úspora času a financí a statistiky návštěv. Za zmínku jistě stojí i možnost komunikace do sociálních sítí, jako google+, facebook, nebo twitter.

Cena je pak nastavena dle verze. Základní verze, jsou omezeny počtem registrovaných uživatelů, počtem rezervací a nelze je využívat na vlastní doméně. Tyto verze jsou placeny paušálně a řádek několika stovek korun měsíčně a to po celou dobu využívání systému. Jednorázová cena je pak dostupná pouze pro nejvyšší verzi „Unlimited“, která je funkcemi, počtem uživatelů i počtem rezervací neomezená, ale cena zřízení, včetně instalace a zavedení se pohybuje za hranicí 20 000 korun. [1]

#### **SuperSaaS**

Tento rezervační systém je též velmi oblíbený. Jeho design a konstrukce je cílena nejen na fitness a pohybové aktivity, ale i na další rezervační možnosti, jako rezervace u lékaře, re-

zervace jednacích místností atp. Na propagačních stránkách se dozvíme výhody systému a to následující: Komunikace prostřednictvím sms, emailu, internetové platby, bezpečnost, spolehlivost a další. Za nevýhody bych pak označil zavádění systému. Díky své universálnosti a robustnosti pro použití v různých druzích služeb, je instalace a individuální nastavení mnohem zdlouhavější. Proškolení personálu složitější i samotná orientace v programu je pro uživatele složitější. Objektivně se mi zdá SuperSaaS, ve srovnání s dříve uvedeným „Fitness-Rezervace“ o něco méně přehledný, a základní design poněkud strohý a nudný, i když výrobce udává téměř nekonečné možnosti individualizace.

Cena SuperSaaS se opět pohybuje dle verze, počtu registrací, rezervací. SuperSaaS však oficiálně dle ceníku nenabízí variantu jednorázové platby, ale pouze paušální měsíční platby za využívání programu. Nicméně je zde dostupná verze Free, která je bezplatná, sice velmi omezená na 50 registrovaných uživatelů a 50 rezervací, ale jistě jsou aplikace, které by se spokojily i s tímto měsíčním minimálním vytížením. Nejdražší a opět neomezená verze, zde nazvaná „Varianta E“ je poskytována za měsíční poplatek 750 korun. [2]

## **iSport**

Třetím a posledním rezervačním systémem, která vybral autor práce ke srovnání je rezervační online systém iSport. Výhody systému iSport vyzdvihuje výrobce hlavně v oblasti ovládání a práce s administrací. Na oficiálních stránkách se dočteme, že systém má snadné a intuitivní ovládání pro obsluhu, oznamování a potvrzování obsluhy, statistiky a reporty. Komunikace pomocí sms, permanentky a BlackList, zajímavá funkce vedení neomluvených absencí. Je zde uvedeno také propojení na sociální sítě, i když v demoverzi jsem se s touto funkcí nesetkal.

Cena iSport rezervačního systému je rozvržena do tří verzí, které jsou rozděleny opět hlavně dle počtu rezervací a počtu typů provozovaných lekcí. Každá verze se nabízí nejen za cenu měsíčního paušálního poplatku, ale i za cenu jednorázovou, která, za nejvyšší dostupnou verzi, činí přijatelných 6000 korun. Tu samou verzi může provozovatel užívat za 700 korun měsíčně. Je možná i adaptace vícejazyčné verze, nicméně za příplatek 1500 korun. [3]

Název	Cena /měsíc	Maximální počet budoucích rezervací	Reklamy
<b>SuperSaaS</b>	750 Kč	1500	Bez reklam
<b>Fitness-rezervace.cz</b>	500 Kč	neomezeně	Bez reklam
<b>iSport</b>	700 Kč	neomezeně	Bez reklam

Tabulka 1 – Ceny nejpoužívanějších komerčních rezervačních online systémů

[1, 2, 3]

Všechny uvedené rezervační systémy měly spoustu společných funkcí. Funkce kalendáře, a denního přehledu najdeme v každém z nich. Další společnou funkcí byl kreditový systém a elektronické platby. Nezbytností byly i přehledy rezervací, cen a další statistiky, které jsou nezbytné pro přehled provozovatele. Pro komfort klientů pak notifikační systém pomocí emailu a sms. Spousta funkcí se ale naopak zdají zbytečně robustní a pro naši aplikaci pouze na spinning centrum celkem zbytečné. Nevyužity by zůstaly funkce jako Last-minute slevy, či permanentky.

Při projektování systému si vzal autor inspiraci z těchto komerčních řešení a po konzultaci s provozovatelem spinning centra Mad Jack Zlín, sestavil rezervační systém co nejvíce na míru dle jeho požadavků.

### 3 IDENTIFIKACE KLIENTA

Protože náš systém bude shromažďovat osobní informace o klientech spinning centra a zároveň bude uchovávat data a jistých platebních prostředcích, je nutné zabezpečit a oddělit jednotlivé účty uživatelů. Nicméně bude nutný i uživatelský přístup do systému a za tímto účelem bude zvolená identifikace dle potřeby.

Další nutností identifikace bude v našem případě identifikace klienta přímo ve spinning centru na lekci, kdy v případě nesrovnalostí v obsazenosti lekce, je potřebné vyloučit z lekce osoby, které nemají lekci rezervovanou nebo zaplacenou.

#### 3.1 Možnosti identifikace

Dnešní moderní technologie nabízí spoustu možností identifikace, které lze využít v široké oblasti zaměření. Dají se rozdělit do dvou základních skupin a to „Biometrické“ a „Nebio-metrické“. V této kapitole si nastíníme dnešní nejpoužívanější možnosti.

##### 3.1.1 Biometrické metody identifikace

Tyto metody jsou ve všeobecnosti založeny na identifikaci člověka na základě jeho fyziologických charakteristik, které se v průběhu života přirozeně nemění. Tyto metody se různí zejména mírou spolehlivosti a ceny. Jejich nejsilnější výhodou je pravděpodobně fakt, že pro jejich použití není třeba žádných přenosných médií, nebo nutnosti zapamatování číselného, či jiného bezpečnostního hesla, protože si identifikační médium nosíme vždy při sobě, a je jím naše tělo. Tyto metody jsou ale z technického hlediska zpravidla velmi složité a to se odráží především na jejich ceně. Nicméně si některé základní uvedeme. [4]

##### Verifikace hlasu

Tento způsob je v principu velmi jednoduchý. Testovaná osoba nahraje jako vzor pro kontrolu určitou zvolenou frázi, která je elektronicky zpracována metodou tzv. krátkodobé analýzy. Tato předloha se pak srovnává s identifikací uživatele a v případě shody je uživatel verifikován. [4]

### **Rozpoznávání obličeje**

Každý člověk má odlišnou stavbu tváře. Díky tomu zle jednoznačně vzájemně odlišit jednu osobu od druhé. Zde se opět předem nahraje předloha tváře identifikované osoby a s touto předlohou je elektronicky srovnáván snímek při identifikaci. Obraz je samozřejmě elektronicky zpracován a jsou vyhodnoceny klíčové znaky tváře, jako výška čela, vzdálenost očí, nebo velikost úst a nosu. [4]

### **Otisky prstů**

Tento způsob využívají i moderní kriminalisté a je známa už od roku 1877, kdy byla poprvé tato metoda ve svém nejjednodušším podání a prakticky použita. Metoda se jinak nazývá daktyloskopická a využívá charakteristik papilárních linií na prstech a dlaních člověka, kdy jsou tyto charakteristiky pro každého člověka jedinečné. [4]

### **Další metody**

Jako biometrické metody identifikace jsou používány ještě tyto metody: Síticový otisk, vzor žil na hřbetu ruky, geometrie ruky, geometrie prstu. [4]

#### **3.1.2 Nebiometrické metody identifikace**

Nebiometrické metody, jsou metody, které využívají výlučně technických prostředků, a identifikace osoby probíhá pomocí dalšího přenosného média, jako je karta, nebo přenosný chip. Toto zařízení se označuje také, jako *Token* Můžeme sem započítat i mobilní telefon, nebo online formulář pro přihlášení. Tyto metody si rozdělíme na dvě podkategorie, a budou to metody s přenosným médiem a bez přenosného média. Mezi metody s přenosným médiem patří:

#### **Karta s magnetickým pruhem**

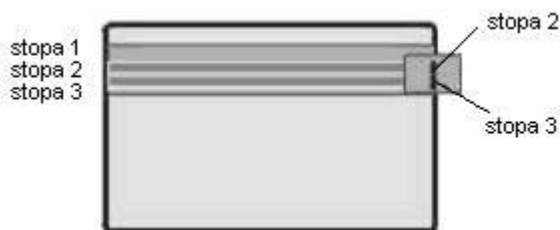
Jedna s nejznámějších metod identifikace pomocí přenosného média a v minulosti pravděpodobně nejrozšířenější. Dnes však již mnohokrát překonána a například v bankovní sféře

již její bezpečnost zdaleka nedosahuje potřebných standardů. Identifikace probíhala pomocí magnetického proužku na delším okraji karty, ze kterého byla načtena protažením čtečkou informace o identifikaci. Byla časem nahrazena kartami čipovými, už jen proto, že magnetický proužek uchoval, ve svých třech stopách maximálně pouhých 206 Bajtů. [5]

### Čipové karty a RFID karty

Jak již bylo psáno, jsou to nástupci karet s magnetickým proužkem. Největší oblibě se těší v bankovní sféře, ale i v budovách jako přístupové klíče, nebo platební médium v jídelnách, u turniketu lyžařských středisek a dalších aplikacích.

Původně se tyto karty začali osazovat kontaktními čipy paměťovými, které byli k vidění kartách do telefonních automatů a dodnes je má téměř každý člověk ve formě SIM karty v mobilním telefonu.



Obrázek 1 - Karta s magnetickým proužkem

Později se začaly osazovat čipy RFID. Jejich výhodou je nejen bezpečnost, ale také bezkontaktní využití. Princip načtení karty: čtecí jednotka vysílá pulsy, obvykle do vzdálenosti asi 30cm, pomocí kterého si karta při jeho zachycení krátkodobě nabije svůj vnitřní kondenzátor a tuto energii využije na odpověď zpět čtecímu zařízení.

Odě karty, jak už paměťové, nebo RFID jsou označovány jako bezpečné, protože mohou obsahovat kryptografické funkce. Často je využívána jejich kombinace a pojištěna tzv. Pin kódem, který musí klient zadat po jejím načtení pro maximální bezpečnost. [5]

### Jména, Hesla a PINy

Jedná se o nejjednodušší způsob autorizace. Právě proto jsou tyto metody používány ve velkém množství aplikací. Jméno je použito jako jednoznačný identifikátor klienta, zde může být využita přezdívka, email nebo unikátní identifikační číslo. Tímto se klient identifikuje a pro potvrzení, že jsou vlastníkem této identifikace je pro autorizaci nutné doplnit heslo, nebo PIN kód, který je znám pouze majiteli konkrétní identifikace. Heslo je známo pouze majiteli účtu a mělo by být v ideálním případě minimálně v délce 6-8 znaků. Řetězec by měl obsahovat kombinace malých a velkých písmen spolu s číslicemi. PIN kód na rozdíl od Hesla, je pouze v číselné formě o délce 4-6 číslic. [5]

### QR kód

QR z anglického „*Quick Response*“, neboli kód rychlé reakce. Tuto metodu budeme využívat v našem projektu, takže ji více rozebereme. Použití této metody bylo zvoleno pro svou jednoduchost, přístupnost a levné využití. QR kód je vlastně obrázek souhrnu býlích a černých políček uspořádané do matice ve čtyřiceti různých velikostech. Od nejmenších velikostí (21x21 bodů) až po největší (177x177 bodů). QR kód dokáže ve své největší velikosti pojmout až 4300 znaků textové informace. To je poměrně hodně na rozdíl od svých předchůdců čárových kódů, kdy například čárový kód EAN 14 pojme pouze maximálně 14 číselných znaků a pro jeho spolehlivé přečtení je nutná speciální laserová čtečka čárových kódů. Tím se dostáváme k další přednosti QR kódů, kterým je jeho čitelnost libovolným zařízením s kamerou a obslužnou aplikací. Tímto zařízením je myšlen především mobilní telefon, který, je vybaven fotoaparátem o rozlišení alespoň 2Mpixels, který je dnes běžnou součástí výbavy každého, zejména mladého člověka. Jeho další výhodou je, že k jeho generování a vyrobění tzv. *tokenu*, nepotřebujeme nic víc, než některou z volně dostupných aplikací a osobní tiskárnu, kterou si QR kód vytiskneme a připravíme k přečtení. Není potřeba žádného speciálního zařízení, či speciálního papíru.

Nevýhodou pak je, že při poškození přenosného média, našeho *tokenu*, poškrábáním, po-  
mačkáním, či namočením se QR kód na něm obsažený může natolik znehodnotit, že se stává nečitelným, zejména pak jeho velké verze. Jako nevýhoda se jeví také fakt, že pokud nemáme po ruce zařízení s rozlišením alespoň 2Mpixels, stává se přečtení datově objemnějších QR kódů dosti problémové. [6]



Obrázek 2 – Náhled QR kódu se zakódovaným textovým řetězcem



## 4 DATABÁZE

Databázová část má za úkol shromažďování a uchovávání dat, které budou v její struktuře uloženy a na požádání k dispozici.

### 4.1 Databáze a způsob uložení dat

Za základ celého systému můžeme označit Databázi, ve které jsou uložena veškerá data, která jsou zapotřebí pro chod aplikace. Dále data, která se dynamicky plní při chodu aplikace. Tyto data jsou dále vyhodnocována a vypisována uživatelům pomocí nadřazené aplikační vrstvy, nebo chceme-li „Databázového klienta“. V našem případě byla použita jedna z nejrozšířenějších databází, databáze MySQL.

### 4.2 MySQL

MySQL je relační databázový systém. Dle spousty názorů jde o rychlejší, spolehlivější a levnější variantu ke konkurenčním produktům jako Oracle, nebo DB2. MySQL je multiplatformní databázový systém, který dnes spolupracuje s většinou používaných programovacích jazyků. Nejčastěji PHP, nebo stále populárnější JAVA. MySQL byl původně vyvinut švédskou společností MySQL AB, konkrétně dvojicí vývojářů Michael Windenis a David Axmark. Velkou výhodou MySQL je možnost využívání širokou veřejností pod volnou licencí GPL. [7]

### 4.3 Návrh databáze

Abychom mohli vytvořit správnou databázi, je třeba dodržovat několik pravidel a zásad.

#### 4.3.1 Zásady pro návrh databází

Jako první zásadu, můžeme označit nedělitelnost záznamů. Což znamená, že každý jednotlivý záznam v databázi, by měl být samostatně nedělitelný. V praxi to tedy znamená, že pokud budeme chtít uložit například jméno a příjmení uživatele, je třeba tyto údaje uvést každý zvlášť. Tudíž budeme mít dva sloupce, jeden se jménem a druhý s příjmením.[8]

### 4.3.2 Ukládání dat pod správnými datovými typy

Abychom mohli data správně ukládat, je třeba si je rozdělit podle správných datových typů. Můžeme je tedy ukládat například jako číslo (Integer, Float, atd.), dále je lze uchovat ve formě znakových řetězců (Valchar, Text), avšak tyto data pak nejsou přirozeně systémem interpretována jako čísla a nelze s nimi provádět početní operace. Dalším úskalím pro zvolení správného datového typu, jsou předpokládané maximální potřeby daného záznamu. Jako příklad můžeme uvést předpokládané číslo s plovoucí desetinou čárkou. V tomto případě je nutné zvolit takový datový typ, jež dokáže toto číslo interpretovat správným způsobem (Float). Avšak každý takovýto datový typ si pro uložení záznamu rezervuje jiný prostor v úložišti. Tento prostor je vždy maximální, v případě, že si tedy nastavíme uložení řetězce maximálně pro 600 znaků, potom bude tento záznam zabírat v paměti úložiště místo pro 600 znaků, i když jich bude použito méně. Je tedy nutné dobře zvážit vhodný datový typ, aby nezabíral příliš mnoho místa v paměti úložiště. Neboť v případě velkého množství pro ukládání se databáze zpomaluje. Vhodným navolením tedy můžeme ulehčit a zrychlit celkovou práci při vyhledávání v databázi. V případě, že bychom zvolili špatný datový typ, který by byl příliš velký, tento nedostatek by se projevil, až při zaplnění databáze větším množstvím dat. [8]

### 4.3.3 Každý řádek by měl obsahovat jednoznačný identifikátor

Touto zásadou je myšleno, že každý řádek by měl obsahovat takzvaný identifikátor nebo jak lze říci pouze zkratkou, jedná se o ID. Zde napomáhá funkce SQL „auto inkrement“, jež přiděluje chronologicky pořadové číslo záznamům, již při jejich vytváření.[8]

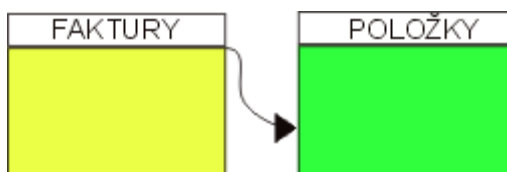
### 4.3.4 Nežádoucí duplicita dat

Pro zajištění správné struktury databáze je třeba, aby každý záznam, který je do ní vložen, zde byl pouze jednou. Pokud si tedy uložíme jméno uživatele do tabulky „Uživatel“ a pak i do tabulky „Kupující“, je to chyba. V prvních okamžicích se může zdát, že se jedná o ulehčení práce programátora, avšak tahle domněnka je velice mylná. Za prvé, je tento postup velice neefektivní co se datové úspory týče, dokonce následně vzniká chaos při psaní dotazů na databázi a při tvorbě struktury obslužné aplikace. Výjimku tvoří tzv. odkazy nebo

primární klíče, kterými bývají převážně právě ony číselné unikátní identifikátory jednotlivých záznamů, o nichž jsme si řekli již výše. [8]

#### 4.3.5 Relace

Databáze MySQL má přívlastek „*Relační*“, a proto bychom měli využívat tuto její vlastnost v maximální možné míře. Relace více tabulek se dají specifikovat v několika základních poměrech. 1:N je relace, kdy pro jeden záznam řádku referenční tabulky existuje neurčitý počet záznamů v tabulce podřízené.



Obrázek 3 - Relace 1:N [9]

Vztah N:N je v databázích méně častý, jedná se o tzv. asociativní tabulky, kdy jedna tabulka s relací N:N obsahuje cizí klíče dvou nadřazených tabulek. Jako příklad z reálného života můžeme uvést například situaci, kdy máme tabulku „*Knihy*“ a tabulku „*Čtenáři*“ a bylo by nutné přiřadit jednotlivé knihy čtenářům dle výpůjček, přičemž by nebylo předem jasné, kolik knih si čtenář vypůjčí, stejně tak jako není jasné, kolik čtenářů si půjčí každou z knih.



Obrázek 4 – Relace N:N [9]

Tyto *Relace* se skládají ze sady tzv. primárních klíčů, které obsahuje každá referenční tabulka, a „*Cizích klíčů*“, které obsahují tabulky podřízené. Údaj v cizím klíči pak obsahuje odkaz na řádek primárního klíče referenční tabulky a spojuje tak záznamy z obou tabulek. Relace nemají ale za úkol pouze odkazovat a spojovat záznamy v různých tabulkách, ale

také konkretizují vztahy těchto záznamů a stanovují jejich vzájemné chování a vztahy při úpravě nebo smazání souvisejících dat. Pomocí atributů jednotlivých relací (CASCADE, RESTRICT, NO ACTION, SET NULL) se v databázích typu InnoDB stanovuje toto chování, a tím se zachovává integrita dat databáze. [9]

#### **4.3.6 Pevně daná struktura databáze**

Každá správně vytvořená databáze by měla být navrhnutá tak, aby obslužná aplikace nemusela vytvářet žádné další tabulky, ale pouze záznamy v tabulkách. Pokud bychom tuto zásadu nedodrželi, celá struktura se stane velice nepřehlednou a obzvlášť datově zbytečně objemnou. [7]

#### **4.3.7 Kompaktní struktura**

V případě, že programátor, jež navrhoval databázi dodrží výše zmíněné zásady a použije své logické myšlení, pak vytvořená databáze bude fungovat rychle a efektivně. Zatížíme-li později databázi větším datovým objemem, nedojde k razantnímu zpomalení při zpracování dat v této databázi. Velice důležitým ukazatelem, je že je databáze navržena správně je potom úprava struktury při nutnosti přidat další položku do této struktury. V případě, že je navržena správně, pro tento úkon je zapotřebí pouze minimálních úprav k dosažení požadovaného cíle a celý tento proces nijak nenaruší další funkce této databáze a jejích obslužných aplikací. V opačném případě, kdy tomu tak není, je v návrhu struktury zásadní chyba a tudíž se může stát, že zdánlivě jednoduchý úkol se stává skoro téměř neřešitelným.[7]

### **4.4 Datové typy**

V této části naší práce se budeme zabývat datovými typy. Každá databáze MySQL vyžaduje již předdefinované datové typy pro každý záznam v ní uvedený. Řekneme si tedy, jaké datové typy byly v naší databázi použity. [7]

## Integer

První datovým typem bude Integer. Jedná se o celočíselný datový typ, který můžeme označit zkratkou INT(x). V zápisu zkratky písmeno „x“ pak definuje počet míst, jež může políčko ukládat. Zápis tedy potom může například vypadat takto INT(3). Takhle zapsaný Integer nám říká, že do tohoto pole lze ukládat maximální hodnotu 999. Celkový maximální rozsah hodnot, v nichž se může záznam nacházet je - 2147483648 až 2147483647. [10]

## Float

Float, neboli v překladu plovák, nám označuje datový typ, jež obsahuje plovoucí desetinnou čárku. Tento typ využijeme v případě, je-li třeba zapsat jiné hodnoty než pouze celočíselné. Nejmenší nenulové hodnoty jsou pak tedy  $\pm 1,175494351E-38$ ; největší nenulové hodnoty jsou  $\pm 3,402823466E+38$ . [10]

## Varchar (M)

Dalším datovým typem je Varchar (M), který používáme pro řetězec pevně dané délky dlouhý 0 až M. M musí být celé číslo v rozmezí od 0 do 255. Budeme-li chtít uložit delší řetězec do databáze než je M, tak se zkrátí na délku M a co je nad tento limit se tzv. „vytrátí“. Avšak jedná-li se o případ opačný, neboli ukládaný řetězec je kratší než M, pak jsou koncové mezery při ukládání odstraněny. [10]

## Text

Tímto datovým typem je řetězová hodnota, jež má neomezený počet znaků. [10]

## Date a Datetime

Tyto dva datové typy jsou si nadmíru podobné. Jedná se o časový datový typ, přičemž Date při svém ukládání ukládá pouze hodnotu data, avšak naproti tomu Datetime tuto hodnotu rozšiřuje o přesný čas. [10]

## 5 WEBOVÁ APLIKACE

Aplikační část má za úkol komunikovat s uživateli, vypisovat stav systému a pracovat přitom s daty, ke kterým přistupuje prostřednictvím databáze. Tyto data a další informace vypisuje uživateli na obrazovku a nastavuje mantinely práce s aplikací.

### 5.1 Architektura MCV

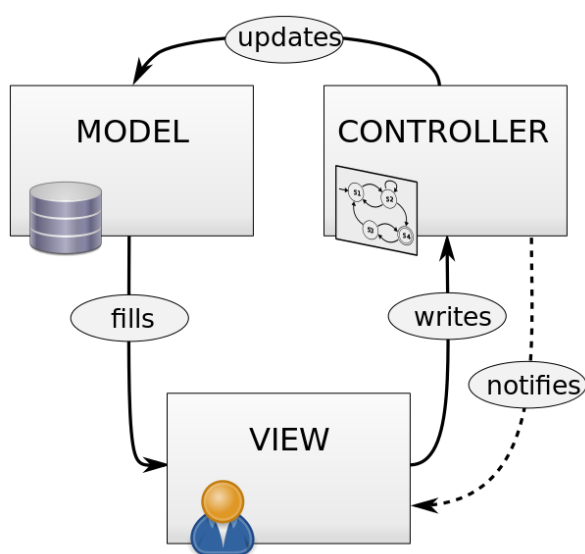
MCV z anglického slova Model-view-controller. Jedná se o nejmodernější a dnes nejpoužívanější architekturu, ve které se tvoří nejen webové aplikace.

#### 5.1.1 Princip

**Model** – Model reprezentuje informace, například data z databáze, ke kterým aplikace jeho prostřednictvím přímo přistupuje.

**View** – View (pohled) přebírá data od modelu a transformuje je do podoby, ve které jsou srozumitelné pro uživatele.

**Controller** – Controller (řadič) čeká na akce uživatele a na jejich základě provádí změny ve vrstvách View a Model.



Obrázek 5 – Schéma práce architektury MCV [2]

## 5.2 PHP

Z anglického Hypertext Preprocessor v překladu Hypertextový preprocesor. Jedná se o scriptovací jazyk. Jeho počátky se datují kolem roku 1995, kdy na jeho vývoji pracoval Rasmus Lerdorf, ještě pod jménem PHP/FI. Základ syntaxe jazyka je v jazyce Perl. V roce 1997 se rozhodli vývojáři Zeev a Andi spojit s Rasmusem a jazyk zcela přepracovat. Zde vznikla první plnohodnotná verze PHP 3. Dnes je jazyk PHP již ve verzi PHP 5, konkrétně 5.5.12 vydán 1. 6. 2014. Jedná se o multiplatformní programovací jazyk, který podporuje i moderní objektový model programování. Implementuje komunikaci s databázovou základnou MySQL, PostgreSQL, MSSQL, Oracle. Podporuje celou řadu webových protokolů jako HTTP, FTP, IMAP, POP3 a další. Už proto, je nejvyužívanějším jazykem pro vývoj webových aplikací a dalece převyšuje svoje přímé konkurenty ASP, .NET. [11]

## 5.3 Nette

Pro tvorbu webových aplikací je dnes k dispozici spousta tzv. frameworků. Framework je soubor knihoven, které šetří vývojářům a programátorům spoustu práce, protože obsahují soubory rutin, které jsou nejvíce využívány při vývoji webových aplikací. Tyto rutiny by si musel každý programátor opakovaně tvořit pro každý další projekt znovu, testovat, debugovat a zde jsou přitom k dispozici ihned a v otestované formě.



Obrázek 6 – Logo nette Framework [13]

Nette, jedná se o opensource Framework pro jazyk PHP od verze 5.0. Jednou z hlavních výhod tohoto frameworku je implementace moderní architektury MCV, která logicky odděluje získávání a zpracovávání dat aplikace od jejich prezentace uživateli a tím v první řadě zpřehledňuje psaný kód. Dále podporuje znovu-použitelnost kódu pomocí tzv. komponent, které jsou opakovaně použitelné v různých částech aplikace jako celek, ne jako

bylo z dřívějších dob programování v jazyce PHP zvykem, opakovaně upravovat podobné části aplikace z ohledem na aktuální umístění.

Původním autorem frameworku Nette je David Grudl, nicméně dnes je vývoj frameworku pod taktovkou Nette Foundation.

Nette je šířeno pod licencemi GNU, GPL a pod vlastní licencí Nette. Aktuální a v našem projektu použitá verze Nette 2.1.2 stable, vydaná 11. 2. 2014. [14]

### 5.3.1 Latte

Latte je šablonovací jazyk, který pohodlně propojuje jazyky HTML a PHP, spojuje všechny jejich přednosti do jednoho souboru. Je původně vytvořen jako součást frameworku Nette. Je ale použitelný i jako samostatná komponenta webové aplikace. Syntaxe vychází z praktických požadavků programátorské komunity. Pomocí helperů a maker, které rozšiřují použitelnost PHP a HTML, je vykreslení stránky velmi přehledné a jednoduché. Dále zabráňuje útokům XSS, a dalších bezpečnostních děr webových aplikací. [15]

### 5.3.2 Laděnka

Částí frameworku Nette je také užitečný nástroj jménem Laděnka. Tento nástroj má pro vývojáře obrovskou výhodou, kdy mu v reálném čase poskytuje detailní informace o běhu aplikace. Konkrétně informuje o propojení a komunikaci s databází, komunikace v obou směrech SERVER -> Klient, i Klient SERVER, informace o hostovském serveru, objem přenesených dat, času zpracování a vykreslení aplikace a v neposlední řadě detailní informace o vzniklých chybách. Celý tento nástroj pomáhá vývojáři detailně debugovat a ladit aplikace. [13]

### 5.3.3 Firebug

Laděnka komunikuje s některými prohlížeči a to pomocí instalovatelných rozšíření prohlížečů, které je podporují. Laděnka tak posílá informace o běhu aplikace přímo do konzoly prohlížeče. Takovým rozšířením je právě Firebug, který je k dispozici pro webové prohlížeče Firefox a Chrome. Při použití Firebugu je debugování aplikace ještě o něco efektivnější.



#### 5.3.4 Logování chyb

Chyby z laděnky je možné nejen vypisovat za chodu aplikace na obrazovku, nebo přes rozšíření prohlížeče Firebug, ale také pomocí tzv. Logů ukládat na určené místo, a to buďto přímo na hostovský server, nebo na vzdálené úložiště. Toto logování je užitečné především pokud je aplikace v produkčním režimu, neboli v ostrém provozu, kdy bychom neradi odhalili uživatelům vnitřní strukturu aplikace a tak je Laděnka skrytá a vypisuje případné chyby pouze skrytě ve formě těchto Logů.

#### 5.3.5 Zabezpečení aplikace

Nette má už v základní instalaci zabezpečení proti nejznámějším útokům pro narušení webových aplikací, jako Cross-Site-Scripting, nebo SQL Injection.

### 5.4 JavaScript

Jedná se o scriptovací multiplatformní jazyk, hojně používaný v oblasti webových stránek. Vkládá se přímo do HTML kódu stránky a obstarává převážně dynamické interaktivní prvky stránek zpracovávaných na straně klienta, respektive webového prohlížeče. Jeho syntaxe vychází z jazyků C/C# a Java. Konkrétně ovládá různá políčka formulářů, tlačítka a obstarává animace prvků. Vyvinut byl společností Sun Microsystem v prosinci 1995, jako doplněk jazyků HTML a Java. Za otce JavaScriptu je pak považován Brendan Eich. JavaScript v různých verzích spolupracuje kromě jazyka HTML i s jazyky PHP, ASP a .NET. Jejím rozšířením je pak Framework jQuery, který rozšiřuje možnosti JavaScriptu do neuvěřitelných rozměrů. [16]

#### 5.4.1 jQuery

V současnosti se snad programovací jazyk JavaScript ve webových aplikacích nepoužívá bez své rozšiřující knihovny jQuery, která svými funkcemi a jednoduchou implementací vdechne stránkám život. Dnes je její vývoj již tak daleko a je tak oblíbená, že podporuje většinu používaných webových prohlížečů. Její počátek je datován do roku 2006 na newyorském BarCampu, kde byla představena její první verze, vytvořena programátorem Johnem Resigem. Dnes je jQuery využíváno nejen v oblasti webu, ale také v mobilních

aplikacích. JQuery je vydáno pod licencí MIT. Aktuální verze v době psaní tohoto textu je JQuery 2.0.3 vydání 3. 7. 2013. [17]

## 6 BEZPEČNOST WEBOVÝCH APLIKACÍ

Zprávy o různých úspěšných či méně úspěšných webových útoků se k nám dostanou čím dál častěji. Nejvíce úspěšné jsou útoky na světoznámé internetové giganty. Jedním z posledních útoků, který byl na tento světoznámý gigant, byl útok na firmu Adobe, kdy došlo k úniku miliónů uživatelských dat. Dalšími známými útoky různých hackerských skupin jsou útoky na webové stránky vládních organizací nebo politických stran. O bližších útocích na webové aplikace si řekneme níže.

### 6.1 Cross-Site-Scripting

Tento typ útoku bývá vývojáři webových aplikací nadmíru podceňovaný, a přitom jde o poměrně nebezpečný typ útoku, který je sice často prezentovaný jako nástroj na rozbití layoutu webové aplikace, avšak přitom dokáže ukrást z prohlížeče návštěvníka, který tuto napadnutou webovou aplikací zobrazí, citlivé data o přihlašovacích údajích a odešle je útočníkovi, takže je potom útočník schopný přistoupit k dané webové aplikaci jako přihlášený uživatel, kterému byli data ukradeny data z prohlížeče.

Účinnou obrannou proti XSS útokem spočívá nejen ve validaci, ale i v escapování formulářových vstupů a dat, které se ukládají do databáze, ale hlavně v escapování vypisovaných dat v HTML šablonách.[17]

### 6.2 Cross-Site Request Forgery (CSRF)

Tento útok je koncipovaný na podvrhování požadavků mezi různými stránkami. K tomuto útoku je využíván samotný uživatel webové aplikace nejvíce tak, že o tom vůbec neví. Stačí, že jako vždy používá webovou aplikaci a nevědomky tak na pozadí vykonává škodlivé požadavky, které mohou mít za následek manipulaci s daty, jejich smazání či odeslání na neautorizované místa. Při tomto útoku dochází ke zneužití http požadavků GET a nebo POST.

Velice častým případem zneužití je například vložení škodlivé URL adresy do HTML tagu <img> a tento tag vložit například někam na velmi navštěvovaný portál, nebo diskusní fórum.

Před CSRF je nejlepší obranou vkládání tokenu jako hodnoty skrytého pole ve formuláři. Tento token je náhodně vygenerovaný řetězec, který je uložený v session a vložený do

formuláře. Jakmile je formulář odeslán, vykoná se kontrola, zda token odeslaný ze skrytého pole formuláře souhlasí s tím, který je uložen v session. Podle tohoto je možné identifikovat, zda je odeslaný formulář opravdu odeslaný z naší webové aplikace nebo je podstrčený třetí stranou. [19]

### 6.3 SQL Injection

SQL Injection je jeden z nejčastějších útoků na webové aplikace, který je jakýmsi nosným typem na webové aplikace. Naproti tomu, je obrana proti tomuto útoku velmi jednoduchá a s tímto typem útoků se setkáváme velmi často. Je totiž velmi nebezpečný, neboť útočník je schopný během několika sekund získat plnou kontrolu nad databází webové aplikace.

Celý tento útok spočívá ve zneužití neošetřených formulářových vstupů ve webové aplikaci. Útočník vloží do formulářového vstupu, který pracuje přímo s databází webové aplikace určitý řetězec, kterým se snaží upravit SQL dotaz na databázi. Získá tak kontrolu nad samotnou databází. Útočník je takto schopen dostat se jak k samotným datům, tak i s nimi pracovat, měnit je, ba dokonce může celou databázi smazat.

Nejlepší obranou před SQL Injection je takzvané escapování hodnot, které jsou do aplikace zasílány pomocí formulářových vstupů. Escapování znamená, že řetězec, který je odeslaný přes formulář webové aplikace projde funkcí, která ošetří všechny nebezpečné znaky, které řetězec může obsahovat a které by mohli svou přítomností v řetězci pozměnit dotaz na databázi. [20]

## **II. PRAKTICKÁ ČÁST**

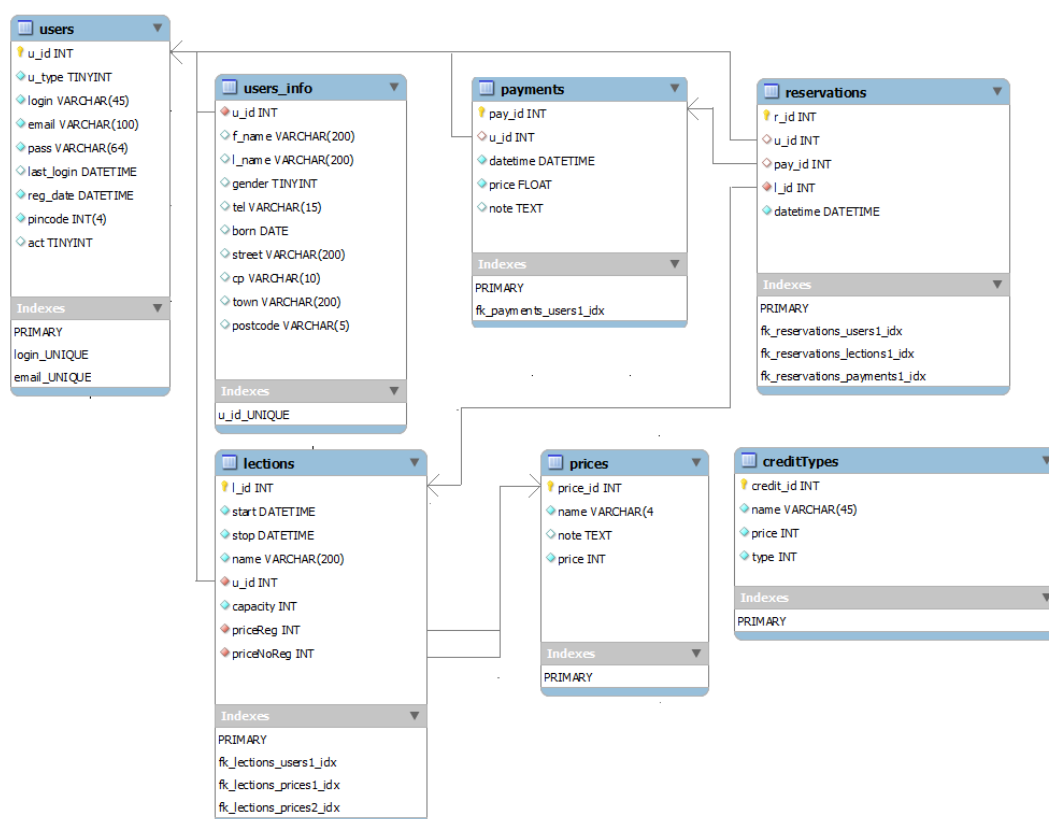
## 7 DATABÁZOVÁ ČÁST

Databázová část, která ukládá všechny data aplikace, byla navržena dle zásad pro tvorbu databází. Při návrhu byl použit software pro tvorbu a vizualizace databází WorkBench.

### 7.1 Struktura databáze

Samotná struktura databáze je tvořena sedmi tabulkami, které mají vzájemné relace. Formát úložiště je typu InnoDB, která nám dovoluje použít relační model a celá databáze je v testovací verzi provozována na komerčním profesionálním hostingu společnosti WEDOS Internet a.s.

Formát kódování, tzv. charset, byl zvolen na základě doporučení v dokumentaci frameworku Nette, který komunikuje s databází. Zvolené kódování je ve formátu UTF-8. [21]



Obrázek 7 – Struktura databáze a propojení tabulek

Databáze rezervačního systému tvoří 7 tabulek, které mají mezi sebou nastavené relační vztahy. Vztahy jsou jednoduše znázorněny ve schématu databáze.

Tabulka *users* obsahuje základní údaje o uživateli, které jsou využívány při běhu aplikace. Na tuto tabulku jsou napojeny téměř všechny ostatní, protože údaje o uživateli participuje ve všech funkcích aplikace.

Tabulka *users\_info* obsahuje dodatečné informace o uživateli, se kterými se nepracuje tak často, jako je bydliště, telefon atd.

Tabulka *lections* pracuje s daty lekcí. Z těchto dat se vypisuje kalendářní zobrazení lekcí.

Tabulka *reservations* je tabulka, která uchovává informace o rezervacích na jednotlivé lekce. Relace pak jsou nutné na tabulku lekcí, uživatelů a plateb.

Tabulka *payments* je důležitá, protože ukládá data o všech záúčtovaných platbách v aplikaci. Z dat této tabulky se počítají veškeré obraty ve všech částech systému, včetně zůstatků jednotlivých uživatelů.

V tabulce *prices* si systém zjišťuje ceny, které jsou přiřazeny jednotlivým lekcím. A pak na jejich základě záúčtuje do tabulky *payments*.

Poslední méně významná tabulka, *creditTypes* ukládá statické možnosti dobíjení kreditu na účty klientů. Tato tabulka jako jediná nemá nastaveny žádné relační vztahy.

## 8 PRAKTICKÁ IMPLEMENTACE

Aplikace byla navržena dle požadavků provozovatele spinning centra Mad Jack ve Zlíně. Jak už bylo předesláno, jedná se o online aplikaci, která využívá výhod přístupnosti internetu. Je postavena na platformě PHP 5 a převážně za pomoci frameworku Nette. Data pak uchovává pomocí databázového systému MySQL. Pro běh testovací verze a samotný vývoj, byla použita placená verze hostingu u společnosti WEDOS.com. Aplikace byla nazvána *SpinningRobot*.

### 8.1 Modul Admin

Do modulu admin má výhradní přístup pouze personál spinning centra. Z tohoto modulu je ovládána celá aplikace a její funkce. Uživatel, který se do administrace přihlásí, bude mít přehled nad děním ve spinning centru, konaných lekcích, registracích a platbách, které se v určitém časovém rozmezí uskutečňovaly. Je možné vytvářet a editovat a rušit lekce, je možné zadávat tzv. pokladní platby za jednotlivé rezervace a dobíjení kreditu na pokladně. Dále je možné zadávat částky pro dobítí a ceny rezervací, jejich editace. Také jsou k dispozici možnosti přidávání a úpravy uživatelů a vyhledávání v uživateli.

#### 8.1.1 Přihlášení

Aby byl uživatel připuštěn do aplikace, je nutné, aby jej některý z administrátorů označil jako administrátora. Obyčejný řádový uživatel, který byl vytvořen volnou registrací, nemá pro vstup do této sekce oprávnění.

#### 8.1.2 Přehled

Po přihlášení a vstupu do administrátorské sekce se nám nabídne krátký přehled, který obsahuje poslední přihlášení do aplikace. Dále zde najdeme seznam posledních plateb a rezervací. Jako poslední údaj je časová osa aktuálního dne. Odtud dále pokračujeme do dalších sekcí aplikace.



Dnes:

0 Přihlášených klientů, 3 plateb, 3 rezervaci

**Poslední platby**

id	uživatel	Platba za	Datum	Cena
76	neznámý	studentská lekce - spinning student	00:20 11. 06. 2014	75
75	neznámý	spinning začátečníci - spinning pokročilí - spalování	00:18 11. 06. 2014	90
74	neznámý	spinning začátečníci - spinning pokročilí - spalování	00:08 11. 06. 2014	90
72	hajkova	dance - spinning začátečníci	21:47 10. 06. 2014	150
73	hajkova	Stržení dance - spinning začátečníci	21:47 10. 06. 2014	-150
71	hobkova	vkład pokladna	20:54 10. 06. 2014	450

**Poslední rezervace**

id	uživatel	Na lekci	Datum
108	neznámý	spinning student	00:20 11. 06. 2014
107	neznámý	spining pokročilí - spalování	00:18 11. 06. 2014
106	neznámý	spining pokročilí - spalování	00:08 11. 06. 2014
105	kreed	spinning ranní	22:43 10. 06. 2014
104	hajkova	spinning začátečníci	21:47 10. 06. 2014
103	voprs	spinning začátečníci	21:41 10. 06. 2014

[středa 11. června 2014](#)

08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
08:15-09:00 spinning pokročilí - spalování 5/10 90 Kč													

Obrázek 8 – Náhled obrazovky při přihlášení do aplikace

### 8.1.3 Uživatelé

Položka „*Uživatelé*“ má za úkol pracovat a přistupovat k datům klientských účtů a uživatelských profilů.

#### Výchozí zobrazení

Ve výchozí obrazovce sekce „*Uživatelé*“ najdeme náhled posledních tří registrovaných uživatelů seřazených v tabulce, kde jsou k dispozici základní údaje, jako uživatelské jméno a datum registrace, stejně jako nástroje pro vstup do sekce plateb klienta a standardní sadu nástrojů pro manipulace s klientským účtem. Sada nástrojů obsahuje vstup do náhledu uživatelského profilu, rychlý přístup do jeho editace, funkce aktivace a deaktivace uživatelského účtu a nakonec možnost jeho smazání.














Další tabulkou, která je zde zobrazena, jsou poslední přihlášení uživatelé do aplikace. Tato je pouze informativní a neobsahuje žádné nástroje k manipulaci se zobrazeným klientským účtem.

## Seznam uživatelů

Podsekce uživatelské části je „*Seznam uživatelů*“ dává možnosti vyhledávání uživatelů, filtrování a seřazování dle různých parametrů, jako jméno, příjmení, email. Po vyhledání se zobrazí opět tabulka s výsledkem vyhledávání seřazen do tabulky, včetně nástrojové lišty pro manipulaci s účtem uživatele.

### Přehled

#### Poslední registrování

id	login	Registrován	platby	nástroje
50	MasFuk	10. 06. 2014		   
48	hobkova	10. 06. 2014		   
47	blecha	10. 06. 2014		   

#### Poslední přihlášení

id	login	poslední přihlášení
1	kreed	23:42
28	hambman	16:29
29	malantikova	20:19

Obrázek 9 – Obrazovka přehledů uživatelů

## Přidání uživatele

Poslední podsekcí v kategorii Uživatelé je „*Přidání uživatele*“, která nás zavede do formuláře pro manuální přidání uživatele se všemi dostupnými položkami, viz. Obrázek 9. Při přidávání se testuje několik parametrů formuláře. Konkrétně se v polích přihlašovací jméno a email testuje srovnáním s údaji v databázi na duplicitu, kdy upozorní uživatele na shodu a vyzve k zadání jiného řetězce. Tyto údaje jsou povinné, stejně jako zadání hesla pro přihlašování vloženého uživatele. Po odeslání formuláře se odešle na zadanou adresu email s registračními údaji.

#### 8.1.4 Lekce

##### Výchozí zobrazení

Výchozí zobrazení pro sekci lekcí nám ukáže v první řadě kalendář, kde jsou zvýrazněny dny, kdy se konají lekce, po kliknutí na konkrétní den, nás aplikace přesměruje na zobrazení časové osy zvoleného dne.

Níže je pak zobrazen tabulkový výpis lekcí, konající se aktuální den. Tabulka nám zobrazuje základní informace o lekci, jako čas konání, název, lektor lekce, ceny lekce a obsazenost rezervací vůči její kapacitě. Dále zde najdeme nástrojovou lištu s možností zobrazení detailů lekce, editaci nebo její zrušení.

##### Seznam lekcí

V této sekci najdeme tabulkový výpis vložených lekcí i lekcí proběhlých. Možnost zvolení dne pro výpis lekcí a vyhledávání dle různých parametrů. V pravé části tabulky se zobrazením výsledku vyhledávání najdeme nástrojovou lištu pro manipulaci a zobrazení detailů konkrétní lekce.

## Seznam lekcí

## Nadcházející lekce

id	Datum	Název	Lektor	Cena rezervací / na místě	Obsazenost	Nástroje
39	19:00 - 19:45	spining pokročilí	hambman	90 / 110	1 / 7	✗ 🔍 📄
40	17:00 - 17:45	spinning začátečníci	kreed	90 / 100	3 / 5	✗ 🔍 📄
24	11:00 - 11:30	spinning začátečníci	hambman	150 / 60	7 / 7	✗ 🔍 📄
36	19:00 - 19:50	spining pokročilí - spalování	kreed	90 / 110	2 / 5	✗ 🔍 📄
45	08:15 - 09:00	spining pokročilí - spalování	drakula	90 / 100	5 / 10	✗ 🔍 📄

## Lekce proběhlé

id	Datum	Název	Lektor	Cena rezervací / na místě	Obsazenost	Nástroje
19	16. 05. 2014 17:20 - 18:15	mažoretky	neznámý	100 / 100	1 / 10	✗ 🔍 📄
13	25. 05. 2014 13:00 - 14:00	Lekce dalsi	neznámý	149 / 149	1 / 1	✗ 🔍 📄
8	25. 05. 2014 15:00 - 15:55	Test spinning	kreed	149 / 149	0 / 1	✗ 🔍 📄
21	29. 05. 2014 12:00 - 12:05	Lekce dalsi	neznámý	60 / 60	0 / 5	✗ 🔍 📄
37	04. 06. 2014 19:00 - 20:00	spinning student	malantikova	75 / 75	0 / 10	✗ 🔍 📄
25	06. 06. 2014 13:00 - 13:55	spinning	hambman	125 / 125	0 / 1	✗ 🔍 📄

Obrázek 10 – tabulkový seznam lekcí

## Přidání lekce

Pro přidání lekce je k dispozici formulář, který si vyžádá pro přidání všechny nutné informace. Jméno lekce, čas konání, lektor, kapacita, délka lekce a nakonec cena lekce. Při zvolení dne v příslušném políčku se dynamicky změní časová osa, aby měl operátor přehled o rozpisu na konkrétní den a vložil lekci tak, aby se časově nepřekrývaly. Pokud se přece jen stane, že z nepozornosti administrátor nastaví časový údaj, kterým by se lekce překrývala s lekcí již vloženou, je zavedena kontrola, která zabráni vložení a upozorní na chybu.

Ceny lekce jsou zde rozděleny do dvou variant. První je cena, kterou hradí uživatel při použití rezervační aplikace a uhradí ji tak z kreditu, který má předplacen. Tato cena by měla být z pravidla nižší, aby byla vyvinuta motivace klienta, využívat rezervační aplikaci a elektronické platby.

**Přidat lekci**

Jméno lekce:

Lektor:

Kapacita:

Začátek:   :

Délka lekce:

Cena lekce:

Cena registrovaných:

Cena na místě:

[vložit](#)

čtvrtek 12. června 2014

08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
								16:15-17:00 spinning začátečníci 2/10 73Kč		18:00-18:50 spinning začátečníci 2/7 20Kč	19:00-20:00 spinning začátečníci 3/10 73Kč		

Obrázek 11 – Přidávání lekce do databáze

## 8.1.5 Platby

### Výchozí zobrazení

Zde si přihlášený administrátor udělá obrázek o stavu nevyužitého kreditu všech uživatelů. Dále jsou zobrazeny platby a rezervace aktuální den.

### Přehled plateb

Zde se zobrazí množství příjmů plateb, které jsou zavedeny do databáze aplikace za určité období. Výchozím jsou čtyři základní časové vymezení, a to za aktuální den, za aktuální týden, za minulý týden, aktuální měsíc a minulý měsíc. Po vstupu do detailů, je možné si časové rozmezí libovolně upravit. V přehledu se pro lepší představu zobrazí i grafická reprezentace údajů plateb přijatých a plateb odečtených.

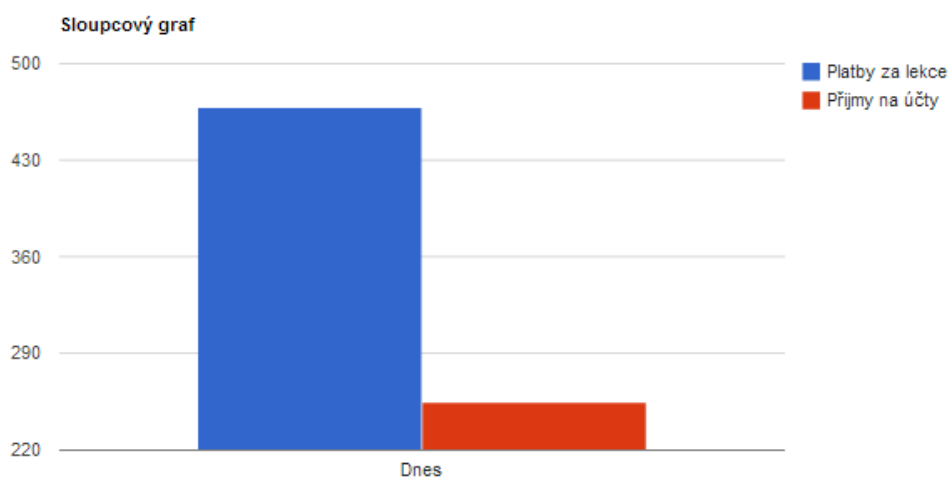
### Platby

id	Uživatel	Datum	Poznámka	Cena
74	-neznámý-	00:08	spinning začátečníci - spinning pokročilí - spalování	90
75	-neznámý-	00:18	spinning začátečníci - spinning pokročilí - spalování	90
76	-neznámý-	00:20	studentská lekce - spinning student	75

### Rezervace

id	Uživatel	Čas	Typ	cena
106	-neznámý-	00:08	spining pokročilí - spalování	90
107	-neznámý-	00:18	spining pokročilí - spalování	90
108	-neznámý-	00:20	spinning student	75

### Grafický přehled



Obrázek 12 – Přehledy plateb a rezervací

### Přidání platby

Přes tuto položku se dostaneme do obrazovky, kde je uživatel nejprve požádán o vyhledání uživatelského účtu, ke kterému chce platbu provést a po zobrazení výsledku vyhledávání dle uživatelského jména, zvolí, zda chce uživateli přidat nebo ubrat kredit. Dalším krokem je již formulář pro připsání nebo ubrání částky z účtu zvoleného uživatele. Zde je pak možnost využít přednastavených částek, nebo ruční zadání částky s poznámkou o provedené kreditové operaci.

### 8.1.6 Ceny

Ceny jsou sekce, kde si administrátor přednastaví ceny, které se pak zobrazí v možnostech při přidání, nebo editaci lekcí. Při rezervaci na lekci je pak klientovy stržena částka, která je ze zde přednastavených částek pro lekci zvolena.

#### Výchozí zobrazení

Částek je menší množství a tak není zapotřebí žádné dodatečné vyhledávání a filtrování vložených cen. Ve výchozím zobrazení se ukáže seznam všech dostupných cen v databázi seřazených do tabulky. Tabulka obsahuje všechny údaje o vložených cenách, a to její název, cenu a nástrojovou lištu. V nástrojích je možnost smazání a úpravy existujících cen.

Avšak pro smazání lekce existuje restrikce, kdy nelze smazat cenu, která je stále nastavena v některé z existujících lekcí. Při zjištění porušení této podmínky se cena odmítne smazat a aplikace upozorní administrátora chybovou hláškou.

#### Přidat cenu

Položka, která dle svého názvu reprezentuje přidání další položky do seznamu použitelných cen lekcí. Pomocí zobrazeného formuláře jednoduše přidáme cenu, přičemž povinné údaje jsou název ceny a její číselná hodnota.

### 8.1.7 Kredity

Sekce kredity mají za úkol přednastavit jednotlivé částky, které si může klient dobíjet na účet aplikace. Každá vložená částka se zobrazí v ceníku v sekci pro dobíjení kreditu v Uživatelském modulu. Zde jsou rozděleny do dvou variant, a to do plateb, které se dají dobíjet na pokladně a plateb, které lze uskutečnit bezhotovostně přes platební bránu PayPal.

#### Výchozí zobrazení

Po vstupu do sekce „Kredity, se zobrazí tabulka se všemi platbami v databázi, které jsou k dispozici. Tabulku lze filtrovat na platby manuální, neboli na pokladně, a platby zvolené pro metodu PayPal.

Samozřejmě je nástrojová lišta pro operace s kredity s možnostmi editace a smazání.

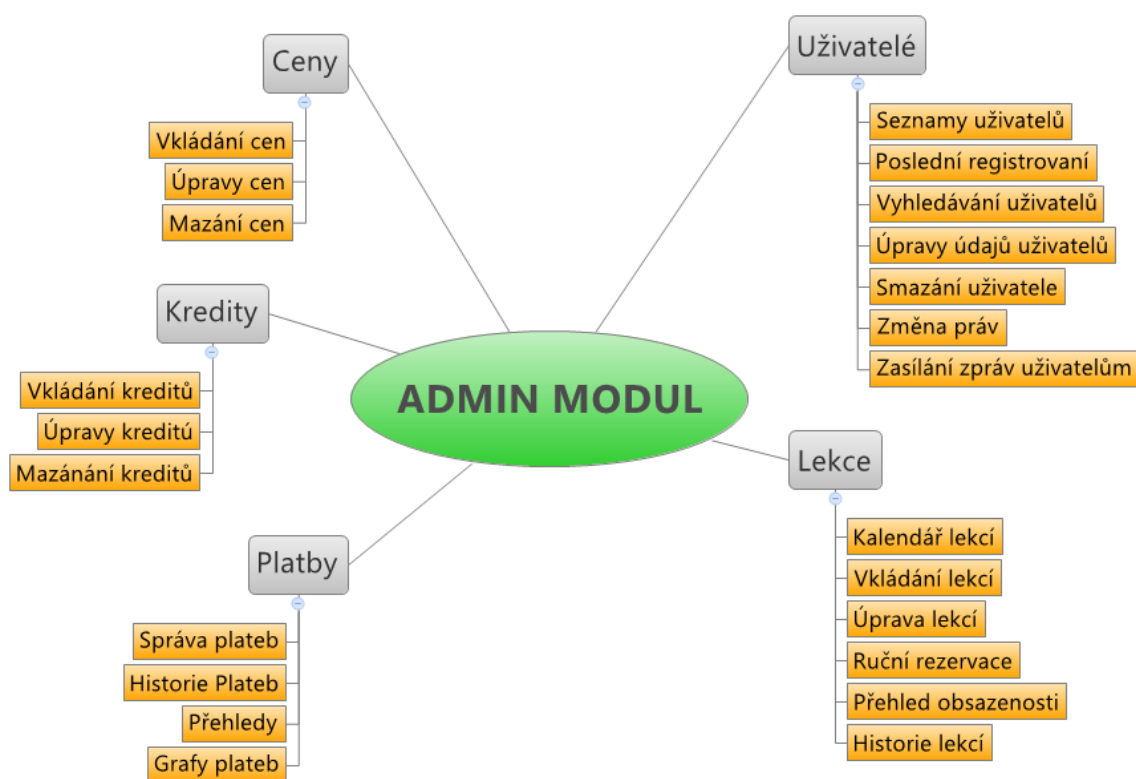
### **Přidání kreditu**

Zde je k mání formulář, kde je k dispozici pouze jméno ceny, její hodnota a typ. Typy jsou PayPal, nebo manuální platba na pokladně, ke kterým se bude částka v ceníku hlásit. Při volení typu platby se jako nápověda mění orientační tabulka, která zobrazuje již vložené platby pro lepší přehled.

#### **8.1.8 Ruční CRON**

Tato položka je přístupná pouze v testovací a presentační verzi aplikace, kde je možné spustit údržbové periodicky spouštěné skripty, takzvaný „CRON“ ručně a demonstrovat, nebo ověřit tak jejich funkci. K dispozici je funkce „*Překlopení rezervací do plateb*“ a „*Smazání lekcí a jejich rezervací starších než jeden měsíc*“.





Obrázek 13 – Diagram aplikace Administrátorské sekce

## 8.2 Modul User

Tento modul je dostupný uživatelům, kteří se by se chtěli zúčastnit lekcí spinning centra a mají tak zájem o registraci na lekce. Po zadání základních registračních údajů a potvrzení registrace, je uživatel připuštěn do systému. Zde má k dispozici náhled kalendáře lekcí a po dobití některé z částek na účet systému, je mu otevřena možnost rezervace míst na lekcích v kalendáři. Při rezervacích pak využívá částky, kterou má na tomto účtu, ze kterého je mu částka v příslušné výši strhávána. Velkou výhodou je pak možné bezhotovostní dobíjení účtu pomocí platební brány PayPal. Dále má k dispozici přehled pohybů na svém účtu, kdy při nastavení časového rozmezí uvidí přehledný tabulkový výpis.

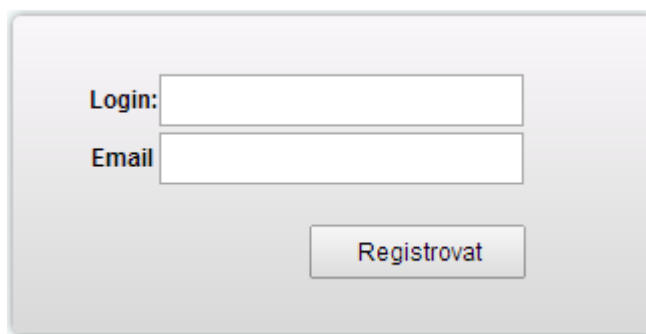
### Přihlášení a registrace do aplikace

Do samotné aplikace se dostaneme ze stránek spinning centra pod odkazem „Rezervační systém“. Při prvním vstupu do aplikace je klient vyzván k přihlášení do aplikace a v případě že si zatím nevytvořil uživatelský účet, bude se muset zaregistrovat.

### Registrace

Po kliknutí na odkaz „Registrace“ je uživatel vyzván k zadání tzv. Loginu a platné emailové adresy. Loginem se rozumí přihlašovací jméno, které bude uživatel používat při přihlášení do aplikace. Email je nutný pro ověření platnosti registrace uživatele a pro případnou budoucí komunikaci.

Po zadání platných údajů, kdy se kontroluje duplicita zadaných informací a jejich správný tvar, je uživateli zaslán potvrzovací email na zadanou adresu, který obsahuje aktivační link. Při otevření URL adresy linku se aktivuje nový účet uživatele. Je mu zaslán nový email, kterým dostane souhrn svých přihlašovacích údajů, kterými se následně přihlásí do aplikace. Aktivační link je pak platný pouze do použití, nebo do smazání registrace.

The image shows a registration form with a light gray background and rounded corners. It contains two input fields: the first is labeled 'Login:' and the second is labeled 'Email'. Below these fields is a button labeled 'Registrovat'.

Obrázek 14 - Registrační formulář

### Přihlášení

Při prvním přihlášení bezprostředně po registraci je uživatel přesměrován na doplnění svých osobních informací, které jsou částečně nutné pro ověření totožnosti klienta při příchodu na lekci.

### 8.2.1 Přehled

Po přihlášení se klientovi zobrazí základní přehled a uvítací stránka. Zde nalezne kalendář, nadcházejících lekcí počínaje aktuálním dnem. Odtud pak dále pokračuje v procházení aplikace.

Přímo zde je také možné vstoupit do registrací po kliknutí na konkrétní lekci v týdenním kalendáři, nebo vybrat některý ze zobrazených dní.



Obrázek 15 – Kalendář se zvýrazněnými dny konání lekcí

### 8.2.2 Moje Lekce

V záložce „Moje lekce“ najde uživatel tabulkový výpis lekcí, na kterých je přihlášen a přehled lekcí, kterých se zúčastnil. Historie lekcí je k dispozici 2 měsíce nazpět, kdy je systém automaticky smaže, aby nehromadil zbytečné informace.

V tabulkovém výpisu najdeme, kromě jména lekce a data konání také funkci pro storno lekce, kterým si může klient zrušit rezervaci.

#### Storno lekce

Storno lekce je funkce, která, už dle názvu, dává možnost zrušit rezervaci na lekci, které se klient z jakéhokoliv důvodu nepřeje zúčastnit. Při samotném stornování se pak stržená částka přičte zpět na účet klienta. Aby však byla zachována korektnost a férový přístup, musí se klient odhlásit nejpozději hodinu před konáním lekce samotné. Po této době je částka za lekci napevno zaúčtována a storno lekce nelze provést.

### **Přihlášení na lekci**

V sekci „Moje lekce“ pod odkazem „Přihlášení na lekci“ najdeme možnosti přihlášení na jednotlivé lekce. Prvním krokem je zobrazený kalendář, kde si vybereme den, kdy se chceme na lekci přihlásit. Dny, kdy jsou vypsány termíny lekcí, jsou zvýrazněny. Po kliknutí na vybraný den se ukáže časová osa dne s chronologickým výpisem lekcí v tomto dni.

Každá kolonka v časové ose znázorňuje lekci a základní informace o ní. Informace na časové ose jsou název lekce, čas konání, obsazenost, kapacita lekce a cena lekce. Po kliknutí na box lekce, se ve spodní části stránky rozbálí další detailní výpis lekce, a mimo základní informace se zde dozvíme lektora, již přihlášené uživatele. Nakonec je zde tlačítko rezervace na lekci, kterým se přihlásíme na daný termín. Tato rezervace pak není možná, pokud již lekce proběhla, pokud uživatel nemá dostatečný kredit, nebo je přihlášeným uživatelem zároveň lektor lekce.

Pokud je uživatel již přihlášen na zvolenou lekci, tlačítko rezervace se změní na tlačítko storno rezervace a po kliknutí je lekce stornována. Storno lekce pak není možné, pokud již lekce proběhla, právě probíhá, nebo je po expirační době.

pátek 20. června 2014

08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
											19:00-19:45 spinning student 4/10 75 Kč		

---

**Lekce spinning student**

Čas: 20. 06. 2014 19:00 - 19:45  
Místa: 4 / 10  
Pod vedením: malantikova

id	Uživatel	email
97	kreed	marty.sedlar@gmail.com
90	blecha	blecha@email.cz
71	hajkova	hajkova.p@seznam.cz
56	rossy568	rossy568@seznam.cz

[Stornuj rezervaci](#)

Obrázek 16 – Náhled obrazovky rezervací a přihlášení na lekci

### 8.2.3 Můj profil

Zde si uživatel může prohlédnout své osobní informace. Důležitější částí pak je nastavení údajů, jako například změna hesla, což se doporučuje bezprostředně po prvním přihlášení, kdy je účtu přiřazeno šestimístné heslo generované. Dále je zde editace osobních údajů jako jméno, příjmení, bydliště, telefon, datum narození. Tyto údaje nejsou povinné, nicméně se doporučují vyplnit, pro případ nutnosti komunikace s uživatelem jinak, nežli elektronickým kanálem.

Další důležitou informací je stav účtu, který se zobrazuje ve spodní části stránky profilu uživatele. Uživatel zde najde také historii pohybu na účtu aplikace, kde si může kdykoliv zkontrolovat, kdy si naposledy dobil, nebo kdy utratil za rezervaci a další služby. Pokud je částka nízká, nebo uživatel zatím ještě nedobíjel kredit, najde zde odkaz na dobítí kreditu a aktuální ceník dobíjení.

Poslední neméně důležitou věcí je odkaz „Voucher“, nás zavede na stránku optimalizovanou pro tisk, kde si uživatel vytiskne kopii svého profilu a QR kód, po jehož načtení jej obsluha v případě nutnosti jasně identifikuje.

kreed

Jméno:	Martin Sedlář
Věk:	27
Pohlaví:	muž
Telefon:	739479174
Bydliště:	Dětkovicev 198 79804 Určice
Narození:	17. 12. 1986
Poslední aktivita:	10.06.2014
Registrace	19. 02. 2014

Stav účtu: 100.00 [Historie](#)**PIN: 2568**

Dobij účet

Uprav mé údaje

Vytiskni voucher

Ukaž PIN

Obrázek 17 – Náhled uživatelského profilu

#### 8.2.4 Účet

Účet je klíčovou součástí aplikace, kam si uživatel dobije prostředky, které pak využívá pro rezervace lekcí. Tyto prostředky jsou ve formě finanční částky, kterou si uživatel může na účet poslat hned dvěma způsoby. Prvním je poměrně obyčejné dobítí v hotovosti na pokladně, kdy se musí uživatel fyzicky dostavit na recepci spinning-centra, kam si musí vložit částku v hotovosti a obsluha mu ručně kredit do systému přidá. Pak je tu ale druhá možnost a to elektronické platby. V popisované verzi je k dispozici online platba dnes nejoblíbenějšího bezhotovostního platebního portálu na světě PayPal. Pokud má uživatel účet pro plat-

by PayPal, pak si pohodlně z domu pomocí několika málo kliknutí dobije částku dle ceníku a může tyto prostředky okamžitě využívat.

V položce „Historie plateb“ se uživatel dozví pohyby na svém účtu, a to buďto posledních 8 transakcí, nebo může specifikovat časové rozmezí jednoduchým formulářem pod výpisovou tabulkou.

### Historie plateb

Aktuální stav účtu: 10 Kč

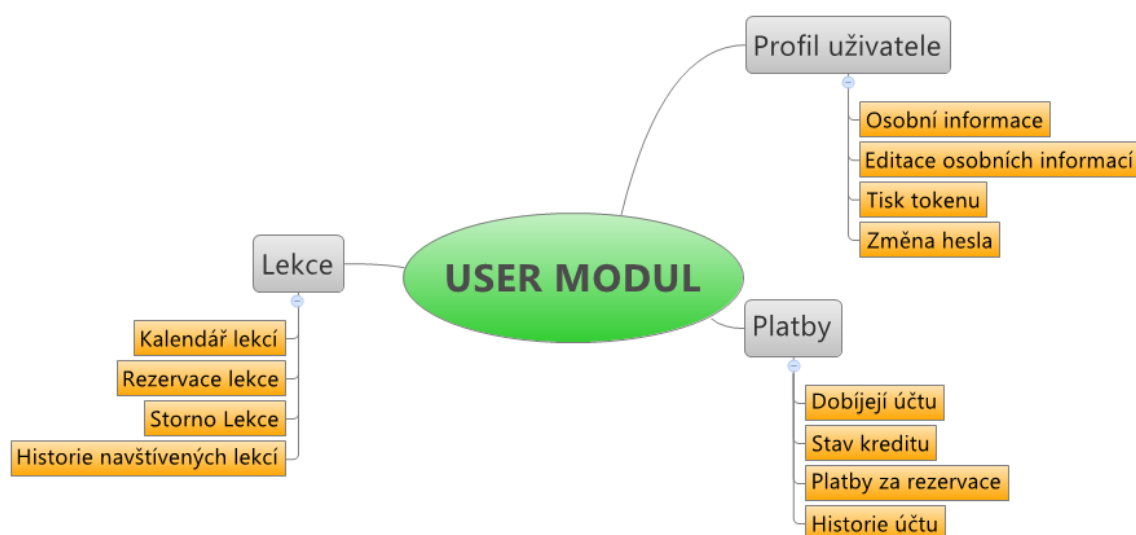
#### Platby za služby

id	čas	Popis	Cena
1	10. 06. 2014	Na pokladně 2	350
2	03. 06. 2014	Lekce mažoretky dne 16. 05. 2014	-1000
3	03. 06. 2014	Testovací	400
4	30. 05. 2014	Na pokladně	500
5	28. 05. 2014	Na pokladně	500
6	27. 05. 2014	cfgb	-20
7	27. 05. 2014	dan	-500

Od:

Do:

Obrázek 18 – Uživatelský výpis historie plateb



Obrázek 19 – Diagram struktury aplikace v uživatelském modulu

### 8.3 Identifikace uživatele

Kromě základní identifikace klienta při vstupu do aplikace je nutné v případě potřeby identifikovat klienta přímo na lekci v případě nesrovnalostí v obsazenosti, nebo manipulaci s účtem klienta přímo u obsluhy. K tomuto účelu, obsahuje uživatelský modul v sekci profilu uživatele možnost vytištění *tokenu* ve formě QR kódu, který může použít jako potvrzení své totožnosti. Po načtení si jej může obsluha vyčíst například připraveným mobilním telefonem a je přesměrována na skrytou webovou stránku s profilem uživatele.

Další možností potvrzení identity uživatele je shoda uživatelského jména a čtyřmístného PIN kódu, který má každý uživatel k nahlédnutí opět v sekci profilu. Obsluha se tedy pro identifikaci vyžádá některý z jeho atributů, například přihlašovací jméno a po zobrazení uživatelského profilu pro potvrzení jeho čtyřmístný PIN kód.

### 8.4 Automatická údržba aplikace

Aplikace shromažďuje spoustu dat a ukládá je do databáze. Do databáze také vstupuje a neustále z ní shromážděná data vyčítá, vyhledává a třídí. Čím větší je ale objem nastřádaných dat v databázi, tím se zvyšuje režie pro jejich manipulace a tím se zpomaluje běh celé aplikace. Proto je nutné zavést nástroj, který by se automaticky zbavoval již nepotřebných záznamů a tím zabránil zbytečnému zpomalování běhu aplikace.

Dále je nutné provádět některé změny týkající se běhu systému a operace plateb.

#### 8.4.1 Překlopení rezervací do plateb

Jelikož je nutné ponechat klientovy možnost zrušení rezervace a navrácení platby, je logika plateb za rezervace nastavena tak, že tyto prostředky jsou po přihlášení na lekci pouze rezervovány až do chvíle vypršení limitu zrušení rezervace a nejsou započítány do celkových plateb. Právě tato funkce má za úkol kontrolovat rezervace, které jsou již po limitu zrušení a započítat je do plateb klienta. Tímto jsou záznamy o platbách odděleny od rezervací a lekcí samotných a je možné si jejich historii zobrazit, i když lekce i s rezervací již dávno nejsou v databázi uloženy.



### 8.4.2 Lekce již proběhlé

Data o lekcích a jejich rezervacích, které již proběhly a nejsou tak aktuální, nejsou pro aplikaci nutné a s daty se dále nijak nepracuje. Proto se kontrolují záznamy lekcí, které proběhly více než před měsícem a tyto jsou i s rezervacemi automaticky smazány. Tato kontrola probíhá jednou denně v nočních hodinách.

### 8.4.3 Neaktivní uživatelé

Pokud je uživatelský účet dlouhodobě neaktivní, je uživatel upozorněn emailem a pokud nadále nereaguje, je jeho účet smazán a jeho kredit mu propadne. Tato kontrola se provádí jednou denně.

## 8.5 Rozpočet naší aplikace

Rozpočet na vývoj naší aplikace vychází z aktuálních údajů Českého statistického úřadu o průměrných platech odborníků v IT sféře, konkrétně v oborech programátorů a vývojářů webových aplikací. Dále tento rozbor vychází z časové náročnosti vývoje naší aplikace, kterou autor této práce vynaložil.

Průměrná čistá hodinová mzda programátora a webového vývojáře ve věkovém rozmezí 25-29 let činí v přepočtu 156 Kč.[údaj ČTU březen 2014]

Autor této práce a současně autor aplikace strávil jejím vývojem 70hodin čistého času.

Do nákladů započítáme i dobu nutnou na zaškolení obsluhy, instalaci, dopravu a další výdaje ve formě nákladů na hosting pro vývoj, konzultaci a další drobné náklady. Výsledná částka včetně rozboru rozpočtu je uvedena v následující tabulce.

Úkon	cena za jednotku	počet	Celkem
Programování PHP, JavaScript, MySQL	156 Kč	55 Hodin	8580 Kč
Úprava, HTML, CSS	156 Kč	10 Hodin	1560 Kč
Zaškolení, instalace	156 Kč	5 Hodin	780 Kč
Doprava	10 Kč	150 kilometrů	1500 Kč
Další výdaje	-	-	400 Kč
			<b>12 820 Kč</b>

Tabulka 2 - Tabulka nákladů na vývoj aplikace

Tyto náklady na vývoj aplikace jsou jednorázové a po instalaci a zavedení aplikace *Spinning robot* již další náklady nenastanou. Jediným pravidelným nákladem je platba za hostingové služby a doménu, na které aplikace poběží. Tyto se ale liší od zřizovatele a poskytovatele hostingu, a tak ji nelze z přesností určit. Navíc tento údaj nebyl uveden ani v oficiálních cenících konkurenčních komerčních systémů, takže ani v našem rozpočtu není tuto částku potřebné uvádět.

## 8.6 Testovací provoz

Testovací provoz byl, po konzultaci s vedoucím práce a provozovatelem spinning centra, proveden krátkodobě v reálném provozu, avšak s omezeným počtem uživatelů. Byla otestována funkčnost aplikace, kdy byly prověřeny všechny avizované funkce.

Před samotným testem bylo do aplikace zavedeno několik fiktivních lekcí, v různých časích, v různých délkách a různých cenách. A byla provedena příprava na vstup klientů do aplikace.

Dále bylo pro test postupně osloveno deset dobrovolníků z řad obyčejných uživatelů internetu, kterým byla popsána situace, byl jim přidělen fiktivní kredit na účet a s tímto kreditem si rezervovali dostupné lekce.

Test probíhal v průběhu několika dní a následně měli dobrovolníci vyjádřit své pocity z aplikace, jejího užívání, přehlednosti a spolehlivosti.

V průběhu testování bylo zjištěno několik drobných chyb, které byly obratem odstraněny.

Testovací verze je k dispozici na adrese <http://spinning.sbrothersweb.eu/>

Přihlašovací jméno: *admin*

Heslo: *spinningguard*

## ZÁVĚR

Úkolem této práce bylo prostudovat a navrhnout funkční rezervační systém pro konkrétní aplikaci ve spinning centru Mad Jack Zlín. Aplikace měla nahradit stávající zastaralý rezervační systém.

Při konzultaci s provozovatelem byla stanoveno a nastíněno několik požadavků, které byly shrnuté v kapitole 1. Při shrnutí požadavků bylo přihlédnuto také k možnostem stávajících komerčních systémů a webových aplikací podobného zaměření.

Studium komerčních systémů pro rezervaci a elektronické platby přineslo zjištění. Prvním je ekonomická náročnost, která je z dlouhodobého hlediska podstatně vyšší, než u námi vytvořené aplikace. Zatímco vývoj aplikace Spinning Robot by vyšla při shrnutí všech nákladů jednorázově na 12 820 korun českých, tak jediný z uvedených komerčních systémů [fittless-reservace.cz](http://fittless-reservace.cz), který nabízel v ceníku jednorázovou platbu za aplikaci, vyšel na 19 000 korun českých, a to pouze za licenci, bez instalace úpravy a dalších nákladů. Ostatní uvedené nabízeli pouze variantu za měsíční poplatek mezi 500 a 750 korunami s DPH, a to také bez dalších nákladů. Z dlouhodobého hlediska se i nejlevnější produkt v ceníku za 500 korun měsíčně, stává za dva roky oproti naší aplikaci nevýhodným.

Dá se říci, že identifikace uživatele a levná náhrada k RFID řešení je celá aplikace jako celek. Aplikace identifikuje uživatele především na straně klienta při rezervaci, kdy při vstupu do aplikace je klient identifikován pomocí uživatelského jména a hesla. Po vstupu do aplikace a rezervaci místa na lekci je mu částka za lekci automaticky stržena a tím je místo na lekci z pohledu provozovatele využito, chceme-li zúčteno a už pro něj není důležité, jestli se osoba na lekci dostaví, nebo ne. Nutnost identifikace a autentifikace přijde vhod ve chvílích, kdy nastává nesrovnalost v obsazení lekce a je nutné zjistit, kdo má rezervaci a kdo ne. Zde jsou dvě možnosti, klient se může identifikovat tzv. *tokenem* ve formě QR kódu, který si může vytisknout po přihlášení do uživatelské sekce aplikace, nebo při nahlášení svého přihlašovacího jména a doplněním bezpečnostního čtyřmístného PIN kódu. Tímto je uživatel jasně identifikován.

Aplikace typu klient-server, která byla nazvána SpinningRobot, byla navržena za 70 hodin čistého času s použitím kombinace programovacích jazyků PHP, JavaScript, HTML, CSS a databázi MySQL. Tato pak splnila všechny požadavky, což prokázal i test provozu, který byl reálný, avšak po konzultaci s vedoucím práce omezen počtem testovacích subjektů na

10. Test prokázal několik nedokonalostí, které byli v průběhu testu a bezprostředně po něm odstraněny. Test tak proběhl úspěšně, což lze soudit i s pozitivních reakcí testujících respondentů.

Nespornou výhodou je také její modularita, kdy použitá architektura MCV dává aplikaci přehlednější tvar z pohledu programátora a jednoduchou rozšiřitelnost.

Autor doufá, že po instalaci do skutečného provozu bude aplikace sloužit tak, jak byla navržena, k plné spokojenosti provozovatele. Také je očekávaná další vývoj aplikace ve spolupráci s provozovatelem spinning centra Mad Jack.

## SEZNAM POUŽITÉ LITERATURY

- [1] Fitness-rezervace.cz. [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://www.fitness-rezervace.cz/cenik-rezervacniho-systemu.html>
- [2] SuperSaas. [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://www.supersaas.cz/info/cenik>
- [3] ISPORT SYSTEM. [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://www.isportsystem.cz/cenik/>
- [4] ŠČUREK, Radomír. VŠB TU OSTRAVA. Biometrické metody identifikace osob v bezpečnostní praxi: Studijní text [PDF]. 2008 [cit. 2014-06-10]. Dostupné z: [http://www.biometrickypodpis.cz/PDF/biometricke\\_metody.pdf](http://www.biometrickypodpis.cz/PDF/biometricke_metody.pdf)
- [5] Přístupové systémy k identifikaci osob: Časopis pro elektrotechniku. ZEŽULA, Radek. UTKO FEKT VUT. Elektrovue [online]. 2002 [cit. 2014-06-10]. Dostupné z: <http://www.elektrovue.cz/clanky/02054/index.html>
- [6] QR kód – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-06-09]. Dostupné z: [http://cs.wikipedia.org/wiki/QR\\_k%C3%B3d](http://cs.wikipedia.org/wiki/QR_k%C3%B3d)
- [7] KOFLER, Michael. *Mistrovství v MySQL 5: [kompletní průvodce webového vývoje]*. Vyd. 1. Brno: Computer Press, 2007, 805 s. ISBN 978-802-5115-022.
- [8] ULLMAN, Larry. *PHP a MySQL: názorný průvodce tvorbou dynamických WWW stránek*. Vyd. 1. Brno: Computer Press, 2004, 534 s. ISBN 80-251-0063-4.
- [9] ZAJÍC, Petr. MySQL (23) - relace 1:N a N:N: Linux Software. In: [online]. [cit. 2014-06-09]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=854-§](http://www.linuxsoft.cz/article.php?id_article=854-§)
- [10] LEHOCKÝ, Zdeněk. Programujte.com - Přehled datových typů v MySQL. In: [online]. [cit. 2014-06-09]. Dostupné z: <http://programujte.com/clanek/2007052903-prehled-datovych-typu-v-mysql/>
- [11] GUTMANS, Andi. *Mistrovství v PHP 5*. Vyd. 2. Brno: Computer Press, 2007, 655 s. ISBN 978-80-251-1519-0.
- [12] klenot.cz: domov pro vaše stránky. [online]. [cit. 2014-05-25]. Dostupné z: <http://www.klenot.cz/cs/podpora/OpenSource/>

- [13] Wikipedie: otevřená encyklopedie. [online]. [cit. 2014-05-23]. Dostupné z: [http://cs.wikipedia.org/wiki/Nette#cite\\_note-2](http://cs.wikipedia.org/wiki/Nette#cite_note-2)
- [14] Wikipedie: otevřená encyklopedie. [online]. [cit. 2014-05-23]. Dostupné z: [http://cs.wikipedia.org/wiki/Model-view-controller#mediaviewer/Soubor:MVC\\_Diagram\\_%28Model-View-Controller%29.svg](http://cs.wikipedia.org/wiki/Model-view-controller#mediaviewer/Soubor:MVC_Diagram_%28Model-View-Controller%29.svg)
- [15] Nette: Dokumentace. [online]. [cit. 2014-05-23]. Dostupné z: <http://doc.nette.org/cs/2.1/templating>
- [16] ŠKULTÉTY, Rastislav. *JavaScript: programujeme internetové aplikace*. Vyd. 1. Praha: Computer Press, 2004, 224 s. ISBN 80-251-0144-4.
- [17] Wikipedie: otevřená encyklopedie. [online]. [cit. 2014-05-23]. Dostupné z: <http://cs.wikipedia.org/wiki/JQuery>
- [18] Cross-site scripting. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-06-10]. Dostupné z: [http://cs.wikipedia.org/wiki/Cross-site\\_scripting](http://cs.wikipedia.org/wiki/Cross-site_scripting)
- [19] Cross-site request forgery. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-06-10]. Dostupné z: <http://cs.wikipedia.org/wiki/CSRF>
- [20] SQL injection. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-06-10]. Dostupné z: [http://cs.wikipedia.org/wiki/SQL\\_injection](http://cs.wikipedia.org/wiki/SQL_injection)
- [21] Nette: Dokumentace. [online]. [cit. 2014-05-23]. Dostupné z: <http://doc.nette.org/cs/2.1/database>

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MySQL	z ang. My Structured Query Language – systém pro řízení databází.
GPL	Z ang. General Public Licence – Obecná veřejná licence.
PHP	Hypertext Preprocessor.
GNU	Z ang. General Public Licence – Všeobecná veřejná licence
BUG	V překladu „Brouk“, neboli chyba v programu
Debugging	V překladu odstranění brouků, zde programátorský výraz pro nalezení a odstranění chyb programu
XSS	Z ang. Cross-site-scripting, jedná se o narušení www stránek využitím bezpečnostních děr scriptů
CRON	Systémový nástroj, který spouští programy v předem definovanou dobu

**SEZNAM OBRÁZKŮ**

Obrázek 1 - Karta s magnetickým proužkem .....	22
Obrázek 2 – Náhled QR kódu se zakódovaným textovým řetězcem.....	24
Obrázek 3 - Relace 1:N [9] .....	27
Obrázek 4 – Relace N:N [9] .....	27
Obrázek 5 – Schéma práce architektury MCV [2].....	30
Obrázek 6 – Logo nette Framework [13].....	31
Obrázek 7 – Struktura databáze a propojení tabulek .....	38
Obrázek 8 – Náhled obrazovky při přihlášení do aplikace .....	41
Obrázek 9 – Obrazovka přehledů uživatelů.....	42
Obrázek 10 – tabulkový seznam lekcí .....	44
Obrázek 11 – Přidávání lekce do databáze .....	45
Obrázek 12 – Přehledy plateb a rezervací.....	46
Obrázek 13 – Diagram aplikace Administrátorské sekce .....	49
Obrázek 14 - Registrační formulář .....	50
Obrázek 15 – Kalendář se zvýrazněnými dny konání lekcí.....	51
Obrázek 16 – Náhled obrazovky rezervací a přihlášení na lekci .....	53
Obrázek 17 – Náhled uživatelského profilu.....	54
Obrázek 18 – Uživatelský výpis historie plateb .....	55
Obrázek 19 – Diagram struktury aplikace v uživatelském modulu .....	55



**SEZNAM TABULEK**

Tabulka 1 – Ceny nejpoužívanějších komerčních rezervačních online systémů ..... 19

Tabulka 2 - Tabulka nákladů na vývoj aplikace ..... 57

**PŘÍLOHA P I: DVD SE SOUBORY DPPRACE.PDF A PINNINGROBOT.ZIP**