



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DATABÁZOVÉ ÚLOŽIŠTĚ VIRTUÁLNÍ ČEKÁRNY

VIRTUAL WAITING ROOM DATA BASE STORAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATEJ SADLOŇ

VEDOUCÍ PRÁCE
SUPERVISOR

MARTIN KOLÁŘ, M.Sc.

BRNO 2016

Abstrakt

Táto práca je súčasťou projektu zvaného Virtuálna čakáreň. Jedná sa o multiplatformovú aplikáciu ktorá umožňuje užívateľom zarezervovať si termín u niektorého z poskytovateľov bez potreby osobného čakania v čakárni. Práca popisuje návrh a implementáciu databázy, knižnice pre prácu s touto databázou a klienta pre správu poskytovateľov a čakární. Pre tento účel boli použité technológie ako MySQL, PHP a HTML.

Abstract

This thesis is part of project called Virtual Waiting Room. It is a cross-platform application which allows users to book a term with a provider without the need to personally wait in a waiting room. The thesis describes the design and implementation of database, library for work with this database, and a client for the administration of providers and waiting rooms. For this purpose, was used technologies such as MySQL, PHP and HTML.

Kľúčové slová

databáza, knižnica, klientska aplikácia, PHP, MySQL, HTML, XAMPP, MyISAM, InnoDB.

Keywords

database, library, client application, PHP, MySQL, HTML, XAMPP, MyISAM, InnoDB.

Citácia

SADLOŇ, Matej. *Databázové úložiště virtuální čekárny*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Kolář Martin.

Databázové úložiště virtuální čekárny

Prehlásenie

Prehlasujem, že som túto bakalárskou prácu vypracoval samostatne pod vedením pána Koláře Martina, M.Sc. . Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Matej Sadloň
11. mája 2016

© Matej Sadloň, 2016.

Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.

Obsah

1	Úvod	4
2	Existujúce riešenia	5
3	Použité technológie	9
3.1	XAMPP	9
3.2	Apache HTTP server	10
3.3	MySQL	10
3.4	PHP	11
3.5	HTML	11
4	Návrh	12
4.1	Neformálna špecifikácia	12
4.2	Návrh štruktúry databázy	12
4.2.1	Užívatelia	13
4.2.2	Poskytovatelia	13
4.2.3	Čakárne	13
4.2.4	Otváracie hodiny	14
4.2.5	Časové rámce	14
4.2.6	Termíny	14
4.2.7	Služby	14
4.2.8	Administrátori	14
4.3	Návrh uloženia dát	14
4.3.1	MyISAM	14
4.3.2	InnoDB	15
4.4	Návrh knižnice pre prácu s databázou	15
4.5	Návrh klienta pre správu poskytovateľov a čakární	16
5	Implementácia	18
5.1	Vytvorenie a konfigurácia databázy	18
5.1.1	Tabuľka providers	18
5.1.2	Tabuľka queue	18
5.1.3	Tabuľka opening_hours	19
5.1.4	Tabuľka services	19
5.1.5	Tabuľka offer_service	20
5.1.6	Tabuľka timeslots	20
5.1.7	Tabuľka orders	20
5.1.8	Tabuľka users	21

5.1.9	Tabuľka <code>admins</code>	21
5.2	Implementácia PHP knižnice	21
5.2.1	Funkcia <code>__construct</code>	22
5.2.2	Funkcia <code>__destruct</code>	22
5.2.3	Funkcia <code>create_queue</code>	22
5.2.4	Funkcia <code>delete_queue</code>	22
5.2.5	Funkcia <code>set_hours</code>	23
5.2.6	Funkcia <code>get_hours</code>	23
5.2.7	Funkcia <code>create_service</code>	23
5.2.8	Funkcia <code>delete_service</code>	24
5.2.9	Funkcia <code>set_service</code>	24
5.2.10	Funkcia <code>create_provider</code>	24
5.2.11	Funkcia <code>update_provider</code>	24
5.2.12	Funkcia <code>delete_provider</code>	25
5.2.13	Funkcia <code>provider_info</code>	25
5.2.14	Funkcia <code>provider_queues</code>	25
5.2.15	Funkcia <code>provider_services</code>	26
5.2.16	Funkcia <code>find_provider</code>	26
5.2.17	Funkcia <code>queue_info</code>	27
5.2.18	Funkcia <code>queue_state</code>	27
5.2.19	Funkcia <code>set_order</code>	28
5.2.20	Funkcia <code>clear_order</code>	28
5.2.21	Funkcia <code>user_conf_order</code>	28
5.2.22	Funkcia <code>provider_conf_order</code>	28
5.2.23	Funkcia <code>register_admin</code>	29
5.2.24	Funkcia <code>delete_admin</code>	29
5.2.25	Funkcia <code>login_admin</code>	29
5.2.26	Funkcia <code>change_admin_psswd</code>	29
5.2.27	Funkcia <code>all_admins</code>	30
5.2.28	Funkcia <code>register_user</code>	30
5.2.29	Funkcia <code>delete_user</code>	30
5.2.30	Funkcia <code>login_user</code>	30
5.2.31	Funkcia <code>auth_user</code>	31
5.3	Implementácia klienta pre správu poskytovateľov a čakární	31
5.3.1	Programová časť klienta	32
5.3.2	Grafická časť klienta	33
6	Testovanie	36
7	Záver	37
	Literatúra	39
	Prílohy	40
	Zoznam príloh	41

A	Obsah CD	42
A.1	database.zip	42
A.2	lib.zip	42
A.3	client.zip	42

Zoznam obrázkov

2.1	Detail čakárne v aplikácii cakarenonline.sk	5
2.2	Výber služby v aplikácii WebCall	6
2.3	Výber dátumu termínu v aplikácii WebCall	7
2.4	Výber času termínu v aplikácii WebCall	8
2.5	Vyplnenie osobných údajov a potvrdenie termínu v aplikácii WebCall	8
3.1	Kontrolný panel programu XAMPP	10
4.1	ER diagram databáze	13
4.2	Diagram prípadov užitia pre klienta	17
5.1	Schéma implementovanej databáze	19
5.2	Koreňový adresár obsahujúci súbory klienta.	32
5.3	Adresár podstránky obsahujúci jednotlivé podstránky klienta.	32
5.4	Grafický vzhľad klienta - detail poskytovateľa.	34
5.5	Chovanie klienta v prípade zadania nesprávnych údajov.	34
5.6	Chovanie klienta v prípade rušenia poskytovateľa.	35

Kapitola 1

Úvod

V dnešnej dobe si nikto nevie predstaviť život bez inteligentného telefónu. Každodenné činnosti a povinnosti si zjednodušujeme za pomoci mobilných aplikácií ktoré nám šetria čas a energiu. Z tohto poznatku vychádza aj aplikácia virtuálnej čakárne, ktorej súčasťou je aj táto práca. Určite sa vám už stalo, že ste museli ísť lekárovi alebo niekam na úrad a keď ste sa dostavili do čakárne zarazene ste zistili že sa dnes aj tak nedostanete na radu a zbytočne ste premrhali cestu a čas. Vďaka aplikácii virtuálnej čakárne sa vám to už znova nestane. Aplikácia umožňuje prehľad jednotlivých poskytovateľov, stavu ich čakární a jednoduché prihlásenie sa na určitý termín iba za pomoci mobilného telefónu, prípadne stolného počítača.

Cielom tejto práce je vytvorenie databázy, rozhrania pre prácu s touto databázou a webového klienta pre správu poskytovateľov a čakární. Databáza uchováva informácie o poskytovateľoch, poskytovaných službách, aktuálnom stave jednotlivých čakární či o ich otváracích hodinách. Tieto informácie sú ukladané a načítané z databázy za pomoci knižnice, ktorú následne využíva či už webový klient určený na správu, alebo server komunikujúci s jednotlivými mobilnými aplikáciami. Aplikácia rozlišuje rôzne role užívateľov ako napríklad poskytovateľ, bežný užívateľ, alebo administrátor v prípade webového klienta pre správu. Podľa tejto role je možné zisťovať a upravovať jednotlivé informácie v databáze.

Mobilná aplikácia Virtuálna čakáreň, ktorej súčasťou je táto práca je vyvíjaná pre operačné systémy Android, iOS a Microsoft Windows.

Kapitola 2

Existujúce riešenia

V tejto kapitole sa bližšie pozrieme na to akým spôsobom je možné riešiť problematiku virtuálnej čakárne a taktiež si ukážeme niektoré z existujúcich riešení.

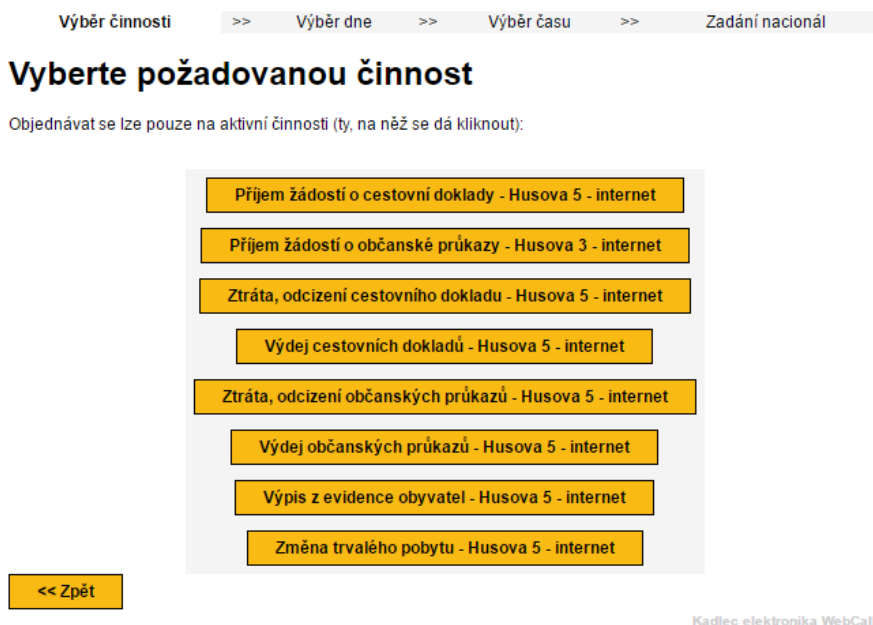
Jedným z existujúcich riešení je webová stránka <http://www.cakarenonline.sk/>. Táto webová stránka poskytuje možnosť zaradenia sa v čakárni u niektorého z poskytovateľov. Momentálne je táto stránka ešte vo vývoji a zatiaľ poskytuje iba základné informácie o stave čakárne, ako sú otváracie hodiny a poradie čakajúcich. Detail čakárne je možné vidieť na obrázku 2.1.

The screenshot shows the website CakarenOnline.sk. The main heading is "Informácie o čakárni". On the left, there is a "Dôležité info" section with a red warning icon, stating "Všeobecný lekár pre dospelých". Below it, a "Máme otvorené" section with a green clock icon lists opening hours: po: 8:00 - 13:00, ut: 8:00 - 13:00, st: 11:00 - 16:00, št: 8:00 - 13:00, pi: 8:00 - 11:00, so: Neordinujeme, ne: Neordinujeme. The "Adresa" section with a house icon provides the address: MUDr. Zuzana Čečetková, Ružinovská 10, 4.poschodie, č.dverí 412, 820 07, Bratislava. On the right, the "Aktuálne poradie" section shows a queue list with the text "posledné načítanie 12:09, obnoviť". The queue list contains the following names: 1, Mracna, Guttmanova, Mraz, Guttmanova, Vincent, Ulehlova, Ulehlova, Malanik, Durova, Sysova, Balazova, and Obsolkova. The names Sysova and Balazova are highlighted with a green border. At the bottom of the page, there is a footer with the text "(C)2016 - CakarenOnline.sk".

Obr. 2.1: Detail čakárne v aplikácii cakarenonline.sk

Ďalším existujúcim riešením je tzv. systém WebCall. Tento systém využíva napríklad mesto Brno na svojich webových stránkach a pomocou neho umožňuje registráciu termínu napríklad pri vybavovaní cestovného pasu. Registrácie termínu rozdeľuje do niekoľkých krokov. Po vstupe do systému je potrebné zvoliť službu ktorú požadujeme (obr. 2.2), následne sa nám zobrazí kalendár s výberom možných dátumov termínu (obr:12). Po výbere dátumu je zobrazená možnosť výberu času termínu (obr. 2.4), ktorá je nasledovaná zadaním kontaktných údajov a potvrdením termínu (obr. 2.5).

Obe z týchto riešení sa javia ako užitočné a poskytujú rôzne pohľady na riešenie problematiky virtuálnej čakárne.



Obr. 2.2: Výber služby v aplikácii WebCall

Červen 2016:

Po	Út	St	Čt	Pá	So	Ne
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Červenec 2016:

Po	Út	St	Čt	Pá	So	Ne
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

<< Zpět

Legenda:

AA

- Na tento den se lze nyní objednat

BB

- Tento den jsou již všechny termíny k objednání obsazeny

CC

- Na tento den se budete moci objednat až v následujících dnech

Obr. 2.3: Výber dátumu termínu v aplikácii WebCall

Vyberte čas, na kdy se chcete objednat:

Pouze doby, na které se dá ještě objednat jsou aktivní (lze na ně kliknout).

- Objednávejte se na činnost Příjem žádostí o občanské průkazy - Husova 3 - internet

- Objednávejte se na den Úterý 31.5.2016

7:00	7:10	7:20	7:30	7:40	7:50	8:00	8:10	8:20
8:30	8:40	8:50	9:00	9:10	9:20	9:30	9:40	9:50
10:00	10:10	10:20	10:30	10:40	10:50	11:00	11:10	11:20
11:30	11:40	11:50						

<< Zpět

Kadlec elektronika WebCall

Obr. 2.4: Výber času termínu v aplikácii WebCall

Zadejte své kontaktní údaje:

- Objednávejte se na činnost Příjem žádostí o občanské průkazy - Husova 3 - internet

- Objednávejte se na den Úterý 31.5.2016

- Objednávejte se na dobu 9:10

Jméno a příjmení: *	<input type="text"/>
Telefonní kontakt:	<input type="text"/>
(00420123456789, +420123456789, 123456789)	
<input type="button" value="Objednat"/>	

<< Zpět

Kadlec elektronika WebCall

Obr. 2.5: Vyplnění osobných údajov a potvrdenie termínu v aplikácii WebCall

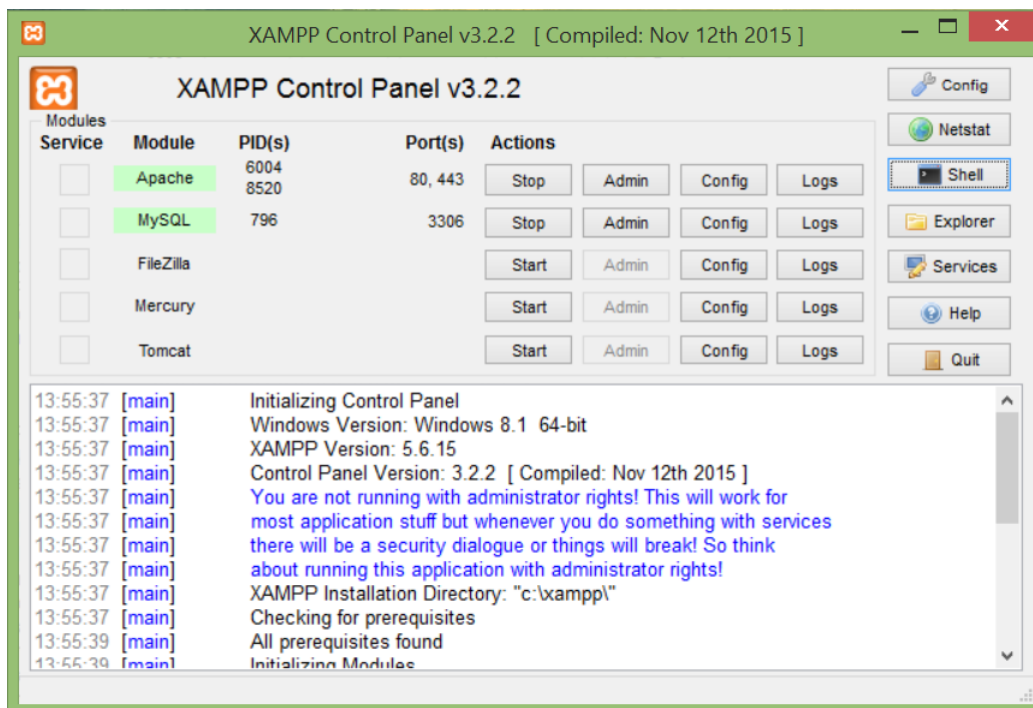
Kapitola 3

Použité technológie

Táto kapitola sa venuje problematike dostupných technológií, ktoré je možné následne použiť pre návrh a vytvorenie databáze, implementáciu PHP knižnice a vytvorenie webového klienta pre správu.

3.1 XAMPP

XAMPP[9] je verejne dostupný a multiplatformový softwarový balík spájajúci rôzne webové technológie do jedného celku vyvinutý spoločnosťou Apache Friends. Obsahuje hlavne Apache HTTP server, MySQL databázu a interpretery pre skriptovacie jazyky ako napríklad PHP alebo Pearl, čo taktiež reprezentujú písmená v jeho názve (Multiplatformnosť - X, Apache - A, MySQL - M, PHP - P, Perl - P). V princípe sa jedná o odľahčenú verziu Apache distribúcie určenej pre rýchlu a jednoduchú tvorbu lokálnych webových serverov pre testovacie účely. Pre jeho použitie stačí jednoduchá úvodná konfigurácia ako napr. zmena hesiel pre prístup do databázy či zmena niektorých parametrov Apache serveru na základe našich potrieb. Keďže je XAMPP multiplatformový funguje spoľahlivo na rôznych platformách ako je Linux, Mac alebo Windows. Pri tvorbe tejto práce bol použitý software XAMPP vo verzii 5.6.15 obsahujúci hlavne Apache server v2.4.4, MySql community server v5.5.32, interpreter pre PHP v5.6.15 a rozhranie pre prácu s databázou phpMyAdmin v4.0.4. Jeho hlavnou výhodou je spojenie všetkých podstatných prvkov pre vývoj webovej aplikácie ako napr. spojenie MySQL databáze a phpMyAdmin pre jej pohodlné ovládanie pomocou užívateľského rozhrania a taktiež jednoduché ovládanie jednotlivých prvkov pomocou kontrolného panelu (obr. 3.1).



Obr. 3.1: Kontrolný panel programu XAMPP

3.2 Apache HTTP server

Apache HTTP server[4], ďalej iba Apache, je svetovo najpoužívanejší softwarový webový server. Je založený na NCSA HTTPd serveri a jeho vývoj začal v roku 1995 spoločnosťou Apache Software Foundation. Jeho hlavnou úlohou je odpovedanie na HTTP požiadavky klientov a zasielanie webového obsahu cez internet. Taktiež poskytuje rôzne webové služby ako napr. podpora XML či CGI. Medzi jeho hlavné výhody patrí rýchlosť, spoľahlivosť a bezpečnosť. Taktiež je veľmi flexibilný a dokáže plniť rôzne požiadavky za pomoci rozšírení či modulov. Apache je napísaný v jazykoch C a XML. Najčastejšie je využívaný na Unixových systémoch ale je dostupný aj na platformách ako eComStation, Microsoft Windows, NetWare, OpenVMS, OS/2 a TPF.

3.3 MySQL

MySQL[7] je systém pre manažovanie relačnej databázy (relation database management system - RDBMS). Jeho prvá verzia bola vydaná 23. mája 1995 a patrí medzi najpoužívanejšie RDBMS vo svete. Názov MySQL je zložený z mena dcéry jedného zo spoluvlastníkov, My a SQL zastávajúceho akronym pre Structured Query Language čiže štruktúrovaný dopytovací jazyk. Do tejto skupiny jazykov patrí napr. aj PL/SQL využívaný firmou Oracle v jej databázových systémoch. Hlavnou výhodou oproti PL/SQL je to, že MySQL zahŕňa aj databázový systém, pričom nie je potrebná zložitá inštalácia samotnej databázy ako pri PL/SQL.

MySQL je častou voľbou hlavne pri webových aplikáciách, tvorí centrálny prvok vo veľa webových aplikačných softvérových balíkoch a taktiež tvorí základ pre rôzne fork projekty akými je napr. MariaDB či Drizzle.

MySQL je napísaný v programovacích jazykoch C a C++ a parser pre SQL je napísaný v jazyku yacc. Dokáže pracovať na rôznych platformách ako AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD a mnohých ďalších.

3.4 PHP

PHP alebo aj Hypertext Preprocessor[8] je široko využívaný verejne dostupný skriptovací jazyk ktorý sa syntaxou podobá na jazyk C. Bol vytvorený hlavne pre potrebu vývoja webových technológií ale ako programovací jazyk je využívaný pre rôzne účely.

Hlavnou výhodou PHP je to, že PHP kód môže byť vložený do HTML kódu alebo môže byť použitý v kombinácii s rôznymi webovými šablónovými systémami, systémami pre správu webového obsahu alebo webovými rámcami zvanými tiež frameworky. PHP kód je spracovávaný PHP interpreterom ktorý je zväčša implementovaný ako modul na webovom serveri. Na tomto serveri je taktiež uložený spúšťaný PHP skript. Výsledkom interpretovaného a spusteného PHP kódu môže byť akýkoľvek typ dát ako aj obrázok. Tieto dáta následne webový server spája napr. s generovanou webovou stránkou a vytvára výslednú stránku, ktorá je následne zobrazená klientovi. PHP neukladá žiadne dlhodobé údaje a ani informácie o vykonanej činnosti. Jeho ďalšou veľkou výhodou je to že v základnej verzii implementuje funkcie pre prácu s MySQL databázou a teda nevyžaduje žiadne rozšírenia.[2]

Implementácia samotného PHP je napísaná hlavne v programovacom jazyku C s niektorými modulmi napísanými v jazyku C++. Je podporovaný pre Unixové a Windows operačné systémy.

V tejto práci bol využitý tento jazyk vo verzii 5.6.15.

3.5 HTML

Hypertextový značkovací jazyk (HyperText Markup Language)[5], je štandardný značkovací jazyk využívaný pre popis webových dokumentov. Popri CSS a JavaScripte je HTML základným prvkom pre tvorbu webových stránok či tvorbu užívateľských rozhraní pre mobilné a webové aplikácie. HTML je využívaný ako hlavný spôsob, ktorým sú zobrazované informácie vo webovom prehliadači.

HTML vytvára pomocou značiek štruktúrovaný dokument, ktorý je rozdelený na rôzne sémantické časti ako napr. nadpisy, paragrafy, zoznamy, formuláre, odkazy a mnohé ďalšie. Samotný HTML je statický pričom vytvára stromovú štruktúru zobrazovanej entity nazývanú DOM model. Tento model využívajú iné programovacie jazyky ako napríklad JavaScript a vďaka nemu môžu túto entitu upravovať za behu a tak do nej vniesť prvok dynamičnosti. Toto je tiež jeden z hlavných dôvodov, prečo je v praxi HTML používané spolu s ďalšími programovacími jazykmi či technológiami. Zväčša sa jedná o už spomínaný JavaScript či PHP, ktorý je do HTML dokumentu vložený ako skript alebo CSS ktoré odkazovaním na jednotlivé značky HTML dokumentu definuje ich vzhľad a umiestnenie v danom dokumente.

Pre potreby projektu je využívaná verzia HTML 5.

Kapitola 4

Návrh

Táto kapitola pojednáva o neformálnej špecifikácii a problematike návrhu jednotlivých častí aplikácie virtuálnej čakárne zahrnutých v tejto práci. Je v nej uvedený návrh databázy, z ktorého vychádza jej výsledná schéma (obr. 5.1). Ďalej je tu preberaná problematika návrhu knižnice pre prácu s touto databázou. Záverom sa táto kapitola venuje návrhu klienta pre správu poskytovateľov a čakární. Z tohto návrhu následne vychádza diagram prípadov užívania (obr. 4.2).

4.1 Neformálna špecifikácia

Cieľom práce je vytvoriť databázové úložisko, ďalej iba databáza, pre aplikáciu virtuálnej čakárne a knižnicu pre prácu s touto databázou. Taktiež je potrebné vytvoriť webového klienta pre vytváranie a správu poskytovateľov a jednotlivých čakární, ktorý spolu so serverom (komunikuje s užívateľskými klientmi) využíva toto úložisko a pracuje s ním za pomoci už spomínanej knižnice.

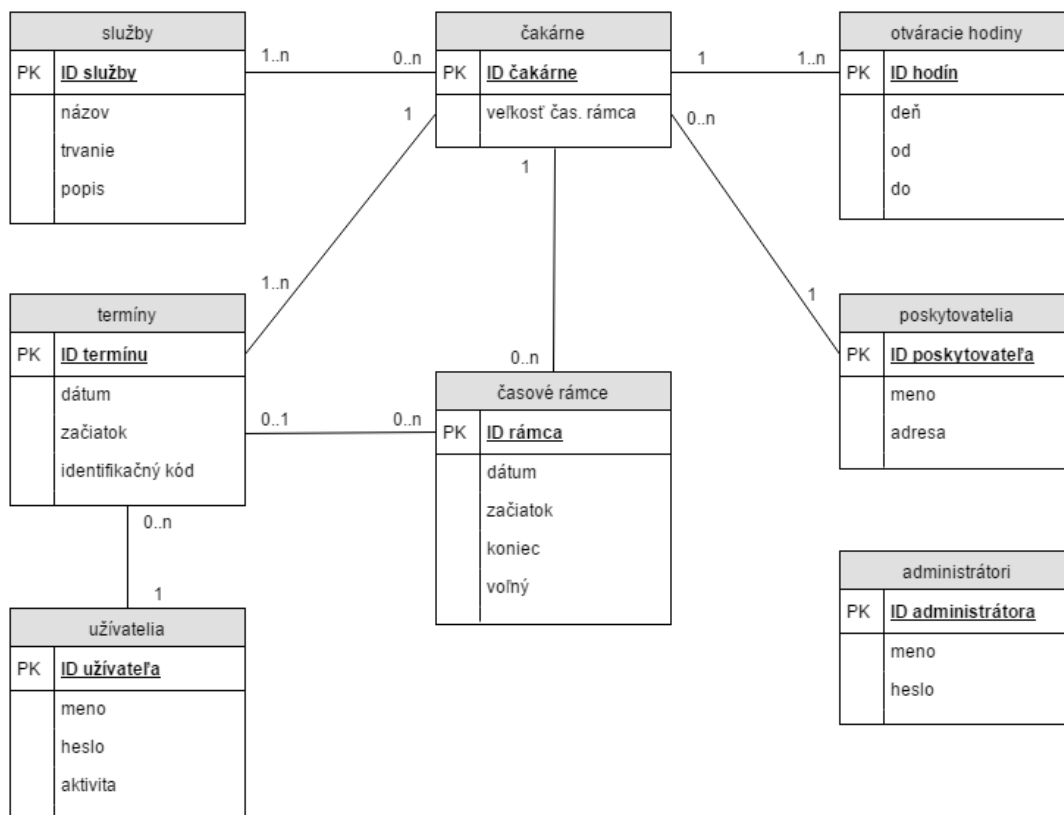
Pre účely virtuálnej čakárne je potrebné v databáze zachovávať informácie o užívateľoch, poskytovateľoch a o samotných čakárňach spolu s ich aktuálnym stavom. Ďalej je nutné vytvoriť knižnicu pre prácu s touto databázou tak, aby umožňovala registráciu nových užívateľov či vytvorenie nových poskytovateľov a čakární, čiže vytvorenie základných prvkov aplikácie. Následne musí dokázať s týmito prvkami pracovať, zistiť ich aktuálny stav, upraviť tento stav, prípadne ich úplne odstrániť z databázy. Táto knižnica má taktiež za úlohu zabezpečenie autentifikácie jednotlivých užívateľov. Funkcionality tejto knižnice budú následne využívať jednotliví klienti napr. pri vyhľadaní poskytovateľa, pri zisťovaní voľných termínov pre určitú čakáreň alebo pri prihlásení sa na daný termín za pomoci dotazov na server.

4.2 Návrh štruktúry databázy

V nasledujúcej podkapitole je vysvetlený význam jednotlivých entít a vzťahov medzi nimi potrebných pre správnu implementáciu databázy. Pre lepšie pochopenie navrhnutej schémy databázy bol zostavený ER diagram zobrazujúci dané entity a prepojenie medzi nimi na obrázku 4.1.

Jeden z hlavných problémov ktoré musí databáza riešiť je to ako budú vytvárané jednotlivé termíny. Keďže rôzne termíny môžu mať rôzne trvanie, tak sa ako najpraktickejší ponúka spôsob využívajúci časové rámce. Tieto rámce rozdeľujú deň na časové úseky určitej

veľkosti. Spojením niekoľkých časových rámcov je následne možné vytvoriť termín s dĺžkou násobku veľkosti rámca. Tento spôsob ponúka pri vhodne zvolenej veľkosti rámca veľmi flexibilné riešenie daného problému.



Obr. 4.1: ER diagram databáze

4.2.1 Užívatelia

Účelom tejto entity je reprezentovať jednotlivých užívateľov a základné informácie o nich. Zaznamenáva užívateľovo prihlasovacie meno, heslo, poslednú aktivitu a ďalšie doplnkové informácie.

4.2.2 Poskytovatelia

Táto entita predstavuje jednotlivých poskytovateľov služieb, ktorými môžu byť napr. doktor alebo kaderník. Zaznamenáva základné informácie o nich, ktorými sú napr. ich názov či adresa.

4.2.3 Čakárne

Entita čakárne slúži na popis jednotlivých čakární, zaznamenáva o nich informácie ako dátum a čas ich vytvorenia, to ku ktorému poskytovateľovi patria a to aké služby sú poskytované v danej čakárni. Taktiež zaznamenáva informáciu o veľkosti časových rámcov pre danú čakáreň.

4.2.4 Otváracie hodiny

Táto entita reprezentuje otváracie hodiny pre dni, v ktorých je daná čakáreň otvorená. Obsahuje informácie o tom, ku ktorej čakárni otváracie hodiny patria, o aký deň sa jedná, čas otvorenia čakárne a čas zatvorenia čakárne.

4.2.5 Časové rámce

Účelom tejto entity je reprezentovať jednotlivé časové rámce. Tieto rámce sú vytvárané pre každú čakáreň samostatne podľa otváraciej doby a reprezentujú určitý časový úsek. Z týchto úsekov sú následne vyskladané jednotlivé termíny. O každom časovom rámci sú zaznamenávané informácie ako jeho dátum, začiatok rámcu, koniec rámcu a to či patrí k niektorému z termínov.

4.2.6 Termíny

Entita **termíny** predstavuje jednotlivé termíny/objednávky pre určitú čakáreň. Sú v nej zaznamenávané údaje o dátume termínu, o jeho začiatku, o službe, ktorá má byť na ňom vykonaná, o tom, ktorým užívateľom je termín obsadený a taktiež je pri každom termíne uložená informácia o jeho potvrdení zo strany užívateľa a poskytovateľa. Ďalšiu uchovávanú informáciu predstavuje kód potrebný pre autentifikáciu sa po príchode do čakárne.

4.2.7 Služby

Účelom tejto entity je reprezentovať jednotlivé služby poskytované v čakárňach. O jednotlivých službách zaznamenáva informácie o ich názve, dobe trvania, popise danej služby a dátume a čase vytvorenia služby.

4.2.8 Administrátori

Účelom entity **administrátori** je reprezentovať administrátorské účty pre správu poskytovateľov a čakární. Zaznamenáva informácie o mene a hesle administrátora.

4.3 Návrh uloženia dát

Entity navrhnuté v predchádzajúcej podkapitole je následne potrebné pretransformovať na databázové tabuľky, ktoré budú slúžiť ako úložiská dát potrebných pre správny beh aplikácie. Pre vytvorenie týchto tabuliek je potrebné vhodne zvoliť ich úložný systém. MySQL databáza nám ponúka k tomuto výberu viacero úložných systémov, avšak reálne používané a vhodné pre naše potreby sú len dva a to MyISAM a InnoDB.

4.3.1 MyISAM

MyISAM[11] je úložný systém založený na systéme ISAM (Indexed Sequential Access Method) a oproti svojmu predchodcovi obsahuje veľa užitočných rozšírení. V tomto úložnom systéme je tabuľka a dáta, ktoré obsahuje uložená pomocou troch súborov. Súbor s príponou **.frm** obsahuje formát tabuľky. Súbor obsahujúci dáta má príponu **.MYD** (MYData). Posledný zo súborov je indexový súbor s príponou **.MYI** (MYIndex). Do verzie 5.5 bol využívaný ako základné úložisko systému MySQL.

4.3.2 InnoDB

InnoDB[6] [10] je veľmi spoľahlivý a výkonný úložný systém pre MySQL. Je vytvorený tak aby dodržiaval ACID model. Tento model zahŕňa 4 vlastnosti ktoré musia byť dodržané, a to atomicitu (angl. atomicity, A), konzistenciu (angl. consistency, C) izolovanosť (angl. isolation, I) a trvanlivosť (angl. durability, D). InnoDB taktiež zahŕňa podporu pre cudzie kľúče. Na rozdiel od MyISAM ukladá tabuľky a indexy do tabuľkových priestorov (tablespaces), ktoré môžu pozostávať z niekoľkých súborov alebo aj zo samostatnej diskovej partície. Vďaka tejto zmene umožňuje aby boli tabuľky veľmi veľké aj na operačných systémoch, ktoré limitujú veľkosť súboru. Od verzie MySQL 5.5 je využívaný ako základný úložný systém.

Pri tomto úložnom systéme je taktiež možné zvýšiť výkon databáze nadstavením niektorých premenných. Medzi najhlavnejšie patrí `Innodb_buffer_pool_size` obsahujúca veľkosť buffera pre cache pre dáta a indexy tabuliek alebo `Innodb_log_file_size` obsahujúca veľkosť logovacieho súboru či `Innodb_file_per_table`, ktorá pri aktivácii začne ukladať každú tabuľku do samostatného súboru. Z týchto nadstavení sa pre potreby našej aplikácie javí ako opodstatnená iba možnosť využitia premennej `Innodb_file_per_table`, ktorej zapnutie je vhodné hlavne pri menších databázach s malým počtom tabuliek. Zvyšné možnosti by bolo vhodné použiť pri väčších databázach s veľkým obsahom dát.[3]

4.4 Návrh knižnice pre prácu s databázou

Nasledujúca podkapitola pojednáva o problematike návrhu knižnice pre prácu s databázou. Sú v nej uvedené požiadavky na funkcionality, ktorú musí poskytovať a taktiež obsahuje návrh jednotlivých funkcií, ktoré by mala následne implementovať.[1]

Táto knižnica má za úlohu vytvoriť rozhranie medzi serverovou časťou a databázovou časťou. Po tom ako server dostane požiadavku od klienta, ktorá zahŕňa prácu s databázou, využije jednu alebo viacero funkcií (záleží na povahe požiadavky) implementovaných v knižnici a následne vygeneruje odpoveď pre klienta.

V nasledujúcej sekcii sú uvedené základné funkcionality, ktoré musí knižnica poskytovať:

- vytvorenie/zrušenie užívateľského účtu
- zmena hesla pre užívateľský účet
- prihlásenie užívateľa do systému
- prihlásenie užívateľa na termín
- odhlásenie užívateľa z termínu
- potvrdenie termínu poskytovateľov/užívateľom
- vytvorenie/zrušenie poskytovateľa
- výpis údajov o poskytovateľovi
- úprava údajov o poskytovateľovi
- prehľad poskytovateľov
- vyhľadanie poskytovateľa

- vytvorenie/zrušenie čakárne
- prehľad čakární pre daného poskytovateľa
- prehľad termínov pre danú čakáreň
- vytvorenie/zrušenie služby
- prehľad služieb
- vytvorenie/zrušenie administrátorského účtu
- zmena hesla pre administrátorský účet
- prihlásenia administrátora do systému
- prehľad administrátorov

4.5 Návrh klienta pre správu poskytovateľov a čakární

Táto podkapitola je venovaná problematike návrhu webového klienta pre správu poskytovateľov a ich čakární. Hlavnou úlohou klienta by malo byť vytvorenie a základná konfigurácia poskytovateľov a taktiež možnosť vytvorenia a konfigurácie k nim príslušných čakární. Taktiež je potrebné aby umožňoval zobrazíť aktuálny stav a prípadné mazanie týchto čakární a poskytovateľov. Ďalej by mal umožňovať vytváranie, rušenie a prehľad poskytovaných služieb.

Práca s týmto klientom by mala byť umožnená iba administrátorom a preto vzniká potreba vytvorenia role administrátora. Túto rolu budú zastávať administrátorské účty, no tie je potrebné taktiež spravovať a tak sa nám prirodzene ponúka možnosť vytvorenia role hlavného administrátora. Jednotlivé prípady užitia pre tieto role sú uvedené v diagrame prípadov užitia na obrázku 4.2.

Následným problémom, ktorý je potrebné riešiť je vhodné zvolenie implementačných nástrojov pre klienta. Keďže bude klient vytvorený formou webovej stránky, je tento výber vcelku jednoznačný, a pre jeho vytvorenie budú použité bežné webové nástroje ako značkový jazyk HTML v kombinácii so skriptovacími jazykmi PHP a JavaScript. Ďalšiu podstatnou časťou pri návrhu klienta je jeho grafický dizajn. Ponúka sa nám možnosť použiť niektorý zo širokej palety pred vytvorených vzorov pre webové stránky (templates) alebo si vytvoríť vlastný za pomoci niektorej z CSS knižníc ako sú Pure, Skeleton, Bootstrap či mnohé ďalšie.



Obr. 4.2: Diagram prípadov užitia pre klienta

Kapitola 5

Implementácia

V nasledujúcej kapitole budú preberané zvolené riešenia a zdôvodnený ich výber. Detailnejšie sa pozrieme na to, akým spôsobom bola vytvorená databáza, ďalej si povieme niečo k implementácii PHP knižnice a k funkciám, ktoré poskytuje a nakoniec sa povenujeme problematike tvorby front-endu a back-endu klienta pre správu poskytovateľov a čakární.

5.1 Vytvorenie a konfigurácia databázy

V tejto podkapitole je obsiahnutá konečná schéma databázy, taktiež je v nej uvedený detailný popis jednotlivých tabuliek. Táto podkapitola taktiež zdôvodňuje prečo bol zvolený práve takýto koncept databázy.

Výsledná schéma zobrazená na obrázku 5.1 vychádza z ER diagramu databázy (obr. 4.2) uvedenom v návrhu. Oproti návrhu, obsahujú tabuľky niekoľko nových položiek. Potreba týchto položiek sa objavila až v procese implementácie a preto neboli v návrhu uvažované. Ako úložný systém pre tieto tabuľky bol zvolený systém InnoDB hlavne pre jeho podporu cudzích kľúčov a možnosť lepšieho ladenia databázy v prípade, že sa aplikácia rozšíri s čím súvisí aj nárast objemu jej dát. Uvedenú štruktúru databázy je možné vytvoriť za pomoci priloženého SQL skriptu čím je zaručená relatívne rýchla úvodná konfigurácia aplikácie.

V nasledujúcej časti sú detailne vysvetlené tabuľky a ich atribúty z obrázku 5.1.

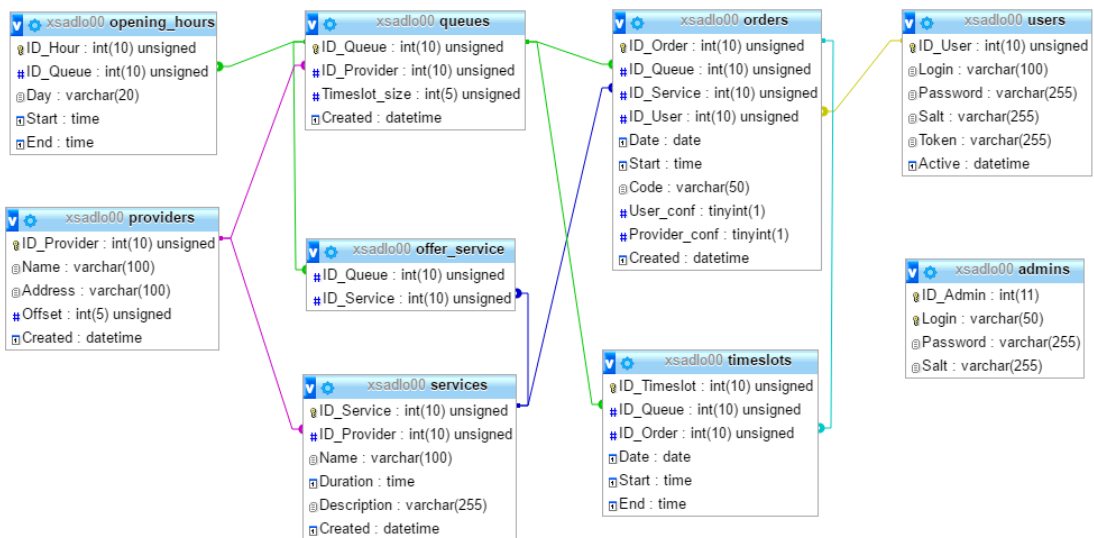
5.1.1 Tabuľka providers

Táto tabuľa reprezentuje v systéme poskytovateľa a uchováva dáta o ňom.

- ID_Provider - ide o primárny kľúč a identifikátor poskytovateľa.
- Name - uchováva informáciu o mene poskytovateľa.
- Address - uchováva informáciu o adrese poskytovateľa.
- Offset - určuje o aký čas (v minútach) poskytovateľ dovoľuje prípadný posun termínu.
- Created - uchováva informáciu o čase vytvorenia poskytovateľa.

5.1.2 Tabuľka queue

Tabuľka queues uchováva dáta o jednotlivých čakárňach.



Obr. 5.1: Schéma implementovanej databáze

- ID_Queue - predstavuje primárny kľúč a identifikátor jednotlivých čakární.
- ID_Provider - určuje, ktorému z poskytovateľov čakareň patrí a predstavuje vzťah 1:N medzi entitami poskytovateľa a čakárne.
- Timeslot_size - určuje veľkosť generovaných rámcov (timeslots) v minútach.
- Created - uchováva informáciu o dátume a čase vytvorenia čakárne.

5.1.3 Tabuľka opening_hours

Táto tabuľka uchováva informácie o otváracíj dobe pre jednotlivé čakárne.

- ID_Hour - predstavuje primárny kľúč a identifikátor otváracích hodín.
- ID_Queue - určuje, ku ktorej z čakární otváracie hodiny patria a taktiež predstavuje vzťah 1:N medzi entitami čakárne a otváracie hodiny.
- Day - určuje deň v týždni, pre ktorý sú dané otváracie hodiny určené
- Start - uchováva informáciu o začiatku otváracíj doby pre daný deň a čakareň.
- End - uchováva informáciu o konci otváracíj doby pre daný deň a čakareň.

5.1.4 Tabuľka services

V tejto tabuľke sú ukladané dáta o jednotlivých službách.

- ID_Service - predstavuje primárny kľúč a identifikátor jednotlivých služieb.
- ID_Provider - určuje ktorý z poskytovateľov ponúka danú službu a tým vytvára vzťah 1:N medzi entitami poskytovateľa a služby.

- Name - uchováva informáciu o názve služby
- Duration - predstavuje informáciu o tom ako dlho daná služba trvá.
- Description - uchováva detailný popis danej služby

5.1.5 Tabuľka `offer_service`

Táto tabuľka vznikla ako výsledok transformácie ER diagramu a predstavuje vzťah M:N medzi entitami čakárne a služby. Jej primárnou úlohou je uchovať informáciu o tom, ktoré služby sú poskytované jednotlivými čakárňami.

- ID_Queue - predstavuje identifikátor čakárne, ku ktorej je služba priradená.
- ID_Service - predstavuje identifikátor priradovanej služby.

5.1.6 Tabuľka `timeslots`

Účelom tabuľky timeslots je uchovávať informácie o jednotlivých časových rámcoch (timeslots). Tieto rámce predstavujú určitý časový úsek a sú využívané pri vytváraní termínu ako základný stavebný prvok.

- ID_Timeslot - predstavuje primárny kľúč a identifikátor jednotlivých rámcov.
- ID_Queue - udáva to, ku ktorej z čakární daný rámec patrí. Reprezentuje vzťah 1:N medzi entitami časové rámce a čakárne
- ID_Order - určuje, či daný rámec patrí k niektorému z termínov. V prípade NULL hodnoty ide o voľný rámec. Taktiež reprezentuje vzťah 1:N medzi entitami časové rámce a termíny.
- Date - toto pole uchováva informáciu o dátume, ktorý prísluší danému rámcu
- Start - určuje začiatok časového intervalu pre daný rámec.
- End - určuje koniec časového intervalu pre daný rámec.

5.1.7 Tabuľka `orders`

Tabuľka orders uchováva informácie o jednotlivých termínoch.

- ID_Order - predstavuje primárny kľúč a identifikátor termínu.
- ID_Queue - určuje ku ktorej z čakární daní termín patrí. Predstavuje vzťah 1:N medzi entitami termíny a čakárne.
- ID_Service - určuje ktorá zo služieb je na danom termíne poskytovaná. Taktiež predstavuje vzťah 1:N medzi entitami termíny a služby.
- ID_User - toto pole uchováva informáciu o tom, ktorý užívateľ je prihlásený na daný termín. Reprezentuje vzťah 1:N medzi entitami termíny a užívatelia.
- Date - uchováva informáciu o dátume vykonania termínu.

- Start - informuje o čase začiatku termínu
- Code - predstavuje identifikačný kód potrebný pre overenie pri príchode užívateľa do čakárne.
- User_conf - ide o booleovskú hodnotu určujúcu či bol termín potvrdený zo strany užívateľa.
- Provider_conf - ide o booleovskú hodnotu určujúcu či bol termín potvrdený zo strany poskytovateľa.
- Created - určuje dátum a čas vytvorenia termínu.

5.1.8 Tabuľka users

Táto tabuľka má za úlohu ukladať informácie o jednotlivých užívateľov. V momentálnej implementácii uchováva iba najpodstatnejšie informácie potrebné pre správny beh aplikácie.

- ID_User - predstavuje primárny kľúč a identifikátor užívateľa.
- Login - predstavuje prihlasovacie meno užívateľa.
- Password - uchováva informáciu o užívateľovom hesle vo forme hashu.
- Salt - predstavuje tzv. "sol" využívanú pre lepšiu bezpečnosť užívateľského účtu.
- Token - predstavuje náhodný reťazec využívaný pri autentifikácii užívateľa serverom po tom ako bol prihlásený.
- Active - ukladá informáciu vo forme dátumu a času poslednej aktivity užívateľa.

5.1.9 Tabuľka admins

Riadky tabuľky admins predstavujú jednotlivé administrátorské účty.

- ID_Admin - predstavuje primárny kľúč a identifikátor administrátora.
- Login - predstavuje administrátorské prihlasovacie meno.
- Password - uchováva informáciu o hesle k administrátorskému účtu vo forme hashu.
- Salt - predstavuje tzv. "sol", využívanú pre lepšiu bezpečnosť administrátorského účtu.

5.2 Implementácia PHP knižnice

Táto podkapitola je venovaná implementácii PHP knižnice pre prácu s databázou. Budú v nej zdôvodnené zvolené riešenia, jej väčšia časť je následne venovaná popisu jednotlivých funkcií poskytovaných knižnicou. Pri každej funkcii bude taktiež uvedený zoznam jej vstupov a výstupov.

Samotnú knižnicu predstavuje súbor `lib.php`. V tomto súbore je implementovaných 31 funkcií ktoré zaberajú 791 riadkov z čoho 601 riadkov zaberá kód, 83 riadkov predstavujú komentáre a zvyšok sú prázdne riadky vložené kvôli prehľadnosti a čitateľnosti kódu.

Vo výslednej implementácii je knižnicu možné používať ako PHP objekt, nad ktorým je možné volať rozličné metódy podľa potreby. Tieto metódy predstavujú jednotlivé funkcionality knižnice. Toto riešenie bolo zvolené pretože najlepšie popisuje modelovanú realitu, kde PHP objekt predstavuje SQL pripojenie k databáze a metódy tohto objektu predstavujú jednotlivé druhy požiadavok na databázu. Nasledujúca časť bude venovaná detailnejšiemu popisu jednotlivých funkcií.

5.2.1 Funkcia `__construct`

Ide o funkciu, ktorá je volaná pri vytváraní objektu knižnice, konštruktor. Pri jej zavolaní je vytvorené spojenie z databázou a následne uchované v objektovej premennej `$link`. Parametre potrebné k vytvoreniu spojenia sú pevne zadane v knižnici.

Hlavička funkcie:

```
__construct()
```

Návratová hodnota:

Bez návratovej hodnoty.

5.2.2 Funkcia `__destruct`

Jedná sa o deštruktor, ktorý pri rušení objektu ukončí mysql spojenie s databázou.

Hlavička funkcie:

```
__destruct()
```

Návratová hodnota:

Bez návratovej hodnoty.

5.2.3 Funkcia `create_queue`

Účelom tejto funkcie je vytvoriť novú čakáreň pre daného poskytovateľa.

Hlavička funkcie:

```
create_queue($id_provider, $timeslot_size)
```

- `$id_provider` - identifikátor poskytovateľa, pre ktorého je čakáreň vytváraná
- `$timeslot_size` - veľkosť časových rámcov generovaných pre danú čakáreň v minútach, v prípade zadania NULL hodnoty je nastavená základná hodnota 5

Návratová hodnota:

Identifikátor vytvorenej čakárne.

5.2.4 Funkcia `delete_queue`

Účelom tejto funkcie je zrušiť danú čakáreň.

Hlavička funkcie:

```
delete_queue($id_queue)
```

- `$id_queue` - identifikátor čakárne, ktorá má byť zrušená

Návratová hodnota:

Vracia TRUE pri úspešnom zmazaní, inak vracia FALSE.

5.2.5 Funkcia `set_hours`

Funkcia `set_hours` vytvára otváracie hodiny pre určitú čakáreň a deň v týždni.

Hlavička funkcie:

```
set_hours($id_queue, $day, $start, $end)
```

- `$id_queue` - identifikátor čakárne, pre ktorú sú otváracie hodiny nastavované
- `$day` - deň v týždni, pre ktorý sú otváracie hodiny určené (Monday, Tuesday...)
- `$start` - čas začiatku otváracej doby čakárne
- `$end` - čas konca otváracej doby

Návratová hodnota:

Identifikátor novovytvorených otváracích hodín.

5.2.6 Funkcia `get_hours`

Vracia informácie o otváraciej dobe pre určitú čakáreň a deň v týždni.

Hlavička funkcie:

```
get_hours($id_queue, $day)
```

- `$id_queue` - identifikátor čakárne, pre ktorú chceme zistiť otváracie hodiny
- `$day` - deň v týždni, pre ktorý chceme zistiť otváracie hodiny (Monday, Tuesday...)

Návratová hodnota:

V prípade úspechu vracia pole (`$hours`) s nasledujúcimi hodnotami:

- `$hours['ID_Hour']` - identifikátor otváracích hodín
- `$hours['Start']` - začiatok otváracej doby
- `$hours['End']` - koniec otváracej doby

5.2.7 Funkcia `create_service`

Táto funkcia má za úlohu vytvoriť novú službu patriacu zadanému poskytovateľovi.

Hlavička funkcie:

```
create_service($id_provider, $name, $duration, $description)
```

- `$id_provider` - identifikátor poskytovateľa, ktorý vytváranú službu ponúka
- `$name` - názov služby
- `$duration` - trvanie služby
- `$description` - slovný popis služby

Návratová hodnota:

Identifikátor novovytvorenej služby.

5.2.8 Funkcia `delete_service`

Funkcia pre zrušenie zadanej služby.

Hlavička funkcie:

```
delete_service($id_service)
```

- `$id_service` - identifikátor rušenej služby

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.9 Funkcia `set_service`

Táto funkcia priradí zadanú službu k zadanej čakárni.

Hlavička funkcie:

```
set_service($id_queue, $id_service)
```

- `$id_queue` - identifikátor čakárne, pre ktorú je priradovaná služba
- `$id_service` - identifikátor priradovanej služby

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.10 Funkcia `create_provider`

Funkcia slúžiaca k vytvoreniu nového poskytovateľa.

Hlavička funkcie:

```
create_provider($name, $address, $offset)
```

- `$name` - názov poskytovateľa
- `$address` - adresa poskytovateľa
- `$offset` - udáva čas v minútach, o ktorý poskytovateľ dovoľuje posunúť termín

Návratová hodnota:

Identifikátor novovytvoreného poskytovateľa.

5.2.11 Funkcia `update_provider`

Funkcia pre zmenu údajov o poskytovateľovi.

Hlavička funkcie:

```
update_provider($id_provider, $name, $address, $offset)
```

- `$id_provider` - identifikátor upravovaného poskytovateľa
- `$name` - názov poskytovateľa.
- `$address` - adresa poskytovateľa

- `$offset` - udáva čas v minútach, o ktorý poskytovateľ dovoľuje posunúť termín

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.12 Funkcia `delete_provider`

Funkcia pre zrušenie zadaného poskytovateľa. Spolu s poskytovateľom zruší všetky jeho čakárne aj s termínmi.

Hlavička funkcie:

`delete_provider($id_provider)`

- `$id_provider` - identifikátor rušeného poskytovateľa

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.13 Funkcia `provider_info`

Funkcia vracajúca informácie o zadanom poskytovateľovi.

Hlavička funkcie:

`provider_info($id_provider)`

- `$id_provider` - identifikátor poskytovateľa, o ktorom chceme zistiť informácie

Návratová hodnota:

V prípade úspechu vracia pole (`$info`) s nasledujúcimi položkami.

- `$info['ID_Provider']` - identifikátor poskytovateľa
- `$info['Name']` - názov poskytovateľa
- `$info['Address']` - adresa poskytovateľa
- `$info['Offset']` - maximálny poskytovateľom povolený posun termínu v minútach
- `$info['Created']` - čas vytvorenia poskytovateľa

5.2.14 Funkcia `provider_queues`

Funkcia slúži na získanie informácií o všetkých poskytovateľových čakárňach.

Hlavička funkcie:

`provider_queues($id_provider)`

- `$id_provider` - identifikátor poskytovateľa, o ktorého čakárňach chceme zistiť informácie

Návratová hodnota:

V prípade úspechu vracia dvojrozmerné pole (`$queues`), v ktorom prvý index určuje čakáreň a druhý predstavuje kľúč k nasledujúcim položkám.

- `$queues[$position]['ID_Queue']` - identifikátor čakárne
- `$queues[$position]['ID_Provider']` - identifikátor poskytovateľa vlastniaceho čakárne
- `$queues[$position]['Timeslot_size']` - veľkosť časového rámca (v min.) danej čakárne
- `$queues[$position]['Created']` - dátum a čas vytvorenia čakárne

5.2.15 Funkcia `provider_services`

Funkcia slúži na získanie informácií o všetkých poskytovateľových službách.

Hlavička funkcie:

`provider_services ($id_service)`

- `$id_provider` - identifikátor poskytovateľa, o ktorého službách chceme zistiť informácie

Návratová hodnota:

V prípade úspechu vracia dvojrozmerné pole (`$services`), v ktorom prvý index určuje službu a druhý predstavuje kľúč k nasledujúcim položkám.

- `$services[$position]['ID_Service']` - identifikátor služby
- `$services[$position]['ID_Provider']` - identifikátor poskytovateľa poskytujúceho službu
- `$services[$position]['Name']` - názov služby.
- `$services[$position]['Duration']` - trvanie služby vo formáte hh-mm-ss
- `$services[$position]['Description']` - textový popis služby
- `$services[$position]['Created']` - dátum a čas vytvorenia služby

5.2.16 Funkcia `find_provider`

Táto funkcia slúži na vyhľadanie jedného alebo viacerých poskytovateľov.

Hlavička funkcie:

`find_provider($name)`

- `$name` - názov alebo časť názvu vyhľadávaného poskytovateľa. V prípade prázdneho reťazca uvažuje všetkých poskytovateľov

Návratová hodnota:

V prípade úspechu vracia dvojrozmerné pole (`$providers`) kde prvý index určuje poskytovateľa a druhý predstavuje kľúč k nasledujúcim položkám.

- `$providers[$position]['ID_Provider']` - identifikátor poskytovateľa

- `$providers[$position]['Name']` - názov poskytovateľa
- `$providers[$position]['Address']` - adresa poskytovateľa
- `$providers[$position]['Offset']` - čas v minútach, o ktorý poskytovateľ dovoľuje posunúť termín
- `$providers[$position]['Created']` - dátum a čas vytvorenia poskytovateľa

5.2.17 Funkcia `queue_info`

Táto funkcia vracia informácie o zadanej čakárni.

Hlavička funkcie:

```
queue_info($id_queue)
```

- `$id_queue` - identifikátor čakárne, o ktorej chceme zistiť informácie

Návratová hodnota:

V prípade úspechu vracia pole (`$info`) s nasledujúcimi hodnotami:

- `$info['ID_Queue']` - identifikátor čakárne
- `$info['ID_Provider']` - identifikátor poskytovateľa, ktorému čakáreň patrí
- `$info['Timeslot_size']` - veľkosť časových rámcov pre danú čakáreň
- `$info['Created']` - čas vytvorenia čakárne

V prípade neúspechu vracia `FALSE`.

5.2.18 Funkcia `queue_state`

Táto funkcia vracia jednotlivé časové rámce pre zadanú čakáreň a dátum.

Hlavička funkcie:

```
queue_state($id_queue, $date)
```

- `$id_queue` - identifikátor čakárne, pre ktorú chceme získať časové rámce
- `$date` - dátum, pre ktorý chceme získať časové rámce

Návratová hodnota:

Funkcia vracia dvojrozmerné pole (`$slots`) v ktorom prvý index určuje službu a druhý predstavuje kľúč k nasledujúcim položkám.

- `$slots[$position]['id_timeslot']` - identifikátor časového rámca
- `$slots[$position]['id_queue']` - identifikátor čakárne, ku ktorej daný rámec patrí
- `$slots[$position]['id_order']` - identifikátor termínu/objednávky, ku ktorej daný rámec patrí. V prípade hodnoty `NULL` je rámec voľný
- `$slots[$position]['date']` - dátum časového rámca
- `$slots[$position]['start']` - čas začiatku časového rámca
- `$slots[$position]['end']` - čas konca časového rámca

5.2.19 Funkcia set_order

Funkcia pre vytvorenie nového termínu.

Hlavička funkcie:

```
set_order($id_user, $id_service, $id_timeslot)
```

- \$id_user - identifikátor užívateľa, ktorý vytvára termín
- \$id_service - identifikátor služby požadovanej pre daný termín
- \$id_timeslot - identifikátor prvého rámca patriaceho k objednávke. Predstavuje čas, kedy objednávka začína

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.20 Funkcia clear_order

Funkcia slúžiaca na zrušenie objednávky.

Hlavička funkcie:

```
clear_order($id_order)
```

- \$id_order - identifikátor rušenej objednávky

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.21 Funkcia user_conf_order

Funkcia súžiaca na potvrdenie objednávky zo strany užívateľa.

Hlavička funkcie:

```
user_conf_order($id_order)
```

- \$id_order - identifikátor potvrdzovanej objednávky

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vracia FALSE.

5.2.22 Funkcia provider_conf_order

Funkcia súžiaca na potvrdenie objednávky zo strany poskytovateľa.

Hlavička funkcie:

```
provider_conf_order($id_order)
```

- \$id_order - identifikátor potvrdzovanej objednávky

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vracia FALSE.

5.2.23 Funkcia register_admin

Táto funkcia slúži na vytvorenie nového administrátorského účtu.

Hlavička funkcie:

- \$name - prihlasovacie meno k vytváranému účtu
- \$passwd - heslo k vytváranému účtu

Návratová hodnota:

V prípade úspechu vracia identifikátor vytvoreného účtu. V prípade neúspechu vypíše chybové hlásenie.

5.2.24 Funkcia delete_admin

Funkcia slúžiaca na zrušenie administrátorského účtu.

Hlavička funkcie:

delete_admin(\$id_admin)

- \$id_admin - identifikátor rušeného účtu

Návratová hodnota:

V prípade úspechu vracia TRUE. V prípade neúspechu vypíše chybové hlásenie.

5.2.25 Funkcia login_admin

Funkcia slúžiaca na overenie prihlasovacích údajov k administrátorskému účtu.

Hlavička funkcie:

login_admin(\$name, \$passwd)

- \$name - prihlasovacie meno k overovanému účtu
- \$passwd - heslo k overovanému účtu

Návratová hodnota:

V prípade zhody údajov vracia TRUE, inak vracia FALSE.

5.2.26 Funkcia change_admin_passwd

Táto funkcia slúži na zmenu hesla k administrátorskému účtu.

Hlavička funkcie:

change_admin_passwd(\$name, \$old_passwd, \$new_passwd)

- \$name - prihlasovacie meno k účtu, pre ktorý chceme zmeniť heslo
- \$old_passwd - staré heslo k upravovanému účtu
- \$new_passwd - nové heslo k upravovanému účtu

Návratová hodnota:

V prípade úspešnej zmeny hesla vracia TRUE. V prípade neúspechu vracia FALSE.

5.2.27 Funkcia `all_admins`

Táto funkcia vracia údaje o všetkých administrátorských účtoch.

Hlavička funkcie:

```
all_admins()
```

Návratová hodnota:

Funkcia vracia dvojrozmerné pole (`$admins`) kde prvý index určuje administrátorský účet a druhý predstavuje kľúč k nasledujúcim položkám.

- `$admins[$position]['ID_Admin']` - identifikátor administrátorského účtu
- `$admins[$position]['Login']` - prihlasovacie meno k administrátorskému účtu

V prípade nulového počtu administrátorských účtov vracia `FALSE`.

5.2.28 Funkcia `register_user`

Táto funkcia slúži na vytvorenie nového užívateľského účtu.

Hlavička funkcie:

```
register_user($name, $passwd)
```

- `$name` - prihlasovacie meno k vytváranému účtu
- `$passwd` - heslo k vytváranému účtu

Návratová hodnota:

V prípade úspechu vracia identifikátor vytvoreného účtu. V prípade neúspechu vypíše chybové hlásenie.

5.2.29 Funkcia `delete_user`

Funkcia slúžiaca na zrušenie užívateľského účtu.

Hlavička funkcie:

```
delete_user($id_user)
```

- `$id_user` - identifikátor rušeného účtu

Návratová hodnota:

V prípade úspechu vracia `TRUE`. V prípade neúspechu vypíše chybové hlásenie.

5.2.30 Funkcia `login_user`

Funkcia slúžiaca na overenie prihlasovacích údajov k užívateľskému účtu.

Hlavička funkcie:

```
login_user($name, $passwd)
```

- `$name` - prihlasovacie meno k overovanému účtu
- `$passwd` - heslo k overovanému účtu

Návratová hodnota:

V prípade zhody údajov vracia `TRUE`, inak vracia `FALSE`.

5.2.31 Funkcia `auth_user`

Funkcia slúžiaca na získanie identifikátora užívateľského účtu, ku ktorému patrí zadaný identifikačný reťazec.

Hlavička funkcie:

```
auth_user($token)
```

- `$token` - náhodný reťazec využívaný pri komunikácii serveru a klienta pri prihlásení k užívateľskému účtu

Návratová hodnota:

V prípade úspechu vracia identifikátor užívateľského účtu, ktorému patrí zadaný token. V prípade neúspechu vracia `FALSE`.

5.3 Implementácia klienta pre správu poskytovateľov a čakární

Táto podkapitola je venovaná implementácii klienta pre správu poskytovateľov a čakární. Sú v nej uvedené príklady použitia jednotlivých funkcií z predchádzajúcej podkapitoly. Taktiež je v nej preberaná problematika grafického vzhľadu klienta a zdôvodnený výber konečných riešení s ňou spojených.

Klient je implementovaný vo forme webovej stránky. Toto riešenie bolo zvolené hlavne pre jeho dostupnosť a prenositeľnosť v prípade zmeny serveru. Klient by mal byť v praxi dostupný na doménovej adrese patriacej serveru aplikácie. Pre jeho inštaláciu stačí skopírovať zdrojové súbory z obrázkov 5.2 a 5.3 a správne nastaviť údaje o databáze v knižnici pre prácu s touto databázou.

Ako už bolo spomínané základným účelom klienta je vytvorenie a úvodná konfigurácia poskytovateľa. Z toho vyplýva skutočnosť, že jeho väčšia časť je tvorená stránkami obsahujúcimi formuláre pre zadávanie a prípadnú úpravu údajov. Jednotlivé stránky klienta taktiež obsahujú veľké množstvo tabuliek slúžiacich na zobrazovanie dát. Všetky spomínané prvky sú implementované ako HTML objekty, ktoré ukladajú alebo získavajú dáta za pomoci volania funkcií z knižnice pre prácu s databázou. Tieto volania sú umiestnené priamo v ich HTML kóde ako úseky PHP kódu.

Pre prácu s klientom je potrebné byť prihlásený pomocou administrátorského účtu. Ako základný je vytvorený administrátorský účet s prihlasovacím menom *admin* a heslom *admin*. Heslo k tomuto účtu je následne možné zmeniť pomocou klienta. Tento účet taktiež slúži na vytváranie a rušenie ostatných administrátorských účtov.

Obrázok 5.2 zobrazuje koreňový adresár klienta, kde súbor `lib.php` predstavuje PHP knižnicu pre prácu s databázou. `index.php` predstavuje hlavný súbor klienta ktorý spája jednotlivé časti do celku a taktiež k nim pripája grafický vzhľad uložený ako kaskádové štýly v súbore `bootstrap.css`.

Názov	Dátum úpravy	Typ	Veľkosť
podstranky	26.4.2016 14:17	Priečinok súborov	
bootstrap.css	5.4.2016 15:37	Súbor CSS	146 kB
index.php	5.4.2016 14:09	Súbor PHP	9 kB
lib.php	22.4.2016 15:04	Súbor PHP	30 kB

Obr. 5.2: Koreňový adresár obsahujúci súbory klienta.

Názov	Dátum úpravy	Typ	Veľkosť
add_admin.php	23.3.2016 15:39	Súbor PHP	3 kB
add_provider.php	5.4.2016 15:49	Súbor PHP	3 kB
add_queue.php	15.4.2016 11:35	Súbor PHP	18 kB
add_service.php	15.4.2016 11:33	Súbor PHP	4 kB
admins.php	23.3.2016 14:18	Súbor PHP	2 kB
change_admin_psswd.php	23.3.2016 15:58	Súbor PHP	3 kB
opening_hours.php	11.3.2016 16:38	Súbor PHP	4 kB
provider.php	12.4.2016 19:40	Súbor PHP	6 kB
select_provider.php	5.4.2016 14:18	Súbor PHP	2 kB
state_queue.php	2.4.2016 14:45	Súbor PHP	4 kB

Obr. 5.3: Adresár podstránky obsahujúci jednotlivé podstránky klienta.

5.3.1 Programová časť klienta

Základnými prvkami klienta sú jeho jednotlivé podstránky uložené v adresáry podstránky (obsah adresára podstránky je možné vidieť na obrázku 5.3). Každá z týchto podstránok zastáva jednu alebo viac funkcionalít klienta. Ako už bolo spomenuté, podstránky sú prepojené pomocou indexového súboru `index.php` ktorý vytvára hlavnú štruktúru stránky a rozhoduje o tom ktorá z podstránok má byť aktuálne zobrazená. Pre tento účel využíva metódu `GET` a štruktúru `switch` obsahujúcu kód jednotlivých podstránok (využíva `include`). Knižnica pre prácu s databázou je pripojená k indexovému súboru pomocou príkazu `require 'lib.php'`; kde `lib.php` predstavuje súbor obsahujúci kód knižnice. Spojenie s databázou je následne naviazané pomocou vytvorenia nového objektu knižnice pomocou príkazu `$connection = new Connection();`. Po tejto úvodnej konfigurácii je možné využívať funkcie knižnice ako volanie metód objektu predstavujúceho spojenie s databázou a to napríklad ako `$connection>login_admin($name, $psswd);`. Tento príklad simuluje prípad, kedy chceme overiť prihlasovacie meno (`$name`) a heslo (`$psswd`) administrátora, ktorý sa momentálne pokúša prihlásiť ku klientovi pre správu poskytovateľov a čakární.

Presný účel jednotlivých podstránok je vysvetlený v nasledujúcom zozname:

- `add_admin` - slúži na vytvorenie nového administrátorského účtu s využitím funkcie `register_admin`.

- `add_provider` - slúži na vytvorenie nového poskytovateľa za pomoci funkcie `create_provider`.
- `add_queue` - zabezpečuje vytvorenie novej čakárne za pomoci funkcie `create_queue`.
- `admins` - slúži ako prehľad administrátorských účtov, ktorý taktiež umožňuje ich mazanie. Z implementovanej knižnice využíva funkcie `all_admins` a `delete_admin`.
- `change_admin_psswd` - slúži na zmenu hesla k aktuálne prihlásenému administrátorskému účtu. Pre tento účel využíva rovnomennú funkciu `change_admin_psswd`.
- `opening_hours` - zobrazuje otváracie hodiny pre zvolenú čakáreň s využitím funkcie `get_hours`.
- `provider` - táto podstránka slúži na zobrazenie detailu poskytovateľa. Sú v nej zobrazované základné informácie o poskytovateľovi za pomoci funkcie `provider_info`. Tie je taktiež možné upraviť pomocou funkcie `update_provider`. Ďalej táto podstránka zobrazuje jednotlivé čakárne a služby patriace danému poskytovateľovi za pomoci funkcií `provider_queues` a `provider_services`. Taktiež umožňuje mazanie týchto čakární a služieb za pomoci funkcií `delete_queue` a `delete_service`.
- `select_provider` - zobrazuje poskytovateľov a vyhľadávanie medzi nimi, ktoré zabezpečuje funkcia `find_provider`. Taktiež umožňuje mazanie jednotlivých poskytovateľov za pomoci funkcie `delete_provider`.
- `state_queue` - zobrazuje stav danej čakárne pre zadaný dátum a to s využitím funkcie `queue_state`.

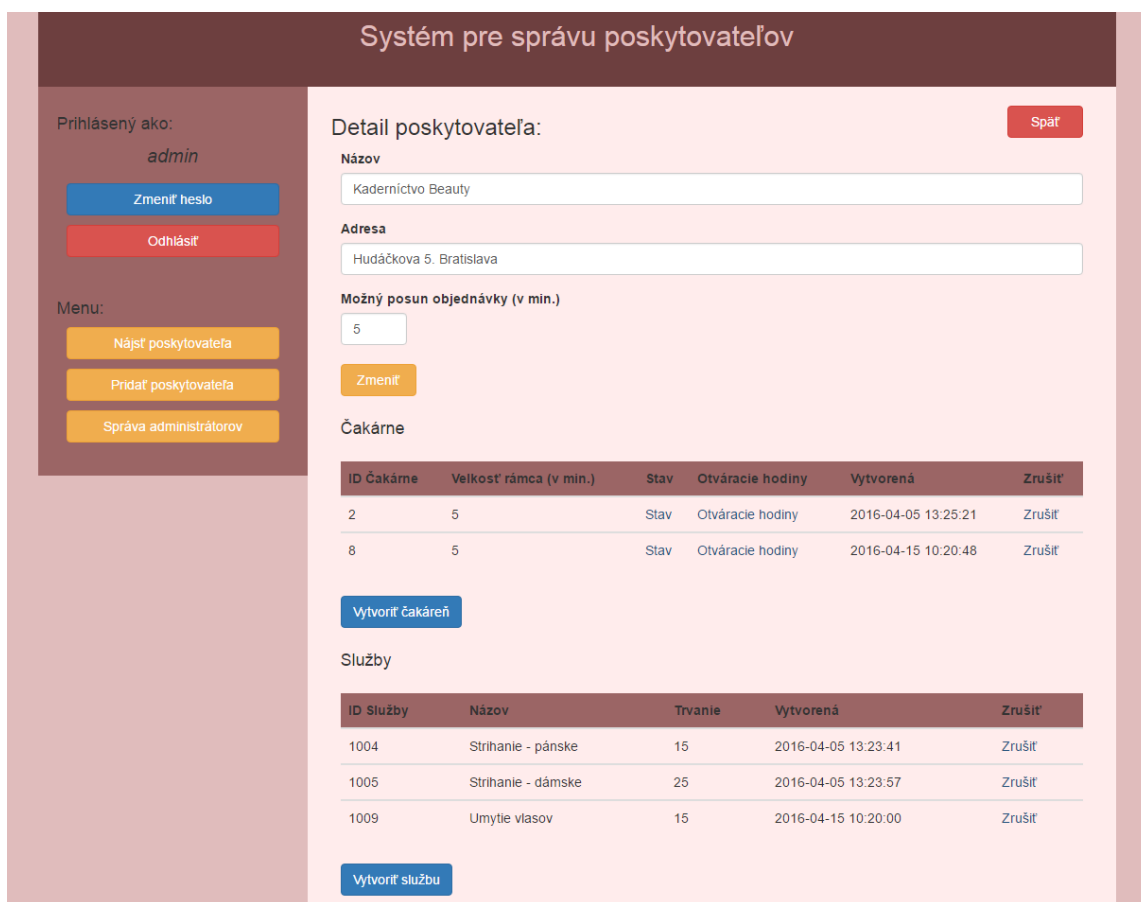
Zvyšné funkcie ktoré sú implementované v knižnici a zostávajú klientom nevyužité sú vytvorené pre potreby serveru a sú zväčša zamerané na overovanie či potvrdenie určitých údajov.

5.3.2 Grafická časť klienta

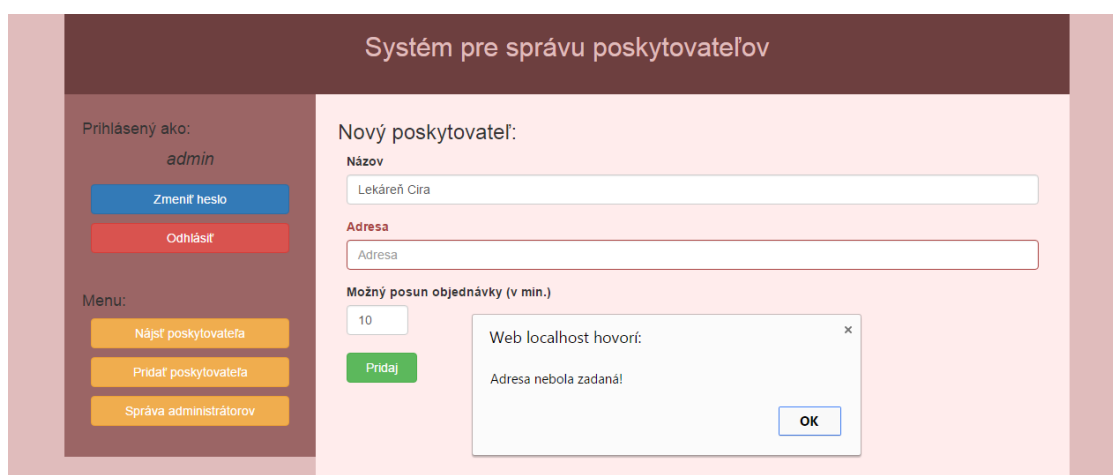
Ďalším problémom, ktorý musí klient riešiť je vizualizácia dát získaných pomocou funkcií knižnice. Tento problém bol riešený najmä s ohľadom na prehľadnosť a intuitívne ovládanie klienta. Pre štylizovanie celého klienta bola využitá CSS knižnica bootstrap do ktorej bolo pridaných niekoľko štýlov tak aby spĺňala niektoré špecifické potreby klienta. Na obrázku 5.4 je možné vidieť ukážku vzhľadu klienta. Jedná sa o detail poskytovateľa.

V konečnom riešení sú pre vizualizáciu dát využité štruktúry bežne zastávajúce tento účel a tými sú najmä tabuľky a formuláre. Klient musí taktiež zabezpečiť konzistenciu dát predávaných funkciám knižnice. Toto je zabezpečené kontrolou dát po odoslaní formulárov metódou POST. V prípade neplatnosti dát je potrebné uviesť o tom užívateľa. Toto je zabezpečené zvýraznením nesprávne vyplneného poľa a vyskakovacím oknom obsahujúcim informáciu o type chyby. Toto okno zabezpečuje funkcia `alert()` jazyka javascript. Ukážku je možné vidieť na obrázku 5.5.

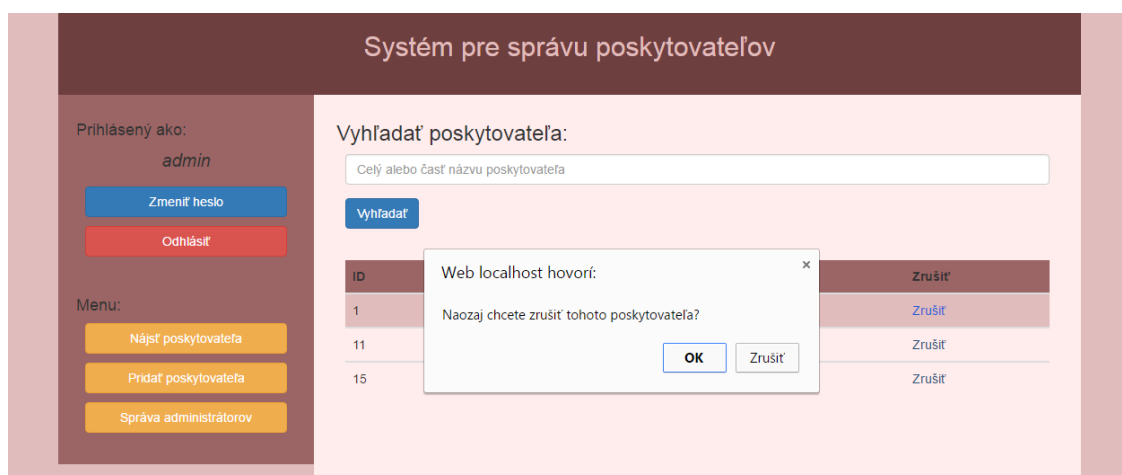
Ďalším vizuálnym prvkom je potvrdzovanie mazanie niektorej z predtým vytvorených položiek. Toto potvrdzovanie je potrebné hlavne pre prípady nechceného mazania napríklad prekliknutím sa. Tento problém je riešený za pomoci funkcie `confirm()` jazyka javascript. Ukážku je možné vidieť na obrázku 5.6.



Obr. 5.4: Grafický vzhľad klienta - detail poskytovateľa.



Obr. 5.5: Chovanie klienta v prípade zadania nesprávnych údajov.



Obr. 5.6: Chovanie klienta v prípade rušenia poskytovateľa.

Kapitola 6

Testovanie

Nasledujúca kapitola je venovaná testovaniu výslednej implementácie. Samotná databáza spolu s knižnicou bola testovaná za pomoci klienta. Toto testovanie prebiehalo formou simulácie práce s týmto klientom. Bol zvolený človek so základnou znalosťou práce so stolným počítačom ktorý následne prešiel a otestoval všetky funkcionality klienta. Pri tomto testovaní boli taktiež overené prípady zadania nesprávnych údajov. Funkcie knižnice nevyužité pri práci s klientom boli testované samostatným volaním v externom PHP skripte.

Aplikácia ako celok nebola doposiaľ otestovaná z dôvodu doterajšej nekompatibility jednotlivých častí. To bolo zapríčinené najmä rôznorodosťou riešení jednotlivých častí aplikácie a nedostatočnou komunikáciou v tíme vo fáze návrhu. Tento problém bol však už zväčša vyriešený a toto testovanie je plánované v blízkej dobe.

Kapitola 7

Záver

Cieľom tejto práce bolo navrhnuť a implementovať databázu, rozhranie pre prácu s databázou a klienta pre správu poskytovateľov a ich čakárni pre aplikáciu zvanú Virtuálna čakáreň. Účelom tejto aplikácie je poskytnúť užívateľom možnosť virtuálneho čakania v čakárni bez potreby osobnej prítomnosti a to za pomoci inteligentného telefónu alebo webového prehliadača.

V úvode práce boli spomenuté technológie použité pri tvorbe tejto práce. Jednalo sa hlavne o webové technológie ako HTML, PHP či MySQL, na ktorom je založená výsledná databáza. Nasledujúca kapitola bola venovaná návrhu. Pri návrhu databázy bola preberané jednotlivé entity systému, ktoré boli následne implementované ako tabuľky databázy. V tejto kapitole bol taktiež zdôvodnený výber PHP knižnice ako rozhrania pre prácu s databázou a tiež obsahuje zoznam funkcionalít, ktoré by mala knižnica poskytovať. Záver tejto kapitoly je venovaný návrhu klienta pre správu poskytovateľov a čakárni a taktiež je v nej uvedený diagram prípadov užitia zobrazujúci úlohy jednotlivých rolí pracujúcich s týmto klientom. Nadväzujúcou kapitolou k návrhu je kapitola o implementácii. Táto kapitola pojednáva o procese tvorby jednotlivých častí a ich výsledných riešeniach. Je v nich uvedená schéma implementovanej databázy s popisom jednotlivých tabuliek. Ďalej obsahuje popis jednotlivých funkcií, ktoré poskytuje PHP knižnica pre prácu s databázou a na záver je v nej uvedený účel jednotlivých častí klienta pre správu poskytovateľov a čakárni spolu s funkciami knižnice, ktoré využívajú.

Riešenie vytvorené v tejto práci predstavuje funkčnú verziu podporujúcu základné funkcionality požadované aplikáciou Virtuálna čakáreň. Táto aplikácia je stále vo vývoji a niektoré z najaktuálnejších zmien sa nestihli premietnuť do tejto práce. Databáza taktiež uchováva iba základné informácie o jednotlivých entitách vyskytujúcich sa v systéme. Niektoré z týchto a následne plánovaných zmien sú spomenuté v nasledujúcom odseku ako prípadné rozšírenia tejto práce.

Jedným z navrhovaných rozšírení spočíva v rozdelení poskytovateľa na viacero možných oddelení sídliacich na tom istom mieste. Pre toto rozšírenie by bolo potrebné pridať tabuľku pre tieto oddelenia, a teda pretvoriť celý návrh databázy a taktiež upraviť niektoré z funkcií knižnice. Ďalšie z prípadných rozšírení je pridanie kapacity pre jednotlivé termíny a ich pevné rozdelenie na väčšie časové úseky. Toto rozšírenie umožňuje flexibilnejšiu prácu s aplikáciou zo strany poskytovateľa a taktiež lepšie reaguje na neočakávané zmeny v termínoch. Pre tento účel by bola potrebná modifikácia tabuľky uchovávajúcej informácie o jednotlivých termínoch. Ďalším zaujímavým rozšírením je geolokácia poskytovateľov. Pre tento účel by bolo potrebné pridať informáciu o polohe poskytovateľov a taktiež pridať funkciu pre vyhľadanie poskytovateľov vzdialených určitéj vzdialenosti od zadaného bodu.

V praxi by tento bod predstavoval aktuálnu polohu užívateľa aplikácie a vzdialenosť od tohto bodu by vymedzovala poskytovateľov v užívateľovom okolí. Možných rozšírení aplikácie sa naskytuje oveľa viac a väčšina z nich by mala reflektovať požiadavky vyplývajúce z nasadenia aplikácie do praxe.

Výsledné riešenie bolo vytvorené na lokálnom hostingu avšak bolo implementované tak aby bol jeho prenos na prípadný webový server čo najjednoduchší a najpraktickejší. Pre správne fungovanie všetkých častí stačí vytvoriť databázu za pomoci priloženého skriptu, upraviť údaje o databáze, s ktorou má knižnica pracovať v zdrojovom súbore knižnice a v prípade, že chceme využívať klienta pre správu poskytovateľov a čakární, stačí skopírovať jeho zdrojové súbory na zvolený webový server spolu so zdrojovým súborom knižnice.

Literatúra

- [1] GUTMANS, A., BEKKEN, S. S. a RETHANS, D.: *Mistrovství v PHP5*. Brno: Computer Press, 2005, ISBN 80-251-0799-X.
- [2] KOFLER, Michael a Bernd ÖGGL: *PHP 5 a MySQL 5: průvodce webového*. Brno: Computer Press, 2007, ISBN 978-80-251-1813-9.
- [3] SCHNEIDER, R. D.: *MySQL Oficiální průvodce tvorbou, správou a laděním databází*. Grada, 2006, ISBN 80-247-1516-3.
- [4] Wikipedie: Otevřená encyklopedie: *Apache HTTP Server*. [Online; navštíveno 6.4.2016].
URL https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [5] Wikipedie: Otevřená encyklopedie: *HTML*. [Online; navštíveno 29.3.2016].
URL <https://en.wikipedia.org/wiki/HTML>
- [6] Wikipedie: Otevřená encyklopedie: *InnoDB*. [Online; navštíveno 8.4.2016].
URL <https://en.wikipedia.org/wiki/InnoDB>
- [7] Wikipedie: Otevřená encyklopedie: *MySQL*. [Online; navštíveno 6.4.2016].
URL <https://en.wikipedia.org/wiki/MySQL>
- [8] Wikipedie: Otevřená encyklopedie: *PHP*. [Online; navštíveno 6.4.2016].
URL <https://en.wikipedia.org/wiki/PHP>
- [9] Wikipedie: Otevřená encyklopedie: *XAMPP*. [Online; navštíveno 29.3.2016].
URL <https://en.wikipedia.org/wiki/XAMPP>
- [10] WWW stránky: *The InnoDB Storage Engine*. [Online; navštíveno 8.4.2016].
URL <https://dev.mysql.com/doc/refman/5.0/en/innodb-storage-engine.html>
- [11] WWW stránky: *The MyISAM Storage Engine*. [Online; navštíveno 8.4.2016].
URL <http://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html>

Prílohy

Zoznam príloh

A	Obsah CD	42
A.1	database.zip	42
A.2	lib.zip	42
A.3	client.zip	42

Príloha A

Obsah CD

A.1 database.zip

Archív obsahuje súbor `database.sql` predstavujúci skript pre vytvorenie databázy.

A.2 lib.zip

Archív obsahuje súbor `lib.php` predstavujúci knižnicu pre prácu s databázou.

A.3 client.zip

Archív obsahuje zdrojové súbory klienta pre správu poskytovateľov a čakární. Nachádzajú sa v ňom nasledujúce súbory:

- `index.php` - hlavný súbor spájajúci podstránky klienta.
- `bootstrap.css` - súbor obsahujúci grafický vzhľad klienta uložený ako kaskádové štýly.
- `podstranky` - adresár obsahujúci podstránky klienta.