

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE
A MATEMATIKY
ODDELENIE INFORMATIZÁCIE A RIADENIA PROCESOV**



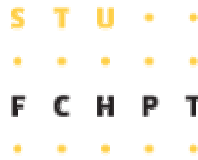
Systém na správu protetických výkonov

Diplomová práca

Vypracoval: Bc. Martin Zubček

Vedúci práce: Ing. Ľuboš Čirka, PhD.

BRATISLAVA 2009



ZADANIE DIPLOMOVEJ PRÁCE

Autor práce: Bc. Martin Zubček (22460)
Študijný program: chemické inžinierstvo a riadenie procesov
Zameranie: riadenie procesov
Študijný odbor: 5.2.17 chemické inžinierstvo

Vedúci práce: Ing. Ľuboš Čirka, PhD.

Názov témy: **Systém na správu protetických výkonov**

Rozsah práce: **40**

Špecifikácia zadania:

Cieľom práce je vytvoriť systém na správu protetických výkonov, ktorý môže byť prístupný na webovom systéme firmy. Tento systém bude slúžiť na spracovanie informácií z protetických štítkov za účelom vytvorenia faktúry za vykonané práce.

Dátum zadania diplomovej práce: **18. 02. 2009**

Termín odovzdania diplomovej práce: **22. 05. 2009**

Bc. Martin Zubček
študent

prof. Dr. Ing. Miroslav Fikar
vedúci pracoviska

prof. Ing. Vladimír Bálež, DrSc.
garant študijného programu

Čestné prehlásenie

Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne, podľa pokynov vedúceho práce a s použitím zdrojov uvedených v literatúre.

V Bratislave, 22. mája 2009

.....
podpis

Abstrakt

ZUBČEK, Martin: *Systém na správu protetických výkonov*. [Diplomová práce]. Slovenská technická univerzita v Bratislave. Fakulta chemickej a potravinárskej technológie; Oddelenie informatizácie a riadenia procesov. Školiteľ: Ing. Ľuboš Čirka, PhD. Bratislava: FCHPT STU, 2009. 49 s.

Práca sa zaoberá vytvorením systému na správu protetických výkonov, ktorý môže byť prístupný na webovom systéme firmy. Systém je potrebný na vytvorenie faktúry pre stomatologickú ambulanciu. Práca je členená na dve časti. Prvá časť je teoretická. Opisuje programovací jazyk PHP, databázový systém MySQL, značkový jazyk HTML a kaskádové štýly. Druhá časť opisuje vytvorenie systému a jeho použitie. Opisuje celý proces od vytvorenia databázy cez zápis a implementáciu základných funkcií až po generovanie faktúry vo formáte PDF.

Kľúčové slová

Databáza MySQL. Kaskádové štýly. Programovací jazyk PHP. FPDF. HTML formuláre.

Abstract

ZUBČEK, Martin: *Management system of prosthetic production*. [Diploma thesis]. Slovak University of Technology in Bratislava. Faculty of Chemical and Food Technology; Department of Information Engineering and Process Control. Consultant: Ing. Ľuboš Čirka, PhD. Bratislava: FCHFT SUT, 2009. 49 p.

This Diploma thesis deals with creation of Management system of prosthetic production which can be accessible from web system of a company. The system is needed for creating an invoice for dentists. This work is divided in two parts. First one is theoretical and explains programming language PHP, MySQL database system, Hyper Text Markup Language and Cascading Style Sheets. The second part describes creating of the system and gives directions for use. It contains the whole process which includes database creation, writing and implementation of functions and generating an invoice as a PDF document.

Keywords

Database MySQL. Cascading Style Sheets. Programming language PHP. FPDF. HTML forms.

Predhovor

Predmetom diplomovej práce je vytvorenie systému na správu protetických výkonov za účelom vytvorenia faktúry pre stomatologickú ambulanciu. Dôvod vzniku práce súvisí s vyplnením požiadaviek firmy J. Z. DENT s.r.o., ktorá potrebovala prejsť na nový systém, finančne menej náročný a zameraný hlavne na fakturáciu. Vzhľadom na to, že systém bol vytvorený podľa individuálnych požiadaviek firmy, je určený predovšetkým pre firmu J. Z. DENT s.r.o., čo ale nevylučuje jeho použitie aj pre iné podnikateľské subjekty s podobným zameraním.

Pri vytváraní systému bola nevyhnutná konzultácia s pracovníkmi firmy J. Z. DENT s.r.o., s pomocou ktorých som vytvoril štruktúru systému. Praktické skúsenosti a cenné rady Ing. Ľuboša Čirku, PhD. mi pomohli pri vypracovaní diplomovej práce. Touto cestou by som sa im chcel poďakovať.

Obsah

Zoznam použitých skratiek.....	ix
Zoznam obrázkov.....	x
Úvod	11
1 HTML	12
1.1 Štruktúra HTML stránky	12
1.2 Formuláre	12
1.3 Odkazy	13
2 CSS	14
2.1 Štruktúra CSS	14
3 MySQL	15
3.1 Pojmy relačnej databázy.....	15
3.2 Návrh databázy pre web.....	16
Práca s databázou	18
4 PHP	20
4.1 Štruktúra PHP	21
4.2 Objektovo orientovaný prístup	22
4.3 FPDF	24
4.4 Prístup k databáze MySQL pomocou PHP	25
5 Databáza a tabuľky v MySQL.....	27
5.1 Štruktúra databázy	27
5.2 Vytvorenie databázy a tabuliek	28
6 Grafický dizajn.....	30
6.1 Voľne stiahnuteľná šablóna.....	30
6.2 Aplikácia šablóny	31
7 Prihlasovanie užívateľov.....	32
7.1 Premenná <code>\$_SESSION</code>	32
7.2 Použitie premennej <code>\$_SESSION</code>	32
7.3 Registrácia nového užívateľa	33
8 Základné funkcie	35
8.1 Zobrazenie údajov.....	35
8.2 Upraviť	35
8.3 Vymazanie záznamu z tabuľky.....	36
8.4 Vytlačiť PDF	36
8.5 Vyhľadávanie v tabuľkách	37
8.6 Navigácia.....	38

9	Menu	39
9.1	Doktori	39
9.1.3	Prehľad doktorov.....	39
9.1.4	Nový doktor	40
9.2	Zákazky	40
9.2.3	Prehľad zákaziek	40
9.2.4	Nová zákazka	40
9.3	Cenník	42
9.3.3	Prehľad cenníka	42
9.3.4	Nová položka	42
9.4	Faktúry	43
9.4.3	Prehľad faktúr	43
9.4.4	Nová faktúra	44
9.5	O firme	44
9.5.3	Prehľad údajov	44
9.5.4	Úprava údajov	45
9.6	Odhlásenie	45
	Záver.....	47
	Zoznam bibliografických odkazov	48
	Prílohy	49

Zoznam použitých skratiek

MS DOS	Microsoft Disk Operating System
WWW	World Wide Web
HTML	Hypertext Markup Language
PHP	PHP: Hypertext Preprocessor
OOP	Objektovo orientované programovanie
GIF	Graphics Interchange Format
SQL	Structure Query Language
Perl	Practical Extraction and Report Language
ASP	Active Server Pages
JSP	Java Server Pages
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
FPDF	Free Portable Document Format
PDF	Portable Document Format
IČO	identifikačné číslo organizácie
DIČ	daňové identifikačné číslo

Zoznam obrázkov

Obr. 1	Vzťah medzi klientom a serverom.....	17
Obr. 2	Základná webová databázová architektúra.....	17
Obr. 3	Schémy dedičnosti	23
Obr. 4	Schéma prepojenia PHP so servermi a databázou	25
Obr. 5	Štruktúra databázy	27
Obr. 6	Databáza a tabuľky v phpMyAdmin.....	28
Obr. 7	Šablóna dizajnu.....	31
Obr. 8	Prihlasovací formulár	33
Obr. 9	Registračný formulár.....	33
Obr. 10	Hlavná časť faktúry.....	37
Obr. 11	Vyhľadávanie v tabuľke.....	37
Obr. 12	Navigácia systému	38
Obr. 13	Prehľad doktorov	39
Obr. 14	Prvý krok pridania novej zákazky	41
Obr. 15	Druhý krok pridania novej zákazky	41
Obr. 16	Nová položka cenníka	42
Obr. 17	Prehľad faktúr	43
Obr. 18	Prehľad údajov o firme.....	45

Úvod

Jedna z potrebných vecí na prevádzku zubnej techniky je software, ktorý je schopný spracovať údaje z protetických štítkov. Prvé programy tejto oblasti boli vytvorené pre operačný systém MS DOS. Vzhľadom na rozvoj informačných technológií je tento program v mnohých oblastiach nevyhovujúci. No aj napriek grafickým nedokonalostiam a celkovej spomalenosti systému bolo možné ešte donedávna program využívať a s jeho pomocou vytvárať faktúry. Problém nastal v súvislosti so zavedením EURO meny na Slovensku, čo spôsobilo, že program sa stal úplne nevyhovujúci. V súčasnosti sú na trhu viaceré spoločnosti ponúkajúce riešenia v podobe rôznych programov. Ceny takýchto programov sa pohybujú v cenovej hladine asi 500 - 1000 € a zahŕňajú v sebe veľa, z pohľadu menších firiem, zbytočných nástrojov. Z tohto dôvodu som bol oslovený firmou J. Z. DENT s.r.o. s požiadavkou vytvoriť systém podľa ich individuálnych potrieb.

Bližšia analýza problému ukázala, že firma potrebuje program primárne zameraný na fakturáciu. Vzhľadom na požiadavku firmy, ktorou bola možnosť pracovať so systémom z rôznych pracovísk, som sa rozhodol vytvoriť systém ako webovú aplikáciu naprogramovanú pomocou jazyka PHP s využitím MySQL databázy. Takto vytvorený systém je po umiestnení na webový server prístupný z rôznych pracovísk.

O programovacom jazyku PHP, ako o najpoužívanejšom jazyku na vytváranie webových aplikácií, je veľa dostupných informácií v tlačenej forme ako aj na webových stránkach. Teoretická časť mojej diplomovej práce sa opiera predovšetkým o poznatky získané z rôznych kníh zaoberajúcich sa rozvojom webových aplikácií. Takto získané teoretické poznatky sa stali základom pri vytváraní systému na správu protetických výkonov, ktorého vytvorenie bolo cieľom mojej diplomovej práce.

1 HTML

O rozvoj jazyka HTML sa postarala skupinka vedcov z Európskeho centra pre jadrový výskum vo Švajčiarsku, ktorá hľadala efektívny spôsob zdieľania dokumentov medzi sebou a ďalšími výskumnými centrami vo svete. Prišli s myšlienkou vytvoriť systém, ktorý by umožnil prezentovať rôzne dokumenty a bol by schopný tieto dokumenty logicky vzájomne pospájať. Výsledkom ich snahy bol jazyk HTML – hyper text markup language [1].

1.1 Štruktúra HTML stránky

Stránka napísaná pomocou HTML má svoju štruktúru, ktorá pozostáva z riadka obsahujúceho informácie o verzii HTML, z hlavičky ohraničenej pomocou párom značiek `<HEAD>` a `</HEAD>` a z tela stránky, v ktorom je hlavný obsah dokumentu. Telo stránky je implementované pomocou značiek `<BODY>`, `</BODY>` alebo pomocou elementu `FRAMESET`. Pomocou párových značiek `<HTML>`, `</HTML>` sa dokument začína a končí. Vo vnútri hlavičky sa pomocou značiek `<TITLE>` a `</TITLE>` nastaví text, ktorý bude vypísaný v lište okna prehliadača.

Základná štruktúra stránky vyzerá nasledovne:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>My first HTML document</TITLE>
  </HEAD>
  <BODY>
    Hello world!
  </BODY>
</HTML>
```

1.2 Formuláre

Formuláre slúžia na zadávanie užívateľských dát, ktoré sa majú spracovať serverom. V HTML dokumente je formulár ohraničený párovými značkami `<FORM>`, `</FORM>`. Značka otvárajúca formulár má rôzne atribúty. Medzi najhlavnejšie patria *action* a *method*. Pomocou prvého zo spomenutých atribútov určujeme cestu k súboru, ktorý bude posielané dáta spracovávať. Druhý z argumentov určuje aká metóda sa na odosielanie dát použije. K dispozícii sú hodnoty GET a POST.

GET je metóda základná, a teda v prípade nevyplnenia atribútu *method* je táto voľba prednastavená. Hodnoty premenných aj s ich názvami sú predávané súboru na spracovanie ako súčasť URL. Nachádzajú sa za otáznikom a sú viditeľné pre používateľa. Metóda GET sa používa hlavne pri navigácii a vyhľadávaní na stránkach.

Metódou POST sa dáta posielajú „neviditeľnou cestou“. Hodnoty formulára sa predávajú skriptu na spracovanie ako samostatný HTTP objekt. Tým, že posielané údaje nie sú súčasťou URL, je metóda POST vhodnejšia na posielanie dát z rozsiahlejších vstupov.

Najčastejšie používaná HTML značka vo formulároch je *input*, ktorá je nepárová. Jej hlavné atribúty sú *type*, *name* a *value*. Vlastnosť *type* môže nadobúdať hodnoty *text*, *password*, *hidden*, *radio*, *checkbox* a iné. Pomocou nich určíme o aký druh vstupu sa jedná. Hodnota odosielaná vstupom má názov premennej definovaný atribútom *name*, ktorého hodnota môže byť ľubovoľne zvolená. Prednastavená hodnota vstupu sa dá nastaviť pomocou vlastnosti *value*. Pri textovom vstupe sa prednastavená hodnota zobrazí ako text vypísaný v poli.

Pokiaľ chceme užívateľovi obmedziť možnosť vstupu je ideálnym riešením párová značka *select*, ktorá je ohraničená značkami `<SELECT>` a `</SELECT>`. Prvá z uvedených značiek má atribút *name*, ktorý určuje názov premennej. Vnútri ohraničenia spomenutej značky sa nachádzajú jednotlivé voľby, ktoré chceme užívateľovi ponúknuť. Každá voľba je vložená pomocou značiek `<OPTION>` a `</OPTION>` s atribútom *value*, ktorý určuje aká hodnota premennej sa odošle. Medzi párovými značkami každej voľby je vypísaný text, ktorý sa zobrazí užívateľovi na výber.

Špeciálnym typom pre HTML značku *input* je typ *submit*, ktorého voľbou vznikne vo formulári tlačidlo na odoslanie údajov vyplnených vo formulári skriptu na spracovanie. Hodnotou atribútu *value* sa nastaví nápis na tlačidle odoslania.

1.3 Odkazy

Na prechádzanie medzi stránkami slúžia odkazy. V dokumente HTML sú implementované párovými značkami `<a>`, ``. Medzi ich hlavné atribúty patrí *href* a *target*. Cesta k cieľu, na ktorý sa odkazuje, je vložená vo vlastnosti *href* a druhá vlastnosť určuje, v ktorom okne sa má cieľ odkazu otvoriť. Text nachádzajúci sa medzi značkami odkazu sa v dokumente zobrazí ako odkaz, na ktorý sa dá kliknúť.

2 CSS

CSS je skratkou anglického názvu Cascading Style Sheets, čo v preklade znamená kaskádové štýly. Kaskádové štýly predstavujú mechanizmus na pridanie štýlu do webových dokumentov. Hlavným dôvodom ich vzniku je obmedzená formátovacia schopnosť HTML. S niektorými objektmi sa pomocou HTML nedá robiť skoro nič. Naopak CSS dokáže upravovať každý prvok na stránke, čokoľvek môže získať rámček, farbu pozadia, špeciálne umiestnenie na stránke a iné.

Výhodou CSS je možnosť uloženia štýlu vo osobitnom súbore. Pri malej zmene štýlu stránky nie je potrebné zasahovať do kódu stránky, ale stačí upraviť súbor obsahujúci štýly. Oproti mnohým výhodám má CSS aj niektoré nevýhody. Základnou a najväčšou nevýhodou je rozdielna podpora webových prehliadačov. Všetky základné parametre sú podporované, ale každý prehliadač ich podporuje trochu inak.

2.1 Štruktúra CSS

Najlepšia metóda ako vložiť kaskádové štýly do stránky je pomocou HTML značky LINK. Takto vložíme súbor, ktorý v sebe obsahuje štýly, pomocou kódu:

```
<LINK REL="STYLESHEET" TYPE="text/css" HREF="style.css"
TITLE="Style" />
```

Základný syntax sa podľa [6] skladá z troch častí: selektor, vlastnosť a hodnota. Selektor môže byť HTML značka, ktorú si želáme upraviť alebo vopred zadefinovaná trieda, poprípade identifikátor. Vlastnosť určuje, čo konkrétne chceme na prvku zmeniť. Pri zmene viacerých vlastností sa jednotlivé vlastnosti oddeľujú bodkočiarkou. Napríklad pre HTML tag *p* s modrým písmom na zelenom pozadí bude mať deklarácia konkrétny tvar:

```
p {color: blue; background-color: green}
```

Každý HTML značke možno priradiť atribút *class*, ktorým zadefinujeme triedu pre štýl a pri deklarácii použijeme hodnotu zadaného atribútu *class* s bodkou na začiatku. V prípade zadefinovania identifikátore je na začiatku namiesto bodky mriežka.

3 MySQL

MySQL je relačný databázový systém, to znamená že jednotlivé údaje je možné uložiť na rôzne miesta s možnosťou opätovného prepojenia. Do databázy je možné uložiť prakticky hocijaký údaj. Príkazy MySQL možno zadávať priamo do príkazového riadku alebo prostredníctvom PHP. Hlavnými konkurentmi MySQL sú PostgreSQL, Microsoft SQL Server a Oracle. MySQL má mnoho predností, medzi ktoré podľa [3] patria:

- vysoká výkonnosť,
- nízke náklady,
- ľahká konfigurácia a výučba,
- prenositeľnosť,
- zdrojový kód je k dispozícii.

MySQL je veľmi rýchly. Mnohé testy ukázali, že dokáže bez problémov prekonať svojich konkurentov. K dispozícii je zadarmo pod licenciou Open Source alebo za veľmi nízku cenu, ak sa jedná o komerčnú licenciu. MySQL sa ľahšie konfiguruje oproti iným databázam a je použiteľný na viacerých operačných systémoch. Podobne ako u PHP aj tu sa dá získať a modifikovať zdrojový kód.

3.1 Pojmy relačnej databázy

Relačné databázy sú najbežnejším typom databáz. Ich základ tvorí relačná algebra. Ak chceme používať relačnú databázu, potrebujeme pochopiť základné databázové pojmy medzi ktoré patria: tabuľky, stĺpce, riadky, hodnoty, kľúče, schémy a relácie.

Tabuľka je presne to, čo vyjadruje toto slovo – je to nejaká tabuľka dát. Ak ste už niekedy používali tabuľkový kalkulátor, už ste používali relačné databázy.

Každý stĺpec tabuľky musí mať jedinečný názov a obsahuje nejaké údaje. Má pridelený typ údajov, ktoré obsahuje. Stĺpcom sa často hovorí pole a niekedy tiež atribúty.

Jednotlivé riadky tabuľky reprezentujú inú položku uloženú v tabuľke. Riadkom sa hovorí záznamy.

Každý riadok obsahuje množinu hodnôt, ktoré prislúchajú konkrétnym stĺpcom. Každá hodnota musí mať dátový typ špecifikovaný pre daný stĺpec.

Musí existovať spôsob, akým by sa dali identifikovať jednotlivé záznamy. Mal by to byť spôsob, pri ktorom vieme jednoznačne rozlíšiť každé dva záznamy od seba. Univerzálny spôsob identifikácie záznamov spočíva v pridelení jedinečného identifikačného čísla každému záznamu. Pre umelo vytvárané čísla v databáze je ľahké zabezpečiť ich jedinečnosť.

Identifikačný stĺpec tabuľky sa nazýva kľúč alebo primárny kľúč. Kľúč môže pozostávať aj z viacerých stĺpcov avšak ani kombináciou týchto stĺpcov nemusíme získať jedinečný kľúč.

Databáza sa väčšinou skladá z viacerých tabuliek, pričom kľúč jednej tabuľky odkazuje na druhú tabuľku. Termín, ktorý sa pre tento vzťah používa je nevlastný kľúč.

Návrhy všetkých tabuliek v databáze tvoria jej schému. Sú tu zaznamenané tabuľky, stĺpce a ich typy dát spolu s primárnymi a nevlastnými kľúčmi. Schémy môžu byť zaznačené pomocou diagramov, relačných diagramov entít alebo v textovej forme.

Relačné kľúče reprezentujú vzťah medzi údajmi dvoch tabuliek. V relačných databázach existujú tri druhy vzťahov. Klasifikujú sa podľa počtu „účastníkov“ na každej strane relácie. Môžu to byť relácie 1:1, 1:N alebo M:N pričom M a N sú čísla väčšie ako jedna. Prvé číslo pomeru určuje koľko riadkov sa má vybrať z prvej tabuľky a druhé číslo určuje koľko má byť vybraných riadkov z druhej tabuľky. Napríklad pre prípad 1:N platí, že k jednému riadku prvej tabuľky je prepojených N riadkov druhej tabuľky.

3.2 Návrh databázy pre web

Vedieť, kedy je potrebná nová tabuľka a čo v nej má byť kľúčom, je istým druhom umenia. Hlavne treba dodržať niekoľko základných princípov.

Keď vytvárame nejakú databázu väčšinou modelujeme prvky a vzťahy skutočného sveta a ukladáme informácie o týchto objektoch a vzťahoch. Vo všeobecnosti platí, že každý objekt reálneho sveta potrebuje v databáze vlastnú tabuľku.

Pri vyplňaní databázy musíme dať pozor, aby sme nezapisovali redundantné dáta. Správna voľba tabuliek tento problém vyrieši. V prípade nevyriešenia problému nastávajú dva závažné nedostatky. Prvým je plytvanie miesta. Druhým nedostatkom je možnosť vzniku anomálie pri aktualizácii. Môže sa narušiť integrita dát a potom by nebolo možné odlíšiť správne údaje od nesprávnych, čoho dôsledkom by bola strata informácií.

Musíme sa vyhnúť trom druhom anomálií a to: anomálii pri zmene, vložení a odstraňovaní. V prípade dobrého ošetrovania databázy pred ukladaním redundantných dát sa možno anomáliám bez problémov vyhnúť.

Každý stĺpec každého riadku obsahuje informáciu, ktorá je ďalej nedeliteľná. Získa sa tak tabuľka, ktorá má informácie o každom zázname rozdelené do viacerých stĺpcov, čoho dôsledkom je ľahšie pracovanie s jednotlivými informáciami daného záznamu.

Pri výbere kľúča sa treba vopred zamyslieť nad údajmi, ktoré sa budú ukladať do tabuľky. V niektorých prípadoch už samotné zapisované údaje obsahujú použiteľný kľúč. Pre iné prípady musíme vytvárať kľúč sami tak, aby bol jedinečný pre každý záznam. Už pri návrhu štruktúry databázy by sme mali vedieť aké požiadavky budú na databázu kladené. Tým zabezpečíme, že databáza bude obsahovať potrebné údaje a vzťahy medzi nimi.

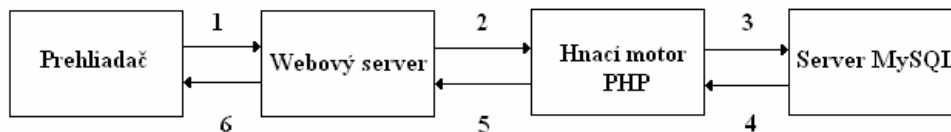
Nie je dobré mať v databáze veľa prázdnych polí. Na jednej strane je to plytvanie miestom, na strane druhej to môže spôsobovať problémy pri použití hodnoty tohto poľa na výpočty. Pokiaľ je hodnota uložená v databáze NULL nie je možné určiť irelevantnosť hodnoty alebo chybu databázy.

Webový server funguje tak, ako je znázornené na obr.1. Pozostáva z dvoch prvkov: webového prehliadača a webového serveru. Medzi nimi je obojsmerné spojenie. Prehliadač odošle požiadavku serveru a server mu spätne pošle odpoveď. Takáto architektúra je vyhovujúca pri doručovaní statických stránok. Architektúra, ktorá do procesu zapája aj databázu, je zložitejšia.



Obr. 1 Vzťah medzi klientom a serverom

Webová aplikácia, ktorú budem v tejto diplomovke budovať zodpovedá všeobecnej webovej databázovej štruktúre, ktorá je zakreslená na obr.2.



Obr. 2 Základná webová databázová architektúra

Práca s databázou

V MySQL sa často používajú príkazy ako CREATE, ALTER, SELECT, DELETE, DESCRIBE, INSERT, UPDATE, DROP.

Pred tým ako začneme s databázou pracovať musíme do nej vložiť nejaké dáta. Najbežnejšie sa údaje vkladajú pomocou SQL príkazu INSERT. Obvyklý zápis príkazu INSERT je:

```
INSERT [INTO] tabulka [(stlpec1, stlpec2, stlpec3,...)]  
VALUES (hodnota1, hodnota2, hodnota3,...);
```

Zástupné slovo *tabulka* sa nahradí skutočným názvom tabuľky, do ktorej chceme zapísať hodnoty. Všetky hodnoty obsahujúce reťazec by mali byť uzavreté v apostrofoch alebo úvodzovkách. Čísla a údaje vyjadrujúce dátum sa do úvodzoviek uzatvárať nemusia. V prípade zámerného vynechania údaje sa do daného poľa tabuľky zapíše hodnota NULL. Je možné pridávať aj viacero riadkov naraz. Každý riadok musí mať svoje vlastné zátvorky a musí byť oddelený čiarkou.

Najdôležitejším príkazom SQL je SELECT. Používa sa na získavanie dát z tabuliek databázy výberom riadkov odpovedajúcich špecifikovaným kritériám výberu. Je veľa ciest ako používať tento príkaz. Základný zápis príkazu SELECT je:

```
SELECT položky FROM tabuľky [WHERE podmienka]  
[GROUP BY typ_skupiny] [HAVING where_definícia]  
[ORDER BY typ_zoradenia] [LIMIT limitné_kritériá];
```

Za kľúčovým slovom SELECT vyberáme stĺpce tabuľky, ktoré chceme vybrať. V prípade výberu všetkých tabuliek nahradíme zástupné slovo *položky* znakom *. Špecifikáciu tabuľky alebo tabuliek, z ktorých chceme vyberať údaje, uskutočníme zámenou slova *tabuľky* za názvy požadovaných tabuliek oddelené čiarkou. Ak chceme pristupovať iba k určitej podmnožine riadkov tabuľky musíme špecifikovať *podmienku* vo voliteľnej klauzule WHERE.

Údaje získané pomocou príkazu SELECT môžeme pomocou jeho voliteľnej klauzuly ORDER BY zoradiť podľa stĺpca, ktorý definujeme zámenou za *typ_zoradenia*. Údaje môžu byť zoradené zostupne, pridaním kľúčového slova *desc*, alebo vzostupne, pridaním kľúčového slova *asc*.

Klauzula GROUP BY mení chovanie celej funkcie. Výsledné zobrazenie bude zoskupené podľa stĺpca uvedeného namiesto slova *typ_skupiny*. V súvislosti so zoskupovaním môžeme použiť aj klauzulu HAVING. Priamo sa spája s predchádzajúcou klauzulou a je niečo podobné ako WHERE, ale aplikuje sa iba na skupiny a celky.

Poslednou z klauzúl je LIMIT. Určuje, ktoré riadky z výsledku majú byť vypísané na výstup. Má dva parametre: číslo riadku, od ktorého sa má začať, a počet riadkov, ktoré sa majú vrátiť.

Na úpravu zapísaných údajov v databáze slúži príkaz UPDATE. Obvyklý zápis tohto príkazu je:

```
UPDATE nazov_tabulky
SET stĺpec1=výraz1,stĺpec2=výraz2,...
[WHERE podmienka] [LIMIT limitné_kritériá];
```

Základná myšlienka spočíva v aktualizácii príslušného stĺpca na zodpovedajúcu hodnotu výrazu. Príkaz možno obmedziť pomocou klauzúl WHERE a LIMIT podobne ako pri príkaze SELECT.

Záznamy sa dajú odstrániť z databázy veľmi jednoducho pomocou príkazu DELETE, ktorý vo všeobecnosti vyzerá nasledovne:

```
DELETE FROM nazov_tabulky
[WHERE podmienka] [LIMIT limitné_kritériá];
```

Pri používaní tohto príkazu treba byť opatrný. Neopatrným použitím sa môže stať, že sa vymažú všetky záznamy z tabuľky bez možnosti návratu.

Ak chceme odstrániť celú tabuľku alebo celú databázu, použijeme príkaz DROP TABLE alebo DROP DATABASE. Za názvom príkazu nasleduje názov tabuľky alebo databázy, ktorú chceme vymazať.

4 PHP

PHP je jeden z najobľúbenejších skriptovacích jazykov. Umožňuje dynamizáciu webového serveru. PHP je skratka anglického názvu „*PHP: Hypertext Preprocessor*“. Dôvodom rozšírenosti PHP medzi programátormi je jeho ľahká osvojiteľnosť, a to hlavne u programátorov, ktorí už majú skúsenosti s jazykom C, Java alebo Perl. Kód v tomto jazyku je možné nájsť na približne 16 miliónoch webových serveroch [2].

Hlavní konkurenti PHP sú Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) a Allaire ColdFusion. V porovnaní s týmito produktmi má PHP podľa [3] nasledujúce výhody:

- vysoká výkonnosť,
- rozhranie pre mnoho druhov databázových systémov,
- zabudovanie knižnice pre implementáciu bežných webových úloh,
- nízke náklady,
- ľahká výučba a použitie,
- prenositeľnosť,
- zdrojový kód PHP je k dispozícii.

PHP je veľmi efektívny, pričom jeden jednoduchý a nenáročný server je schopný obslúžiť milióny požiadaviek denne. Má schopnosť samostatne sa pripojiť k mnohým databázovým systémom. Od začiatku bol jazyk PHP navrhovaný pre webové aplikácie, a preto obsahuje veľa funkcií určených pre prácu na webe. Iba pár riadkov kódu stačí na vygenerovanie obrázkov vo formáte GIF, pripojenie sa k rôznym sieťovým službám, odosielanie e-mailov, generovanie dokumentov vo formáte PDF atď. Najnovšia verzia PHP je voľne stiahnuteľná kedykoľvek a zadarmo z internetu. Jeho syntax je založená predovšetkým na programovacích jazykoch C a Perl. Pokiaľ už jeden z týchto jazykov ovládate je možné písať kód v PHP prakticky automaticky. Tento programovací jazyk je k dispozícii pre mnoho operačných systémov. Vytvorený kód bude vo väčšine prípadov fungovať bez akýchkoľvek modifikácií. Na rozdiel od komerčných produktov je zdrojový kód PHP voľne prístupný. Netreba sa teda báť, že výrobca skrachuje alebo že sa rozhodne produkt nepodporovať. Môžete si v ňom čokoľvek upraviť alebo pridať.

4.1 Štruktúra PHP

PHP sa v dokumente HTML oddeľuje značkami `<?php` a `?>`. Vo vnútri týchto značiek sa nachádza obsah, ktorý je písaný v jazyku PHP. Jednoduchý kód, ktorý vypíše na stránke nápis „Ahoj svet!“ vyzerá takto:

```
<?php
    echo 'Ahoj svet!';
?>
```

Ako je vidieť na príklade je bodkočiarka za príkazom *echo*, ktorý pošle výraz v ňom obsiahnutý prehliadaču. Každý príkaz je v jazyku PHP oddelený bodkočiarkou takisto ako sa v slovenčine používa bodka na oddeľovanie viet. Vynechanie bodkočiarky je veľmi častá ale ľahko odstrániteľná syntaktická chyba.

Na komentovanie pasáže kódu sa používajú komentáre. Vysvetľujú účel skriptu tak, aby mu ostatní ľahšie porozumeli. PHP neinterpretuje žiadny komentár, preskočí ho ako keby sa jednalo o prázdny znak. Sú podporované rovnaké štýly komentárov ako majú C, C++ a skripty shellu. Viacriadkový komentár býva najčastejšie na začiatku skriptu pričom sa začína `/*` a končí `*/`. Pre jednoriadkový komentár sa použijú znaky `//`, pri ktorých sa všetky znaky za nimi až po koniec riadku pokladajú za komentár.

Hlavným zmyslom kódu je dynamické vytváranie obsahu stránok. PHP má rozsiahlu knižnicu funkcií, ktoré je možné použiť pri vytváraní webovej aplikácie. Pomocou PHP je veľmi ľahké získať informácie z formulára, ktorý užívateľ odoslal. V skripte môžeme získať obsah každého poľa v podobe premennej PHP. Názvy premenných sú jasne rozpoznateľné – začínajú znakom dolár (\$). Aby sme získali hodnoty z formulára je potrebné vedieť akou metódou bol formulár odoslaný. Vzhľadom sa metódu sa údaje získajú pomocou asociatívneho poľa `$_GET[]` alebo `$_POST[]`. Tieto polia majú ako indexy zvolené hodnoty atribútov *name* vstupov z formulárov. Premennú , ktorá sa formulárom poslala metódou GET vo vstupe s názvom *meno*, v PHP zavoláme ako prvok asociatívneho poľa `$_GET['meno']`.

Riadiace štruktúry umožňujú riadiť chod programu alebo skriptu. Môžeme ich zoskupovať do štruktúr pre vetvenie programu a do štruktúr pre opakované vykonávanie alebo cyklov. Základné rozhodovanie podľa podmienok je reprezentované príkazmi *if*, *elseif* a *else*. Neexistuje nič čo by sa nedalo urobiť pomocou jedného alebo viacerých príkazov *if* a *elseif*.

Ak je potrebné vykonať časť kódu niekoľkokrát za sebou, môžeme použiť cyklus a tieto časti opakovať. Túto úlohu spĺňajú cykly *while*, *for* a *foreach*. Cyklus *while* je pomerne bežný. Používa sa hlavne tam, kde nie je známy počet opakovaní cyklu. Pokiaľ je počet opakovaní cyklu známy môžeme použiť cyklus *for*. Základná štruktúra tohto cyklu vyzerá nasledovne:

```
for (výraz1; podmienka; výraz2)
    výraz3;
```

- *výraz1* sa vykoná iba na začiatku.
- *podmienka* sa overuje pred každou iteráciou
- *výraz2* sa vykoná na konci každej iterácie
- *výraz3* sa vykoná jedenkrát za iteráciu

4.2 Objektovo orientovaný prístup

Moderné programovacie jazyky obvykle pri vývoji podporujú alebo vyžadujú objektovo orientovaný prístup. Objektovo orientované programovanie (OOP) sa snaží k uľahčeniu vývoja programov použiť vzájomné vzťahy a vlastnosti objektov, ktoré sú pri vývoji programov zaradené do určitého systému (Pozri [3], s. 183.).

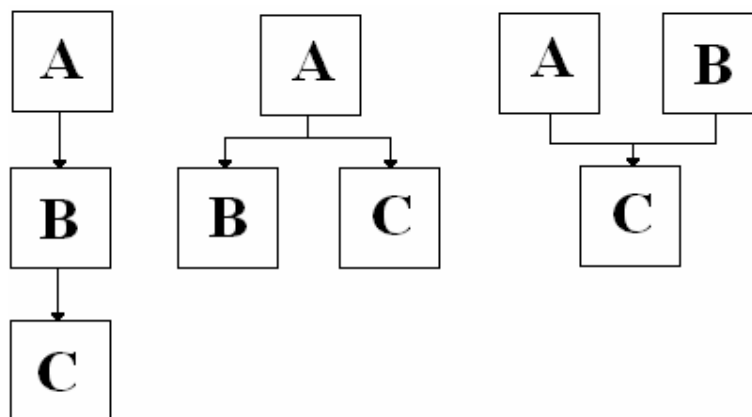
Princíp objektovo orientovaného programovania spočíva v tom, že bežne používané funkcie nie sú súčasťou vkladaného skriptu ale do *triedy*. Trieda je kolekciou premenných (členy triedy) a funkcií (metódy triedy), ktoré sú používané až vtedy, keď je volaná konkrétna *inštancia* danej triedy. Objekt je výsledkom toho, čo vznikne po vytvorení *inštancie* triedy.

Objektové programovanie má niekoľko výhod. Ľahko spája informácie dohromady a pracuje so všetkými dátami ako s jedným celkom. Dáta je možné spracovávať bez obmedzenia a obáv, že nové premenné prepíšu v pamäti premenné staršieho dáta. V jednom okamihu môžeme pracovať s viacerými inštanciami triedy bez toho, aby sa pritom obsah premenných prepísal alebo poškodil.

Objektovo orientovaný programovací jazyk podporuje polymorfizmus. To znamená, že rovnaké operácie rôznych tried sa odlišne správajú. Polymorfizmus je skôr charakteristikou správania než objektov. V PHP môžu byť polymorfické len členské funkcie triedy.

Na vytváranie hierarchických vzťahov medzi triedami a podtriedami slúži dedenie. Podtrieda dedí atribúty a operácie od svojej nadradenej triedy. S použitím dedičnosti

môžeme stavať na už existujúcich triedach a ďalej ich rozvíjať. Z jednoduchšej základnej triedy je možné odvodzovať zložitejšie a úzko zamerané triedy. Tým sa dostávame k jednej z najväčších výhod – opätovné využitie vytvoreného kódu. Dedenie šetrí čas, pretože niektoré operácie môžeme definovať v materskej nadradenej triede, a teda ju netreba definovať v každej ďalšej podtriede. PHP nepodporuje mnohonásobnú dedičnosť. Každá trieda môže dediť iba od jedného predka, ale množstvo potomkov nie je obmedzené. Na obr.3 sú tri rôzne spôsoby ako môžu byť triedy A, B, C ovplyvnené dedičnosťou.



Obr. 3 Schémy dedičnosti

Ľavá a prostredná kombinácia zobrazujú dedenie od jedného rodiča. Ľavá kombinácia znázorňuje triedu C, ktorá dedí z triedy B, ktorá dedí z triedy A. Každá trieda má jedného predka, a teda sa jedná o platné dedenie v PHP. Prostredná kombinácia ukazuje triedy B a C, ktoré sú obe potomkami triedy A. Opäť má každá z tried jedného predka, a preto sa jedná o opäť o platný príklad dedenia. Pravá kombinácia zobrazuje triedu C ako dediča z tried A i B. V tomto prípade má C dvoch predkov, preto sa jedná o prípad mnohonásobnej dedičnosti, čo v PHP nie je platná konštrukcia.

Vlastná trieda, ktorú vytvoríme, potrebuje názov. Vlastná trieda sa deklaruje takto `class meno_triedy{}`. Obsah triedy je vložený medzi zloženými zátvorkami. Každá trieda potrebuje nejaké atribúty. Prvky, ktoré by sa mohli meniť. Atribút je premenná vo vnútri vytvorenej triedy a je deklarovaný pomocou `var $meno_atributu`. Aby vytvorená funkcia mala nejakú funkčnosť je potrebné do nej pridať nejaké operácie vo forme funkcií, a teda metódy danej triedy.

Každá funkcia by mala vykonávať jednu, presne definovanú úlohu. Čím jednoduchšia úloha, tým jednoduchšie je funkciu zapísať a otestovať. Každá operácia je definovaná ako funkcia vo vnútri triedy:

```
function meno_funkcie($atribut1, $atribut2)
{
    obsah funkcie
}
```

4.3 FPDF

FPDF je PHP trieda, ktorá umožňuje vytvorenie dokumentu vo formáte PDF čisto len s využitím PHP bez potreby knižnice PDFlib. Táto trieda je voľne sťahuteľná z internetu a môže byť použitá bez obmedzenia alebo upravená aby spĺňala individuálne požiadavky. Podrobný opis triedy FPDF je prístupný na jej domovskej stránke <http://www.fpdf.org/>. Na tejto stránke sa nachádza podrobný manuál ku všetkým metódam tejto triedy ako aj vzorové kódy vytvorené pomocou tejto triedy.

V skripte, ktorý má vytvoriť PDF dokument pomocou triedy FPDF, nesmie byť žiadny iný výstup okrem generovaného dokumentu. Keďže sa jedná o objektové programovanie, tak sa na začiatku vytvorí nová inštancia triedy FPDF. Medzi základné funkcie, ktoré sú potrebné na vytvorenie dokumentu sú *AddPage()*, *SetFont()*, *Cell()* a *Output()*. Na vytvorenie jednoduchého PDF dokumentu, ktorý bude obsahovať vetu: „Ahoj svet!“ bude stačiť kód:

```
<?php
require('fpdf.php');
$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Ahoj svet!');
$pdf->Output();
?>
```

Na začiatku generovania vytvorím novú inštanciu triedy FPDF s názvom *\$pdf*. Pomocou funkcie *AddPage()* pridám v dokumente novú stránku. Funkcia *SetFont()* nastaví font, ktorý sa v dokumente použije na hrubo vytlačený Arial s veľkosťou písma 16. Volaním metódy *Cell()* triedy FPDF získam v dokumente pole, ktorého šírka bude

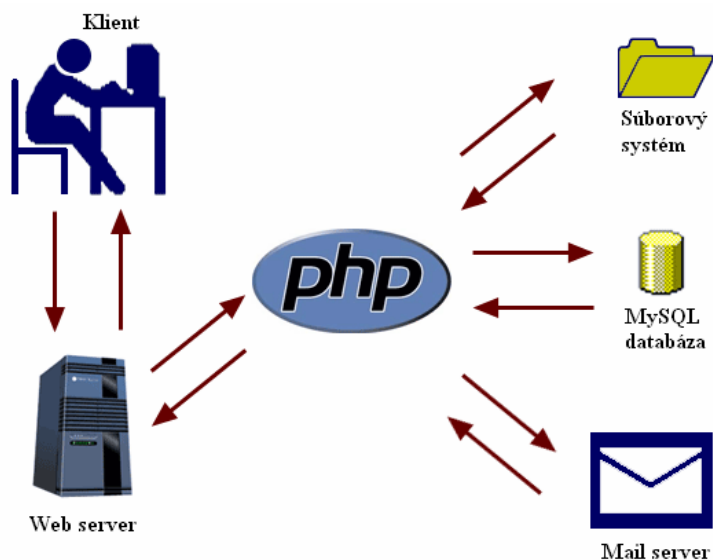
40 mm a výška bude 10 mm a vnútri tohto políčka bude nápis „Ahoj Svet!“. Na záver sa vygenerovaný dokument zobrazí na stránke pomocou funkcie *Output()*.

4.4 Prístup k databáze MySQL pomocou PHP

Pre lepšie pochopenie prirovnajme prácu PHP s databázou MySQL k prvotriednej reštaurácii. Predstavte si, že Váš dynamický web je prvotriedna reštaurácia, do ktorej dennodenne prichádzajú hostia s rôznymi požiadavkami. Hostí nezaujíma proces prípravy jedál, ale zaujíma ich výsledný produkt. V každej dobrej reštaurácii dochádza ku interakcii medzi hosťom a čašníkom, cieľom ktorej je prijatie individuálnych potrieb stravníkov. V tomto scenári možno charakterizovať tri hlavné zložky:

- Web server: Je šéfkuchár v reštaurácii. Ochotne a bez problémov pripraví čokoľvek, o čo hosť požiada. Rýchlosť flexibilita a schopnosť pripraviť mnoho rôznych typov jedál patria medzi jeho prednosti.
- PHP: To je čašník, ktorý zhromažďuje objednávky a predkladá ich kuchárovi. K požiadavkám pripája zvláštne inštrukcie, týkajúce sa spôsobu prípravy jedál.
- MySQL: Predstavuje sklad, v ktorom sa skladujú suroviny a prísady

Hosť sa po príchode do reštaurácie usadí a objedná si jedlo. Čašník (PHP) objednávku prijme a prednesie ju šéfkuchárovi (web server). Šéfkuchár najprv získa do skladu (MySQL), tam vyberie všetky potrebné ingrediencie na prípravu jedla a pripravené jedlo prestrie pred hosťom presne tak, ako si to hosť prial [2].



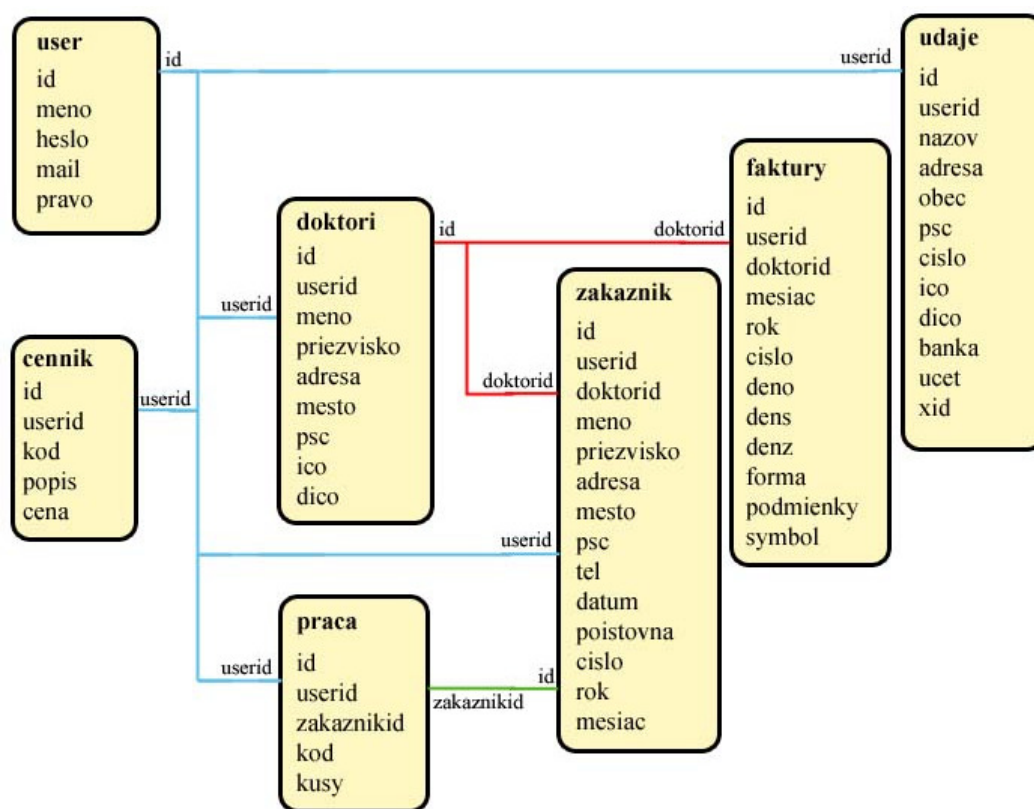
Obr. 4 Schéma prepojenia PHP so servermi a databázou

Na začiatku prístupu k databáze je pripojenie k MySQL serveru pomocou funkcie *mysql_pconnect(server,užívateľ,heslo)*. Následne po pripojení vyberieme databázu, ktorú chceme používať funkciou *mysql_select_db(názov_databázy)*. Na vykonanie SQL príkazov slúži funkcia *mysql_query(výraz)*, kde vo výraze je vložený SQL príkaz. Táto funkcia nám vráti identifikátor výsledku, ktorý je možné uložiť do premennej a spracovať. Existuje mnoho funkcií na spracovanie výsledkov pomocou identifikátora výsledku, ktorý je kľúčom k prístupu k riadkom vráteným od zadaného SQL príkazu. Zistenie počtu riadkov výsledku je prístupné pomocou funkcie *mysql_num_rows()*, ktorej môže byť predaný identifikátor výsledku. Viacero funkcií získava údaje z výsledku SQL príkazu, a to do poľa asociatívneho, indexovaného alebo do objektu. Prácu s databázou je možné ukončiť buď ukončením skriptu alebo použitím funkcie *mysql_close()*.

5 Databáza a tabuľky v MySQL

5.1 Štruktúra databázy

Pri realizácii praktickej časti tejto diplomovej práce bolo nutné vytvoriť databázu, v ktorej sa budú ukladať užívateľské dáta. Vytvoreniu databázy predchádzala štúdia, ktorej cieľom bolo zistiť, aké údaje potrebuje zubná technika, aby mohla byť na konci mesiaca vytvorená faktúra. Na základe informácii, ktoré mi v tomto smere poskytla zubná technika, som vytvoril jednu databázu a sedem tabuliek. Podrobná štruktúra databázy je na obr.5.



Obr. 5 Štruktúra databázy

Pri analýze štruktúry databázy sa objavili nasledujúce dva problémy. Prvý problém sa vyskytol pri zoradovaní vybraných údajov z tabuliek podľa abecedy. Slová, ktorých prvé písmeno bolo s diakritikou, sa umiestňovali na začiatok alebo koniec zoznamu, teda mimo abecedné poradie. Z tohto dôvodu som zvolil zotriedenie, čím sa rozumie

kódovanie údajov na *utf_slovak_ci*. Po tomto kroku sa problém vyriešil a slová s diakritikou na začiatku boli umiestňované v abecednom poradí.

Druhý problém sa objavil v súvislosti so zápisom identifikačného čísla prvého zápisu do tabuľky v stĺpci s vlastnosťou *auto_increment*. Prvý zapisovaný údaj získal hodnotu 0, ktorá je však ako hodnota v PHP signálom prázdnej premennej. Problém som odstránil nastavením daného stĺpca s identifikačným číslom v tabuľke na *PRIMARY KEY*. Prvému zapísanému údajovi tak bola miesto 0 priradená hodnota 1.

5.2 Vytvorenie databázy a tabuliek

Po premyslení presnej štruktúry databázy a vyriešení hore uvedených problémov, som mohol pristúpiť k jej realizácii. Vytvoril som databázu s názvom *dent* pomocou rozhrania phpMyAdmin na prácu s MySQL databázou. Pri vytváraní databázy je potrebné zvoliť názov a zotriedenie, čím sa rozumie kódovanie údajov. Pre účely tejto práce som zvolil zotriedenie na *utf_slovak_ci*.

Vzhľadom na štruktúru práce som vytvoril potrebné tabuľky s názvami *cennik*, *doktori*, *faktury*, *praca*, *udaje*, *user*, *zakaznik* (obr.6).

The screenshot shows the phpMyAdmin interface for a database named 'dent'. The 'praca' table is selected, showing 7 records. The table structure is as follows:

	id	userid	zakaznikid	cennikid	kusy
<input type="checkbox"/>	19	1	36	1	2
<input type="checkbox"/>	26	1	50	26	26
<input type="checkbox"/>	25	1	49	65	98
<input type="checkbox"/>	24	1	49	56	65
<input type="checkbox"/>	23	1	49	23	32
<input type="checkbox"/>	20	1	36	3	4
<input type="checkbox"/>	21	1	36	5	6

Obr. 6 Databáza a tabuľky v phpMyAdmin

Každá tabuľka obsahuje stĺpec *id* s vlastnosťami *PRIMARY KEY* a *auto_increment*, pomocou ktorých sa pre každý riadok tabuľky zaručí jedinečné číslo, a teda nezameniteľná identifikácia v tabuľke.

Aby sa v tabuľkách nevyskytovali redundantné údaje, sú zapisované údaje rozdeľované do viacerých tabuliek. Vďaka tomu sa zachová jednoduchosť práce so zapísanými údajmi. Prakticky som to využil pri nahrávaní novej zákazky do systému. V tabuľke *zakaznik* vystupuje *id* doktora z tabuľky *doktori*, *id* užívateľa z tabuľky *user*. Následne sa *id* zákazky z tabuľky *zakaznik* použije pri zápise položiek z cenníku do tabuľky *praca*, kde sa zapíše aj *id* používateľa z tabuľky *user*. Prepojením tabuliek medzi sebou som získal prehľadnú štruktúru.

Takto navrhnutá databáza je hotová a pripravená na zápis údajov, ktoré budú do nej vkladané pomocou formulárov a následne spracovávané php kódom.

6 Grafický dizajn

6.1 Voľne stiahnuteľná šablóna

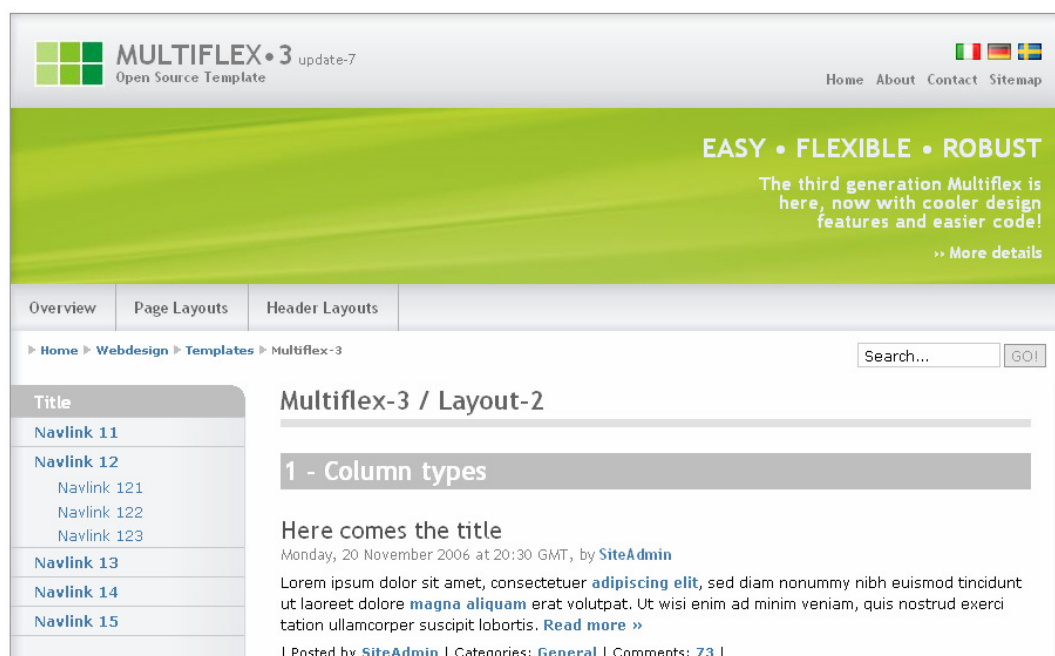
Vzhľad stránky – webdizajn je to prvé s čím prichádza zákazník do kontaktu. Preto je dôležité, aby dizajn stránky dokázal na prvý pohľad osloviť a presvedčiť potenciálneho zákazníka, aby si prečítal niečo viac. Prvý dojem zohráva dôležitú úlohu. V dnešnej dobe je grafický dizajn na vysokej úrovni a nároky na grafické prevedenie sú stále vyššie. Pri tvorení dizajnu je dôležité nezanedbať základné požiadavky medzi ktoré podľa [5] patrí:

- **aktuálnosť** – častá aktualizácia pomáha pri udržiavaní návštevníkov, pretože u Vás vždy nájdete niečo nové
- **prístupnosť** – veľmi dôležité je, aby sa webstránka zobrazila návštevníkom v poriadku a tak, aby všetky informácie boli ľahko čitateľné a všetky ovládacie prvky ľahko ovládateľné
- **použiteľnosť** – na webstránke by sa malo dať ľahko orientovať, aby ju vedel „použiť“ aj ten najnechápavejší návštevník
- **atraktívnosť** – dôležitú úlohu zohráva prvý dojem a „atmosféra“ webstránky

Samotný dizajn je možné riešiť dvomi rôznymi spôsobmi. Prvým je vytvorenie vlastného dizajnu pomocou grafických programov. Keďže tvorbu dizajnu mnohí prirovnávajú k umeleckému dielu a nie každý dokáže byť v tomto zmysle umelcom, existuje aj druhý spôsob, ktorý som zvolil aj ja, a ktorým môžeme vyriešiť problematiku dizajnu. Na internete sa nachádza viacero voľne stiahnuteľných dizajnov, ktoré je možné využiť. Pri tomto spôsobe sa síce stráca jedinečnosť dizajnu, ale na druhej strane je to pomerne lacná a jednoduchá alternatíva.

Ako som už spomínal, vzhľadom na náročnosť a cenu grafických programov som použil voľne stiahnuteľnú šablónu z internetu. Na výber je veľa možností a riešenie sa dá nájsť skoro pre každého.

Na stránke [4] je šablóna, ktorú som použil na navrhovaný systém (obr.7). Je vytvorená pomocou kaskádových štýlov (CSS). Po analýze stiahnutých súborov som implementoval tento dizajn na navrhovaný systém.



Obr. 7 Šablóna dizajnu

6.2 Aplikácia šablóny

Samotná implementácia šablóny prebehla vložení atribútov *class* do html tagov. Takto dané tagy získali atribúty, ktoré sú zapísané v CSS súboroch, podľa toho akú hodnotu má atribút *class*. Aby daná šablóna najlepšie vyhovovala navrhovanému systému upravil som niektoré atribúty šablóny v súboroch CSS.

Šablóna obsahuje viac možností formátovania ako je potrebné. Nepokladal som za nutné nevyužité triedy kaskádového štýlu zmazať, ale doplnil som také triedy, ktoré som potreboval a navrhnutá šablóna ich neobsahovala. Boli to hlavne atribúty tabuliek, kde som zvýrazňoval iné bunky ako predpokladal tvorca šablóny.

Ďalšie zmeny sa týkali ponuky *select* v html formulároch. Ich vzhľad nebol v tabuľke upravovaný a teda bolo nutné ich upravenie pre dodržanie celistvého vzhľadu tabuľky. Tabuľky obsahujú malé obrázky, ktoré slúžia ako odkazy na úpravy záznamov v tabuľkách. Tieto obrázky majú rôznu veľkosť. Najjednoduchším spôsobom ako ich upraviť bolo ich zaradenie do jednej triedy kaskádových štýlov, kde majú pomocou atribútov *width* a *height* zadanú šírku a výšku. V prípade výmeny týchto obrázkov za iné nie je potrebné upravovať ich na potrebný rozmer čo uľahčí prácu pri upravovaní dizajnu tabuľky.

7 Prihlasovanie užívateľov

7.1 Premenná `$_SESSION`

Prvý krok, ktorý musí užívateľ urobiť, aby mu boli sprístupnené jeho dáta, je prihlásenie do systému. Je to spôsob akým sa osobné údaje chránia pred zneužitím inou osobou. Existuje viacero možností, ktorými je možné vytvoriť bezpečný prístup. Jednou z variant je použitie premennej `$_SESSION` jazyka PHP.

Pri práci s touto premennou je nutné zachovať určité postupy. Prvý PHP kód v súbore musí začínať funkciou `session_start()`. Táto funkcia zabezpečí začiatok „sedenia“, čím sa rozumie sledovanie užívateľa počas jeho sedenia so systémom. Zároveň sa vytvorí pre dané sedenie ID pomocou ktorého sa sedenie identifikuje. Vytvorí sa tiež superglobálne pole `$_SESSION`, ktorého úlohou je pamätať si informácie v ňom uložené pri prechode na ďalšiu stránku. Premennú, ktorú chceme registrovať počas sedenia môžeme uložiť v poli `$_SESSION`. V prípade ukončenia sedenia je potrebné zavolať funkciu `session_destroy()`, ktorá odstráni ID daného sedenia.

7.2 Použitie premennej `$_SESSION`

Prvá stránka, ktorá sa načíta sa nazýva *index.php*. Všetky ďalšie podstránky tohto systému sa vkladajú do súboru *index.php*, a preto je to najvhodnejší cieľ na začatie sedenia.

Po odoslaní užívateľského mena a hesla prihlasovacím formulárom (obr. 8) sú tieto údaje porovnávané s údajmi uloženými v tabuľke *user* v MySQL databáze. Heslo uložené v databáze je kryptované pomocou funkcie `crypt()` v PHP, ktorá zabezpečí, aby sa heslo nedalo zistiť pri prehľadávaní databázy nepovolenou osobou. Pomocou vlastnej vytvorenej funkcie *OverU()* sa zistí, či sa kombinácia zadaného užívateľského mena a hesla nachádza v tabuľke a je jedinečná. Potom sa do poľa `$_SESSION` zapíše identifikačné číslo užívateľa a jeho právo na prezeranie informácií. Právo na prezeranie informácií môže nadobúdať tri hodnoty:

- 0 – užívateľ nemá právo prezerat' a vkladat' údaje
- 1 – užívateľ má právo prezerat' iba tie údaje, ktoré sám vložil
- 2 – užívateľ má právo prezerat' všetky údaje, ktoré sú uložené v databáze

► Domov

Prihlásenie

Užívateľské meno:

Heslo:

[Registrácia nového užívateľa](#)

Obr. 8 Prihlasovací formulár

Po úspešnom overení užívateľského mena a hesla sa zobrazí menu s ponukou.

7.3 Registrácia nového užívateľa

Pokiaľ na stránku príde užívateľ, ktorý ešte nemá vytvorené konto, môže toto konto vytvoriť pomocou odkazu *Registrácia nového užívateľa*. Kliknutím na odkaz sa zobrazí formulár na vytvorenie nového užívateľského konta (obr.9).

► Domov ► Registrácia nového užívateľa

Nový užívateľ

Nové užívateľské meno	<input type="text"/>
Heslo	<input type="password"/>
Overenie hesla	<input type="password"/>
Registračný e-mail	<input type="text"/>
Názov organizácie	<input type="text"/>
Ulica, číslo domu	<input type="text"/>
Mesto/Obec	<input type="text"/>
PSČ	<input type="text"/>
Číslo výpisu z OR KS	<input type="text"/>
IČO	<input type="text"/>
DIČO	<input type="text"/>
Peňažný ústav	<input type="text"/>
číslo účtu	<input type="text"/>
5 + 9 =	<input type="text"/>
<input type="button" value="Registovať"/>	

Obr. 9 Registračný formulár

Nové užívateľské meno, heslo, overenie hesla a registračný e-mail sú povinné údaje, ktoré sú potrebné na vytvorenie nového konta. Ostatné údaje registračného formulára sú údaje o užívateľovej obchodnej spoločnosti. Nachádzajú sa tu polia kde sa zadáva názov spoločnosti, adresa, číslo výpisu z obchodného registru, IČO, DIČ a bankové spojenie. Tieto údaje sú povinné, a slúžia na overenie nového užívateľa, pretože používanie systému je určené pre právnické osoby.


Po odoslaní registračného formulára sú odosielané údaje overené. Overuje sa vyplnenie iného užívateľského mena, ako toho čo už existuje v tabuľke *user* a či sa obe zadané heslá zhodujú. Pokiaľ odoslané údaje spĺňajú všetky podmienky zapíše sa užívateľské meno, heslo a e-mail do tabuľky *user*. Pri zápise sa nastaví hodnota stĺpca *pravo* pre nového užívateľa na hodnotu 0. V tejto fáze používateľské konto ešte nie je aktívne. Administrátor systému porovná vložené údaje s údajmi zapísanými v Obchodnom registri Slovenskej republiky. Ak je žiadateľ o konto, ako právnická osoba, zapísaný v Obchodnom registri, tak administrátor aktivuje jeho konto zmenením hodnoty práva z 0 na 1. O aktivácii konta bude užívateľ oboznámený mailom, ktorý mu bude poslaný na mailovú adresu, ktorú zadal do formulára.

Aby som zabránil neželanému odosielaniu registračného formulára robotmi, ktorí znepríjemňujú internet, vložil som na koniec formuláru pole, kde treba vložiť súčet dvoch celých čísel. Výsledok sa overuje a na základe overenia sa vyhodnocuje či sa vyplnený formulár zapíše do databázy.

8 Základné funkcie


V tejto časti sa venujem základným funkciám navrhovaného systému. Medzi základné funkcie patrí zobrazenie údajov, editácia údajov a vymazanie údajov. Pri faktúrach pribúda funkcia navyše, ktorou je vytlačenie faktúry vo formáte PDF.

8.1 Zobrazenie údajov

Podrobnejšie zobrazenie údajov o zázname v tabuľke je prístupné pomocou odkazu, ktorý je reprezentovaný ikonou . Kliknutím na túto ikonu sa zobrazí tabuľka, kde sú zobrazené všetky položky, ktoré sú uložené napríklad v tabuľke *doktori* používanej databázy. Na konci tejto tabuľky sú vyobrazené podobné ikony. Sú to odkazy, pomocou ktorých sa zobrazený údaj môže vymazať alebo upraviť.

Kliknutím na ikonu zobrazenia údajov sa načíta stránka, ktorej URL obsahuje výraz *&do=ukaz+x*. Výraz posiela danej stránke metódou GET typ operácie, ktorý sa má vykonať a id záznamu v tabuľke, ktorého sa daná operácia týka. Na načítanej stránke sa pomocou poľa *\$_GET[]* zistí akú operáciu užívateľ požadoval. V tomto prípade vyhodnotí požadovanú operáciu ako ukázanie záznamu a zobrazí tabuľku, v ktorej sú zobrazené údaje týkajúce sa vybraného záznamu v tabuľke.

8.2 Upraviť


Odkaz na úpravu údajov reprezentuje ikona . Pomocou tohto odkazu je možné upraviť záznam tabuľky. Odkaz na úpravu údajov sa objavuje v tabuľkách prehľadu, kde je zobrazený na konci riadku v stĺpci úpravy a takisto sa nachádza aj v tabuľke zobrazenia údajov na jej konci.

Ikona predstavuje odkaz, v ktorom sa nachádza, podobne ako pri zobrazení údajov, výraz *&do=edit+x*. Pomocou tohto výrazu sa odkazovanej stránke predá metódou GET informácia ohľadom operácie, ktorú používateľ požaduje. Hodnota *edit* naznačuje úpravu údajov, ktoré sú uložené v tabuľke pod ID, ktoré má hodnotu *x*. Vplyvom tejto informácie sa na cieľ odkazu načíta formulár, ktorý slúži na zadávanie nových údajov do tabuľky databázy. Rozdiel oproti vkladaniu nového záznamu je v tom, že v poliach, ktoré treba vyplniť sú už vyplnené údaje. Tieto údaje sú načítané z tabuľky na základe hodnoty ID a vložené vo formulároch pomocou atribútu *value*.

Odoslaním formulára, zobrazeného na úpravy, sa zavolá funkcia *Uprav()*. Funkcia spôsobí úpravu zápisu v tabuľke pomocou požiadavky UPDATE a hodnota stĺpca sa


upraví na požadovanú hodnotu pomocou atribútu SET. Na kontrolu upravených údajov sa automaticky zobrazia údaje z tabuľky s ID záznamom, ktorý bol upravený. V prípade chybného záznamu je možné použiť ikony na úpravu, ktoré sú zobrazené na konci tabuľky.

8.3 Vymazanie záznamu z tabuľky

V prípade neaktuálneho, duplicitného alebo inak nevyhovujúceho záznamu je možné záznam vymazať. Vymazanie záznamu je prístupné cez ikonu  v tabuľke prehľadu záznamov v stĺpci úpravy a v tabuľke zobrazenia údajov na konci. Umiestnenie ikony vymazania je vedľa ikony pre upravenie.

Odkaz na vymazanie obsahuje výraz, ktorý posiela informáciu o operácii požadovanej užívateľom. V prípade vymazania je to výraz `&do=delete+x`. Tak ako v predchádzajúcich úpravách aj v tomto prípade sa metódou GET predáva cieľovej adrese odkazu operácia, v tomto prípade vymazanie. Po kliknutí sa vymaže záznam z tabuľky podľa ID, ktoré je posielané výrazom na mieste *x*. Cieľ odkazu vyhodnotí o akú operáciu používateľ požiadal. Záznam sa vymaže pomocou požiadavky DELETE na MySQL databázu a atribútom, ktorý určí o ktorý záznam sa jedná. Následne po vymazaní záznamu sa zobrazí prehľad údajov danej kategórie už bez vymazaného záznamu.

8.4 Vytlačiť PDF

Pri prehľade faktúr sa nachádza ikona . Pomocou nej je možné vytlačiť PDF dokument faktúry (obr. 10). Vytvorený dokument vo formáte PDF je možné uložiť, odoslať elektronicky alebo vytlačiť. Tieto voľby sú prístupné pomocou programu Adobe Acrobat Reader, ktorý slúži na prehliadanie PDF dokumentov.

Podobne ako v predchádzajúcich úpravách aj tento odkaz obsahuje v URL výraz `?id=7`, ktorý metódou GET posiela cieľu odkazu informáciu, o ktorý záznam z tabuľky *faktury* má používateľ záujem. Údaje sú spracovávané súborom *faktura.php* pomocou ID záznamu z tabuľky *faktury*. Následne sa načítajú všetky potrebné údaje, ktoré sú uložené vo všetkých tabuľkách. Z tabuľky *doktori* je zistená adresa odberateľa podľa id doktora, z tabuľky *udaje* sa vypíšu informácie o dodávateľovi, v tabuľke *zakaznik* sa vyhľadajú záznamy za určený mesiac a rok, čím sa získajú fakturované položky a pomocou tabuľky *cennik* sa určí cena práce a celková fakturovaná čiastka. Všetky údaje sú pomocou triedy FPDF v PHP vypísané vo forme faktúry v dokumente PDF.

DAŇOVÝ DOKLAD - FAKTÚRA		
Dodávateľ:		
J. Z. DENT, s.r.o.	Číslo faktúry:	2009/1
Hlavná 315/16 IČO: 36378739	Platobné podmienky:	bez dane
03852 Sučany DIČO: 2021384882	Forma úhrady:	prevodom
	Konštantný symbol:	0308
Odberateľ		
Výpis z OR KS v Žiline, vl.č. 10998/L	MUDr. Anna Krajčovičová	
Bankové spojenie: VÚB Martin	Košúty	
	03601 Martin	
Číslo účtu: 1244236751/0200	IČO: 35666676	DIČO: 1021864624
Dátum odoslania: 01.01.2009 Dátum splatnosti: 01.01.2009 Dátum zdaniteľného plnenia: 01.01.2009		
Dodacie a platobné podmienky		
Fakturuje Vam práce prevedené v zubnej technike podľa rozpisu protetických prác		
Zákazka	Cena v EUR	
2009/2 Hronský Marek, Šútovo	146,08	
2009/1 Rizoň Ján, Rúčky	20,26	
Celková suma:		166,34 (5 000,92Sk)

Obr. 10 Hlavná časť faktúry

8.5 Vyhľadávanie v tabuľkách

Nad tabuľkou prehľadu, ktorá zobrazuje viac záznamov je umiestnené vyhľadávanie. Pomocou tejto voľby je prístupný formulár na vyhľadávanie v tabuľke nahraných záznamov. Do prvého poľa formulára užívateľ zadáva hľadaný výraz a z druhého poľa vyberá, podľa akého kritéria sa má vyhľadávanie uskutočniť (obr. 11). Požiadavka na vyhľadanie výrazu sa spustí kliknutím na tlačidlo *nájdí*.

Obr. 11 Vyhľadávanie v tabuľke

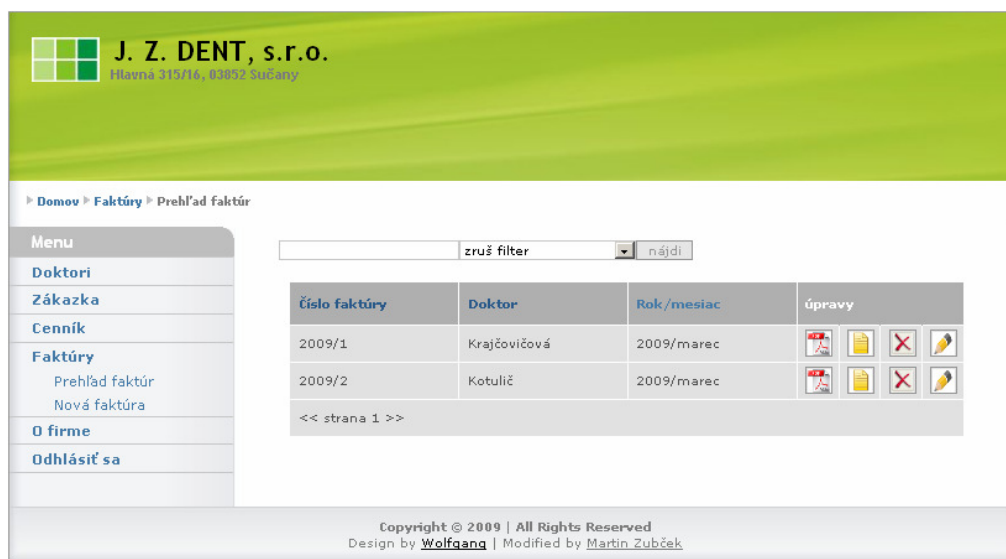
Formulár sa odosiela metódou GET tej istej stránky, na ktorej sa nachádza. Odoslaním vyplneného formulára sa ovplyvní požiadavka na databázu MySQL. Argumenty vlastnej definovanej funkcie na vypísanie obsahu databázy obsahujú premenné *\$searchterm* a *\$searchtype*. Pomocou nich sa z databázy vypíšu len údaje, ktoré spĺňajú zadané kritériá vyhľadávania. Vybrané údaje sa zobrazia vo forme tabuľky

prehľadu údajov a je možné ich prezerat'. Na zrušenie filtra sa prvé pole formuláru ponechá prázdne a v druhom poli sa vyberie voľba zrušenie filtra. Odoslaním takto upraveného formuláru sa filter na vyhľadávanie zruší.

8.6 Navigácia

Dôležitou súčasťou navrhovaného systému je navigácia. Umožňuje pohyb užívateľa v systéme. Úlohou navigácie by malo byť ukázať užívateľovi jeho aktuálnu pozíciu na stránke. Navigáciou sa rozumie menu stránky a takisto aj tzv. navigácia podľa cesty.

Hlavné menu stránky je vytvárané dynamicky z polí, kde sú uložené názvy jednotlivých položiek menu. Položky ponuky sú tvorené odkazmi, ktoré obsahujú výrazy v URL. Tieto výrazy predstavujú premenné posielané metódou GET a podľa hodnôt premenných `$_GET['kod']` a `$_GET['sub']` sa pomocou funkcie *require()* vloží obsah, ktorý je uložený v súbore PHP.



Obr. 12 Navigácia systému

Ukážka navigácie systému je na obr. 12. Ako vidieť na obrázku, nachádzame sa na stránke Prehľadu faktúr. V hornej časti stránky je umiestnená navigácia podľa cesty. Označuje cestu ako sme sa dostali k prehľadu faktúr. Položky navigácie podľa cesty Domov a Faktúry sú hypertextové odkazy. Odkaz Domov načíta stránku, ktorá sa zobrazila ihneď po prihlásení do systému. Kliknutím na položku Faktúry sa zobrazí prvá stránka z danej kategórie, a teda Prehľad faktúr. Posledná položka, Prehľad faktúr, nie je hypertextový odkaz, ale zobrazuje stránku, na ktorej sa užívateľ nachádza.

9 Menu

9.1 Doktori

V sekcii *Doktori* sú uložené údaje o zubných lekároch, ktorí spolupracujú so zubnou technikou. Podľa údajov uložených v tejto sekcii bude neskôr vystavená faktúra, pričom zubný lekár bude na nej vystupovať ako odberateľ.

9.1.3 Prehľad doktorov

Aktuálny zoznam doktorov, s ktorými zubná technika pracuje, je prístupný po kliknutí na položku prehľad doktorov. Zoznam je umiestnený v tabuľke v pravej časti stránky tak, ako je to zobrazené na obr. 13.

J. Z. DENT, s.r.o.
Hlavná 315/16, 83852 Sučany

Domov » Doktori » Prehľad doktorov

Menu

- Doktori
 - Prehľad doktorov
 - Nový doktor
- Zákazka
- Cenník
- Faktúry
- O firme
- Odhlásiť sa

zruš filter nájdi

priezvisko, meno	mesto/obec	úpravy
Libiž Ján	Sučany	
Kotulič Martin	Martin	
Krajčovičová Anna	Martin	
Čubová Antónia	Martin	
Miškolicová Eva	Sučany	

<< strana 1 >>

Copyright © 2009 | All Rights Reserved
Design by Wolfgang | Modified by Martin Zubček

Obr. 13 Prehľad doktorov

V každom riadku je zapísaný jeden doktor. Na konci riadku je stĺpec s názvom úpravy. Každý riadok sa viaže na úpravy ukázať, vymazať a upraviť. Jednotlivé ponuky sú reprezentované ikonami pre zobrazenie, pre vymazanie a pre editáciu. Hlavička tabuľky obsahuje názvy stĺpcov priezvisko, meno a mesto/obec. Po kliknutí na názov stĺpca sa riadky tabuľky zoradia vzostupne a po opätovnom kliknutí sa zoradia zostupne.

9.1.4 Nový doktor

Vkladanie nového záznamu do tabuľky *doktori* je prístupné kliknutím na položku nový doktor v menu v ľavej časti obrazovky. V pravej časti sa zobrazí formulár, kde sú polia na vloženie informácií, ktoré budú potrebné pri vytváraní faktúry. Po odoslaní formulára prostredníctvom tlačidla uložiť sa zobrazí tabuľka s aktuálne uloženými údajmi o novom doktorovi, ktoré je možné ihneď pomocou zodpovedajúcich ikon editovať alebo vymazať.

Odoslaním formulára metódou POST sú údaje predané ďalšej stránke. Poslané údaje sa zapíšu do tabuľky *doktori* pomocou požiadavky INSERT na MySQL databázu. Uložením nových údajov sa hodnota v stĺpci *id* pre práve uložený záznam zvýši o hodnotu jedna od poslednej uloženej hodnoty v tomto stĺpci. Zvýšením hodnoty *id* sa zabezpečí jedinečná hodnota pre zapísaný riadok a v budúcnosti sa na tento riadok bude odkazovať pomocou tejto jedinečnej hodnoty.

9.2 Zákazky

Nová zákazka pre zubnú techniku má podobu protetického štítka. Každý protetický štítok obsahuje viaceré informácie. Vzhľadom na relevantnosť týchto informácií s ohľadom na vytvorenie faktúry nie sú všetky uvedené informácie nevyhnutne potrebné. Informácie, ktoré sú potrebné na identifikovanie jednotlivých zákaziek budú uložené v databáze a použiteľné pri vytváraní faktúry pre zubného lekára.

9.2.3 Prehľad zákaziek

Kliknutím na položku zákazka alebo prehľad zákaziek sa užívateľovi zobrazí tabuľka, v ktorej sú zobrazené zákazky, ktoré má uložené v tabuľke *zakaznik*. Tabuľka vyzerá veľmi podobne ako tabuľka doktorov s tým rozdielom, že tu vystupuje číslo zákazky, meno a priezvisko pacienta a priezvisko doktora, na ktorého sa zákazka vzťahuje. Kliknutím na jednotlivé hlavičky tabuľky je možné záznamy v tabuľke zoradiť zostupne alebo po opakovanom kliknutí vzostupne.

9.2.4 Nová zákazka

Nahrávanie nového protetického štítka sa odohráva v dvoch krokoch. V prvom kroku sa zobrazí formulár na nahranie osobných informácií o pacientovi z protetického štítka. Sú to meno a priezvisko, adresa, kontakt, dátum narodenia a číslo poisťovne. Tieto údaje sa uložia do tabuľky *zakaznik* databázy MySQL po odkliknutí tlačidla *Ďalej* na konci formulára (obr.14).

Domov » Zákazka » Nová zákazka

Menu
Doktori
Zákazka
Prehľad zákaziek
Nová zákazka
Cenník
Faktúry
O firme
Odhlásiť sa

číslo štítku	<input type="text"/>
mesiac	január
rok	<input type="text"/>
Doktor	Čubová
Meno	<input type="text"/>
Priezvisko	<input type="text"/>
Názov ulice, číslo domu	<input type="text"/>
Mesto/Obec	<input type="text"/>
PSČ	<input type="text"/>
dátum narodenia	<input type="text"/>
poistovňa	<input type="text"/>
kontakt	<input type="text"/>
<input type="button" value="Ďalej"/>	

Obr. 14 Prvý krok pridania novej zákazky

Druhým krokom nahrávania novej zákazky do systému je nahranie položiek cenníku, ktoré sa budú vypracovávať v zubnej technike. Zobrazený formulár ponúka dva stĺpce s názvom kód a množstvo (obr. 15). Po načítaní formulára je možné vypísať iba jeden riadok a teda iba jednu položku. Na konci tabuľky s formulárom je odkaz *d'alší riadok*, pomocou ktorého je možné pridať toľko riadkov, koľko je potrebné. Po kliknutí na tlačidlo uložiť sa zobrazí tabuľka s aktuálne uloženými údajmi, ktoré je možné ihneď podľa potreby editovať. Uložené údaje sa zapisujú do tabuľky *praca*.

Domov » Zákazka » Nová zákazka

Menu
Doktori
Zákazka
Prehľad zákaziek
Nová zákazka
Cenník
Faktúry
O firme
Odhlásiť sa

kód	množstvo
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
d'alší riadok	<input type="button" value="uložiť"/>

Obr. 15 Druhý krok pridania novej zákazky

9.3 Cenník

Cenník uložený v tomto systéme slúži pri fakturácii, kedy sa pomocou kódov, ktoré sú uložené v tabuľke *praca*, vypočítava celková fakturovaná suma a aj jednotlivé sumy za každú fakturovanú zákazku.

9.3.3 Prehľad cenníka

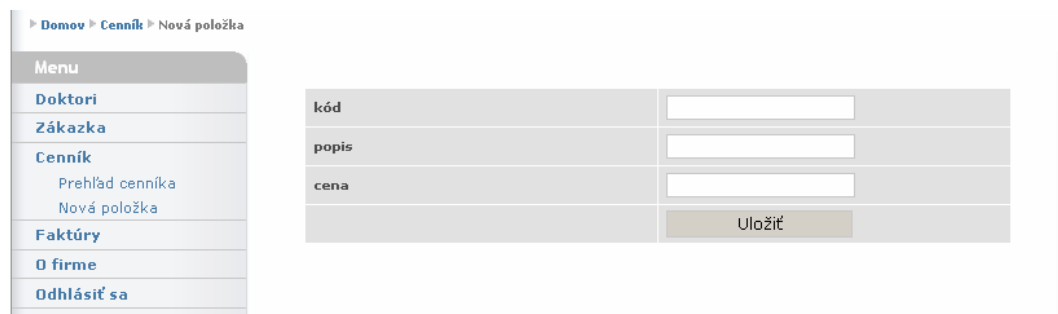
Prehľad položiek, ktoré sú uložené v cenníku, je ľahko prístupný pomocou odkazu v ľavej časti stránky, v menu, uvedenom ako prehľad cenníka. Zobrazená tabuľka má v hlavičke názvy vo forme odkazov. Po kliknutí na názov stĺpca sa údaje v tabuľke zoradia vzostupne podľa údaju v danom stĺpci. Opätovným kliknutím na tú istú hlavičku sa údaje v tabuľke zoradia zostupne.

Na konci každého riadku je stĺpec s úpravami. Pomocou ikon sú prístupné voľby zobrazenie, úpravy a vymazanie. Funkcia jednotlivých ikon totožná ako v prípade úprav záznamov v prehľade doktorov.

9.3.4 Nová položka

V prípade nahrávania novej položky k cenníku je táto možnosť prístupná pomocou novej položky v menu v ľavej časti stránky. Na nahranie novej položky je vytvorený formulár (obr.16). Vyplňajú sa v ňom polia kód, popis a cena. Kód je päťciferné číslo, pomocou ktorého sa na protetickom štítku označuje, ktoré položky boli pri danej práci účtované. Opis slúži na opísanie položky, ktorej sa týka kód a cena. Do cenníka sa nahráva cena v EUR. Pri fakturácii sa pomocou oficiálneho kurzu prepočíta informatívna cena v Slovenských korunách.

Po odoslaní formuláru sa metódou POST informácie odošlú na spracovanie. Pomocou PHP skriptu sa údaje zapísané v poliach formuláru uložia do tabuľky *cennik* v MySQL databáze.



Domov » Cenník » Nová položka	
Menu	
Doktori	
Zákazka	
Cenník	
Prehľad cenníka	
Nová položka	
Faktúry	
O firme	
Odhlásiť sa	

kód	<input type="text"/>
popis	<input type="text"/>
cena	<input type="text"/>
Uložiť	

Obr. 16 Nová položka cenníka

9.4 Faktúry

Cieľom systému na správu protetických štítkov je vytvorenie faktúry za určený mesiac pre odberateľa, t.j. doktora. Faktúra v sebe zahŕňa informácie zo všetkých tabuliek používanej databázy. Z tabuľky *doktori* sa použijú údaje o odberateľovi, tabuľka *zakaznik* poskytuje informácie o pacientoch, teda o zákazkách, ktoré sa vykonali. Pomocou tabuľky *praca* sa v kooperácii s tabuľkou *cennik* určí suma za jednotlivé vykonané zákazky a celková suma za všetky zákazky, ktoré sú fakturované. Tabuľka *udaje* poskytuje informácie o dodávateľovi, teda o firme, ktorá faktúru vystavila.

9.4.3 Prehľad faktúr

Po vytvorení novej faktúry sa údaje potrebné na vytvorenie faktúry uložia do tabuľky *faktury* a pomocou prehľadu faktúr je možné ich editovať, vymazať, upraviť alebo vytlačiť. Voľby úprav sú prístupné cez ikony v poslednom stĺpci tabuľky prehľadu faktúr (obr. 17). Oproti iným tabuľkám prehľadu je pri prehľade faktúr o jednu voľbu viac. Jedná sa o možnosť zobraziť faktúru vo formáte PDF. Pri voľbe zobrazenia sa zobrazia len údaje, ktoré sú uložené v tabuľke *faktury* v databáze a nie konkrétna faktúra. Faktúra sa zobrazí len po kliknutí na ikonu PDF a v novom okne sa zobrazí faktúra vytvorená podľa zadaných kritérií.

J. Z. DENT, s.r.o.
Hlavná 315/14, 83852 Sučany

Domov > Faktúry > Prehľad faktúr

Menu

- Doktori
- Zákazka
- Cenník
- Faktúry
 - Prehľad faktúr
 - Nová faktúra
- O firme
- Odhlásiť sa

zruš filter [vypadať] nájdi

Číslo faktúry	Doktor	Rok/mesiac	úpravy
2009/1	Krajčovičová	2009/marec	[X] [PDF] [X] [edit]
2009/2	Kotulič	2009/marec	[X] [PDF] [X] [edit]

<< strana 1 >>

Copyright © 2009 | All Rights Reserved
Design by [Wolfgang](#) | Modified by [Martin Zubček](#)

Obr. 17 Prehľad faktúr

Stĺpce, ktoré obsahujú údaje, majú v hlavičke názov s odkazom. Pomocou tohto odkazu je možné zoradiť záznamy v tabuľke. Po prvom kliknutí sa zobrazia údaje podľa daného stĺpca vzostupne a po opakovanom kliknutí sa zobrazia zostupne. V prípade stĺpcu mesiac/rok sa údaje najprv zoradia podľa roku a následne podľa mesiaca. Údaje tak budú po sebe nasledovať podľa rokov a v rámci jedného roka podľa mesiacov.

9.4.4 Nová faktúra

Využitím údajov všetkých tabuliek databázy sa vytvorí nová faktúra. Nová faktúra sa vytvára pomocou súboru *faktura.php*. Údaje potrebné na vytvorenie faktúry je potrebné vpísať do formulára. Vo formulári sa vyplňajú polia, ktoré sa na faktúre vyskytujú a nie sú doteraz uložené v žiadnej tabuľke. Sú to údaje ako číslo faktúry, dátum odoslania a spracovania faktúry, dátum zdaniteľného plnenia, spôsob úhrady, podmienky a konštantný symbol. Všetky ostatné údaje budú načítané z tabuliek podľa zvoleného doktora, mesiaca a roka. PHP kód na stránke *faktura.php* vyhľadá na základe zvolených údajov pacientov z tabuľky *zakaznik* za daný mesiac a rok a pomocou tabuľky *praca* v kooperácii s tabuľkou *cennik* vypočíta sumu za jednotlivé práce a celkovú fakturovanú sumu. Pomocou zvoleného doktora sa načítajú údaje o odberateľovi a z tabuľky *udaje* sa vyberú údaje o dodávateľovi. Po odoslaní formuláru sa zobrazí prehľad uložených údajov, pričom na konci zobrazenej tabuľky sú voľby úprav. Je možné práve uložené údaje vymazať, upraviť alebo vytvoriť PDF dokument. PDF dokument sa generuje dynamicky pomocou údajov uložených v databáze a výsledný dokument sa neukladá pokiaľ ho užívateľ sám neuloží.

9.5 O firme

Medzi informácie potrebné na vytvorenie faktúry patria aj informácie o firme ako o dodávateľovi. Tieto údaje sa zapisujú už pri registrácii nového užívateľa. Registračné údaje nie je možné vymazať, iba upraviť. V prípade, že užívateľ zmení nejakú položku, zmeny sa neprejavia ihneď. Upravené registračné údaje administrátor najprv skontroluje a porovná s databázou Obchodného registra Slovenskej republiky a až potom sa zmeny prejavajú.

9.5.3 Prehľad údajov

Tabuľka *udaje* obsahuje informácie o danej firme, a teda nie je potrebná tabuľka, kde sa zobrazia údaje ako v predchádzajúcich prehľadoch. Prehľadom pri údajoch o firme je zobrazenie údajov, ktoré sú uložené v tabuľke *udaje* pre daného užívateľa. Na

konci tejto tabuľky je voľba upraviť. Kliknutím na ikonu úpravy údajov sa zobrazí formulár úpravy údajov s vyplnenými poľami podľa hodnôt uložených v databáze MySQL.

Názov firmy:	J. Z. DENT, s.r.o.
Ulica, číslo domu:	Hlavná 315/16
Mesto/Obec:	Sužany
PSČ:	03852
Číslo výpisu z OR KS:	10998/L
IČO:	36378739
DIČO:	2021384882
Peňažný ústav:	VÚB Martin
Číslo účtu:	1244236751/0200

Copyright © 2009 | All Rights Reserved
Design by [Wolfgang](#) | Modified by [Martin Zubček](#)

Obr. 18 Prehľad údajov o firme

9.5.4 Úprava údajov

Kliknutím na úpravu údajov sa načíta formulár s údajmi, ktoré sú už v databáze vyplnené. Formulár obsahuje polia na zápis názvu firmy, adresy sídla firmy, bankového spojenia. Zapisuje sa tu aj IČO, DIČ a číslo výpisu z obchodného registra. Odoslaním formuláru sa informácie odošlú stránke, na ktorej sa údaje zapíšu do databázy, pričom hodnota stĺpca *xid* tabuľky *udaje* sa nastaví na hodnotu 2. Táto hodnota zabezpečí, že údaje budú pre užívateľa skryté. Až po prekontrolovaní zmenených údajov administrátorom, ako bolo vyššie spomenuté, sa táto hodnota zmení z 2 na 1. Staré údaje sa vymažú a naposledy uložené údaje budú viditeľné a použiteľné pre užívateľa.

9.6 Odhlásenie

Poslednou položkou v ponuke na stránke je možnosť odhlásiť sa zo systému. Odhlásenie je jediná bezpečná možnosť ako ukončiť prácu so systémom. Odhlásením sa

zabráni zneužitiu prihlasovacieho mena neoprávnenou osobou a zároveň sa zabezpečí, že prístup k údajom uloženým v databáze systému má iba oprávnená osoba.

Odhlásenie predstavuje odkaz na úvodnú stránku systému, t.j. *index.php*. Po načítaní úvodnej stránky bez premenných, ktoré sa posielajú metódou GET pomocou URL, sa aktuálne sedenie ukončí a začne sa nové pre prípad prihlásenia iného užívateľa.

Záver

Jednou z dôležitých častí firemnej administratívy je fakturácia. Vytváranie faktúr pomocou rôznych systémov zjednodušuje a urýchľuje inak zdĺhavú prácu. Cieľom tejto diplomovej práce bolo vytvorenie systému na správu protetických výkonov, ktorého výstupom je faktúra.

Pred začatím vytvárania systému bolo potrebné získať informácie o aktuálnom stave a o požiadavkách, ktoré očakáva firma J. Z. DENT s.r.o. od systému. Po analýze problému a vytvorení cieľov bola vytvorená databáza MySQL so všetkými potrebnými tabuľkami ako základ pre ukladanie dát v systéme. Na prácu s uloženými údajmi bol vytvorený kód systému v jazyku PHP. Postupne boli vytvorené formuláre na vkladanie a editáciu údajov z protetického štítka a tabuľky prehľadu na zobrazenie uložených údajov. V prípade prehľadu faktúr bola vytvorená možnosť vygenerovať faktúry vo formáte PDF s možnosťou tlače. Celý systém je možné umiestniť na webový systém firmy.

Samotný systém má niekoľko výhod. Vďaka riešeniu na mieru obsahuje systém len tie ovládacie prvky, ktoré si firma vyžiadala, čo zvyšuje efektivitu práce so systémom. Druhou výhodou je možnosť vygenerovania faktúry v elektronickej podobe, vo formáte PDF. Umiestnenie systému na webovú stránku firmy umožňuje prístup a editáciu nahraných údajov z viacerých miest. Zamestnanci firmy tak môžu pracovať so systémom z rôznych pracovísk.

V budúcnosti by bolo ideálne doplniť tento systém podobným systémom určeným pre zubné ambulancie. Obidva systémy by navzájom spolupracovali a informácie, ktoré sa v dnešných dňoch vymieňajú medzi zubnou technikou a ambulanciou formou protetických štítkov, by tak mohli byť odosielané elektronickou formou. Zvýšila by sa tým efektivita práce, pretože údaje z protetických štítkov by sa už nemuseli manuálne prepisovať do systému.

Zoznam bibliografických odkazov

- [1] ŠKULTÉTY, R.: *JavaScript: Programujeme internetové aplikácie*. Brno: Computer Press, 2004. ISBN 80-251-01444-4.
- [2] NARAMORE, E. a i.: *Vytváříme webové aplikace v PHP5, MySQL a Apache*. Brno: Computer Press, 2006. ISBN 80-251-1073-7.
- [3] WELLING, L., THOMSON, L.: *PHP a MySQL: rozvoj webových aplikací*. Praha: SoftPress, 2004. ISBN 80-86497-60-7.
- [4] 1234.info: Webtemplates [online], © 2008 [cit. 2009-05-07]. Dostupné na internete <<http://www.1234.info/webtemplates/>>
- [5] GLUCHMAN, T.: Webdesign. [online], © 2006 [cit. 2009-05-07]. Dostupné na internete <<http://gluchman.sk/webdesign/>>
- [6] OROSZ, P.: CSS – 2. Syntax. [online], © 2006 [cit. 2009-05-07]. Dostupné na internete <<http://programujte.com/index.php?akce=clanek&cl=2007040103-css-%E2%80%932-syntax>>
- [7] FPDF Library: PDF generator. [online], [cit. 2009-05-08]. Dostupné na internete <<http://www.fpdf.org/>>.
- [8] The PHP Group. PHP. [online], © 2001 -2009, Dostupné na internete <<http://www.php.net/>>.

Prílohy

Zdrojové kódy ako aj elektronická verzia diplomovej práce sa nachádzajú na priloženom CD.