

# **Fakturační systém**

Invoice system

Jakub Nacík

---

Bakalářská práce  
2012



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2011/2012

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub NACÍK**  
Osobní číslo: **A08071**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Fakturační systém**

Zásady pro vypracování:

1. Analyzujte problematiku a vypracujte literární řešení na dané téma.
2. Navrhněte strukturu fakturačního systému.
3. Vytvořte databázi a webovou aplikaci systému.
4. Zajistěte správu a zabezpečení systému.
5. Implementujte systém.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. PROKOPOVÁ, Zdenka. Databázové systémy MySQL+PHP. FAI UTB Zlín, s. 126, 2006, Vysokoškolská skripta. ISBN 80-7318-486-9.
2. LACKO, Lubomír. PHP a MySQL – hotová řešení. Brno : Computer Press, 2006. ISBN 80-251-1249-7.
3. RIORDAN, Rebecca M. Vytváříme relační databázové aplikace. Praha : Computer Press, 2000, 280 s. ISBN 80-7226-360-9.
4. SCHNEIDER, R.,D. MySQL – Oficiální průvodce tvorbou, správou a laděním databází. Grada, ISBN 80-247-1516-3.
5. ULLMAN, L. PHP a MySQL, Brno : Computer Press, 2004, s. 534, ISBN 80-251-0063-4.
6. WELLING, L., THOMSON, L. MySQL Průvodce základy databázového systému. Brno : Computer Press, 2005. ISBN 80-251-0671-3.

Vedoucí bakalářské práce:

**doc. Ing. Zdenka Prokopová, CSc.**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**24. února 2012**

Termín odevzdání bakalářské práce:

**8. června 2012**

Ve Zlíně dne 24. února 2012

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



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

## **ABSTRAKT**

Cieľom mojej bakalárskej práce bolo vytvoriť webovú aplikáciu pre komplexnú evidenciu elektronickej fakturácie firmy poskytujúcej služby v oblasti webhostingu. V teoretickej časti práce sa nachádzajú základné informácie o problematike ako aj priblíženie jednotlivých použitých technológií. Praktická časť popisuje samotnú tvorbu webovej aplikácie a jej jednotlivých súčastí spolu s popisom použitých technických riešení.

Klíčová slova: webová aplikácia, HTML, CSS, PHP, MySQL

## **ABSTRACT**

The objective of my bachelor thesis was to create a web based application for complex management of electronic invoices of webhosting company. In the theoretical part of the thesis are the basic information about this topic and examination of used technologies. Practical part describes creation of application and it's components with description of the technical solutions.

Keywords: web application, HTML, CSS, PHP, MySQL

Ďakujem vedúcej mojej bakalárskej práce, doc. Ing. Zdenke Prokopovej, CSc. za jej trpezlivosť a všetky cenné rady, ktoré mi počas tvorby tejto práce odovzdala.

Motto:

"A person who never made a mistake never tried anything new."

~ Albert Einstein

**Prohlašuji, že**

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

**Prohlašuji,**

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

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD.....</b>	<b>9</b>
<b>I. TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 TEORETICKÁ ŠPECIFIKÁCIA SYSTÉMU.....</b>	<b>11</b>
1.1 PREDSTAVENIE ČINNOSTI SPOLOČNOSTI.....	11
1.2 SÚČASNÝ STAV KOMUNIKÁCIE S KLIENTAMI A FAKTURÁCIE SLUŽIEB .....	11
1.3 TEORETICKÝ NÁVRH RIEŠENIA .....	12
<b>2 TECHNOLOGIE POUŽITÉ PRI TVORBE SYSTÉMU.....</b>	<b>13</b>
2.1 HTML .....	13
2.1.1 <i>História HTML</i> .....	13
2.1.2 <i>Súčasnosc' HTML</i> .....	13
2.1.3 <i>Prehľad verzií HTML podľa dátumu vydania</i> .....	14
2.2 CSS .....	14
2.2.1 <i>História CSS</i> .....	15
2.2.2 <i>Prehľad verzií CSS</i> .....	15
2.3 PHP .....	16
2.3.1 <i>História PHP</i> .....	16
2.3.2 <i>Súčasnosc' PHP</i> .....	16
2.3.3 <i>Syntax PHP</i> .....	17
2.4 MySQL .....	18
2.4.1 <i>Syntax MySQL príkazov</i> .....	18
<b>3 POUŽITÉ PROGRAMOVÉ VYBAVENIE .....</b>	<b>20</b>
3.1 CODA .....	20
3.2 TRANSMIT .....	21
<b>II. PRAKTICKÁ ČÁST .....</b>	<b>22</b>
<b>4 REALIZÁCIA APLIKÁCIE .....</b>	<b>23</b>
4.1 ŠTRUKTÚRA SYSTÉMU.....	23
4.2 NÁVRH UŽÍVATELSKÉHO ADMINISTRAČNÉHO ROZHRANIA.....	24
4.3 NÁVRH ŠTRUKTÚRY DATABÁZ.....	27
4.3.1 <i>Tabuľka "users"</i> .....	27
4.3.2 <i>Tabuľka "clients"</i> .....	28

4.3.3	Tabuľka "domains".....	30
4.3.4	Tabuľka "invoices" .....	31
4.3.5	Tabuľka "invoices_subjects" .....	32
4.3.6	Tabuľka "pricelist" .....	33
4.4	FUNKČNÉ POZADIE SYSTÉMU .....	34
4.4.1	Objektové PHP .....	34
4.4.2	Načítavanie podstránok pomocou funkcie include() .....	35
4.4.3	Pekné URL pomocou mod_rewrite.....	37
4.4.4	Automatické operácie .....	38
4.4.5	Prihlasovanie užívateľov .....	39
4.5	FUNGOVANIE SYSTÉMU V PRAXI .....	40
4.5.1	Objednávanie nových služieb .....	40
4.5.2	Potvrdenie objednávky pracovníkom .....	42
4.5.3	Evidencia úhrady predfaktúry .....	43
4.5.4	Štatistiky .....	44
	<b>ZÁVĚR .....</b>	<b>46</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>47</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>48</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>49</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>50</b>
	<b>SEZNAM TABULEK.....</b>	<b>51</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>52</b>

## ÚVOD

Mojou úlohou v tejto bakalárskej práci bolo navrhnúť a následne zrealizovať elektronický fakturačný systém pre webhostingovú spoločnosť, ktorý bude fungovať ako webová aplikácia a ktorého hlavnou úlohou bude komplexná evidencia zákazníkov a služieb, ktoré využívajú ako aj spracovávanie nových objednávok a správa fakturácie v elektronickej podobe. Pred spracovaním mojej bakalárskej práce fungoval celý proces objednávok, fakturácie a evidencie na princípe manuálneho zapisovania a sledovania údajov. S pribúdajúcim množstvom klientov a objednávok začalo byť problematické kontrolovať celý proces a zachovávať evidenciu jasnú a prehľadnú.

Z dôvodu zefektívnenia práce bolo rozhodnuté vytvoriť nový fakturačný systém. Elektronický fakturačný systém a jeho procesy som navrhol tak, aby mal zodpovedný pracovník firmy vždy jednoduchý prístup k všetkým potrebným informáciám a pritom sa nemusel starať o množstvo každodenných rutinných operácií, ako napríklad rozposielanie výziev na predĺženie služieb, či upomienok za neuhradené služby. Systém sa o tieto veci stará sám a pracovník je tak oslobodený od množstva úloh. Podobne je to aj s evidenciou úhrad, kedy pracovník jednoducho pomocou jedného kliknutia zaznamená úhradu objednávky a systém sám vygeneruje faktúru za služby a vo formáte PDF ju doručí na kontaktnú e-mailovú adresu zákazníka.

Pri návrhu systému som dbal na jeho robustnosť, keďže je v budúcnosti predpoklad jeho ďalšieho rozširovania o nové funkcionality.

## **I. TEORETICKÁ ČÁST**

# 1 TEORETICKÁ ŠPECIFIKÁCIA SYSTÉMU

## 1.1 Predstavenie činnosti spoločnosti

Spoločnosť host4web, s.r.o. sa zaoberá poskytovaním širokého spektra služieb v oblasti webhostingu internetových stránok, prenájmu a správy serverov ako aj registrácii národných a nadnárodných domén.

## 1.2 Súčasný stav komunikácie s klientmi a fakturácie služieb

V súčasnej podobe funguje komunikácia s klientmi ako aj objednávanie a fakturovanie služieb už niekoľko rokov v nezmenenej podobe. Drvivá väčšina každodennej komunikácie prebieha pomocou elektronickej pošty. Najrôznejšie požiadavky smerujú klienti na príslušnú e-mailovú adresu, ktorú sledujú pracovníci technickej podpory a tí riešia jednotlivé požiadavky. Len malé percento klientov používa telefonický kontakt, väčšinou len v prípadoch, keď sa jedná o súrne vyriešenie problému. Pracovníci technickej podpory majú taktiež na starosti vybavovanie nových objednávok na jednotlivé služby, či už z oblasti webhostingu, registrácie domén alebo ich kombinácii. Súčasný proces objednania služieb funguje na princípe vyplnenia formulára na webovej stránke spoločnosti, kde zákazník špecifikuje svoje požiadavky, vyplní kontaktné a fakturačné údaje a odošle objednávku. Tá v podobe elektronickej správy príde na určenú adresu, kde ju následne pracovník manuálne spracuje. Úlohou pracovníka je zriadiť objednané služby na serveri spoločnosti, vytvoriť prístup k webovému priestoru pre zákazníka a vytvoriť faktúru za objednané služby a túto elektronickeou poštou doručiť klientovi spolu s prístupovými údajmi k webovému priestoru. Tento proces je pomerne náročný na jednotlivé kroky, vyžaduje si množstvo času a neposkytuje pracovníkom dostatočnú kontrolu nad jeho priebehom. Je nutné manuálne kontrolovať dodržiavanie splatnosti faktúr, predlžovanie služieb po uplynutí predplateného obdobia a zrušenie existujúcich služieb pokiaľ ich zákazník nepredĺžil na ďalšie obdobie. S narastajúcim počtom klientov začínal byť celý proces objednávanie a fakturácie služieb veľmi neprehľadný, takmer nekontrolovateľný a časovo veľmi náročný.

### 1.3 Teoretický návrh riešenia

Na základe vzniknutej situácie bolo potrebné začať proces fakturácie a objednávania služieb postupne automatizovať. Teda vytvoriť systém, ktorý by na základe prijatých objednávok generoval faktúry a doručoval ich v elektronickej forme klientom, dohliadal na dodržiavanie splatností faktúr a včas upozorňoval klientov na blížiaci sa koniec obdobia predplatených služieb. Vedenie spoločnosti sa rozhodlo investovať do vývoja vlastného systému, ktorý by fungoval vo webovom rozhraní, čo mu umožňuje absolútnu platformnú nezávislosť a dostupnosť prakticky kedykoľvek a kdekoľvek. Takýto systém by značným spôsobom šetril prácu a čas pracovníkom technickej podpory a tým aj znižoval ich mzdové náklady. Umožňoval by taktiež podrobné štatistiky a prehľady o využívaní jednotlivých služieb. Do budúcnosti by bolo možné systém rozšíriť o široké spektrum funkcionalít, napríklad evidenciu e-mailových požiadaviek na technickú podporu, takzvaný tiketový systém.

## 2 TECHNOLOGIE POUŽITÉ PŘI TVORBE SYSTÉMU

### 2.1 HTML

HTML (HyperText Markup Language) je hlavním značkovacím jazykem webových stránek. Jednotlivé HTML elementy tvoří základně stavebné prvky všech webových stránek. Jazyk HTML tvoří značky - elementy, které se zapisují v tvare ostrých zátvoriek - například `<html>`, v rámci kterých je definovaný obsah webové stránky. HTML tvoří převážně párové značky – například `<div> </div>`, poznáme však také i nepárové značky - například `<img>`. Zdrojové soubory webových stránek napsané v HTML se ukládají s příponou `.html` (případně `.htm`). Úlohou webového prohlíдача je HTML soubor přečíst a uvedené značky interpretovat do vizuální podoby webové stránky. HTML značky umožňují vkládat do webových stránek textový či obrázkový materiál nebo vytvářet formuláře, které mohou návštěvníci stránky vyplňovat přímo v prohlíдаči.

#### 2.1.1 Historie HTML

Začátky jazyka HTML sahají do roku 1980, kdy fyzik Tim Berners-Lee pracující v CERNe hledal způsob, ako by mohl v rámci CERNu sdílet dokumenty. Jeho první prototyp se nazýval ENQUIRE. V roce 1989 Berners-Lee specifikoval HTML a vytvořil první webový prohlíдаč. První veřejně dostupná specifikace HTML byla v dokumente „HTML Tags“ koncem roku 1991. Tento dokument popisoval 18 HTML elementů, které umožňovali tvořit první jednoduché webové stránky. V tom čase byl jazyk HTML součástí jazyka SGML, který byl standardním formátem v rámci CERNu.

#### 2.1.2 Súčasnost' HTML

V súčasnosti je na svete v poradí piate vydanie jazyka html oficiálne nazvané HTML5. HTML5 je stále vo vývoji, no teší sa čoraz väčšej popularite vo svete internetu. Jeho hlavným cieľom je zlepšiť podporu jazyka pre najmodernejšie multimédiá a zároveň zostať ľahko čitateľné pre ľudí a konzistentné pre zariadenia. HTML5 je výsledkom pozorovania, že pri bežnom používaní HTML vo verzii 4 a XHTML vo verzii 1 dochádza na webových stránkach a vo webových prohlíдаčoch k miešaniu rôznych špecifikácií, ktoré vedie k mnohým chybám v existujúcich webových stránkach. HTML5 je tiež vhodným kandidátom na použitie pri tvorbe multiplatformných mobilných aplikáciách pre mobilné

zariadenia akými sú smartfóny, tablety a podobne. Štandard HTML5 zavádza mnohé nové prvky, ktoré umožňujú jednoducho vkladať do webovej stránky video alebo audio obsah.

### 2.1.3 Prehľad verzií HTML podľa dátumu vydania

- November 1995 - HTML 2.0
- Január 1997 - HTML 3.2 - publikované ako odporúčanie konzorcia W3C
- December 1997 - HTML 4.0 - publikované ako odporúčanie konzorcia W3C, umožňovalo 3 varianty
  - **strict** - zakazovala použitie zaniknutých značiek
  - **transitional** - dovoľovala použiť zaniknuté značky
  - **frameset** - povoľovala rámcové elementy
- Apríl 1998 - HTML 4.0 prešlo malými úpravami bez zmeny čísla verzie
- December 1999 - HTML 4.01 - publikované ako odporúčanie konzorcia W3C. Poznalo tie isté varianty ako verzia 4.0
- Máj 2000 - "ISO HTML" - verzia HTML 4.01 Strict bola publikovaná ako ISO/IEC medzinárodný štandard.
- Január 2008 - pracovná verzia HTML5

## 2.2 CSS

Jazyk CSS (Cascading Style Sheets) patrí podobne ako HTML k základným stavebným prvkom webových stránok. CSS je primárne navrhnuté na formátovanie dokumentu napísaného v HTML (alebo v inom príbuznom značkovacom jazyku). Medzi najpoužívanejšie funkcie CSS patrí nastavovanie fontov, farieb, okrajov či pozadí. Pomocou CSS je možné taktiež špecifikovať pozície jednotlivým HTML elementom, čo umožňuje veľmi podrobne pracovať so štruktúrou webovej stránky. CSS taktiež umožňuje nastavovať HTML dokumentu rôzne vlastnosti podľa toho, v akej podobe je HTML dokument prezentovaný. Môžeme teda upraviť jeho vzhľad ak ide dokument do tlače alebo je zobrazovaný na monitore s malým rozlíšením.

### 2.2.1 História CSS

Štýly existovali už v počiatkoch SGML v roku 1980. Kaskádové štýly boli vyvinuté s cieľom, aby bolo formátovanie webových dokumentov jednotné. S postupným rozširovaním popularity internetu a pribúdajúcimi verziami jazyka HTML sa samozrejme vyvíjalo aj CSS, no celkovo poznáme len 4 kompletne verzie CSS.

### 2.2.2 Prehľad verzií CSS

- **CSS 1** - prvá špecifikácia sa stala oficiálnym odporúčaním konzorcia W3C, bola publikovaná v decembri roku 1996 a podporovala hlavne nastavovanie fontov, farby textu, pozadia a iných elementov, rôzne atribúty textu ako napríklad medzery medzi slovami, zarovnanie textu či obrázkov. Nechýbala možnosť nastavenia vonkajších a vnútorných okrajov jednotlivým elementom.
- **CSS 2** - druhá verzia CSS bola vyvíjaná a publikovaná konzorciom W3C v máji roku 1998. Oproti verzii 1 obsahovala množstvo nových atribútov medzi ktoré patrí nastavovanie pozície elementu, možnosť nastaviť z-index elementom (používa sa pre rozlíšenie poradia prekrývajúcich sa elementov), pribudol koncept pre rozlišovanie média na ktorom sa webová stránka zobrazuje a taktiež základ pre tieňovanie textu.
- **CSS 2.1** - jedná sa o korektúru vydania CSS 2, v ktorej boli hlavne opravené chyby odhalené v CSS 2 a odstránené funkcie, ktoré mali slabú podporu naprieč prehliadkami. Verzia CSS 2.1 bola prvý krát uverejnená vo februári 2004. Počas svojej existencie prešla niekoľkými revíziami a oficiálne schválená konzorciom W3C bola až v júni 2011.
- **CSS 3** - táto verzia priniesla radu novinek, ktoré umožňujú pridávať tieň pod elementy, prípadne zobrazit' oblé rohy a umožňuje tiež základné animácie prvkov. Najväčšou nevýhodou je ešte stále pomerne slabá podpora prehliadačov, najhoršie je na tom Internet Explorer. Veľkou výhodou CSS 3 je jeho spätná kompatibilita, takže ak je použité CSS 3 a zobrazujeme webovú stránku v prehliadači, ktorý CSS 3 nepodporuje, jednoducho sa funkcie z CSS 3 vynechajú a webová stránka sa zobrazí bez nich, no plnohodnotne. V praxi to znamená, že pri aplikovaní oblých rohov na určitý HTML element sa tieto zobrazia v prehliadači Internet Explorer ako

ostré. Všetky ostatné vlastnosti ako napríklad farba pozadia či textu ostanú zachované.

## 2.3 PHP

PHP (Hypertext Preprocessor) je univerzálny skriptovací jazyk, ktorý sa vykonáva na strane servera. PHP bolo navrhnuté pre vývoj dynamických webových aplikácií. Je jedným z prvých skriptovacích jazykov, ktoré je možné priamo vkladať do zdrojových kódov HTML dokumentov a vzkonávať tak najrôznejšie dynamické akcie (napríklad odosielanie vyplneného formuláru) priamo v konkrétnom dokumente, bez nutnosti volania externého súboru, ktorý by vykonal požadovanú operáciu. PHP kód je interpretovaný na strane webového servera, ktorý skript spracuje a vygeneruje výslednú webovú stránku v čistom HTML. Ku koncovému užívateľovi sa teda vždy dostáva len konkrétny HTML výsledok, nie je možné vidieť syntax PHP skriptov cez webový prehliadač.

### 2.3.1 História PHP

PHP ako ho poznáme dnes je nasledovníkom produktu zvaného PHP/FI, ktorý bol vytvorený v roku 1994 Rasmusom Lerdorfom. Prvá podoba PHP bola napísaná v jazyku C a slúžila na zaznamenávanie návštevnosti na Lerdorfovej osobnej stránke. Tento jednoduchý súbor skriptov pomenoval ako "Personal Home Page Tools". Postupom času pribúdala stále nová funkcionálnosť a Lerdorf prepracoval PHP do novej a omnoho bohatšej implementácie. V júni roku 1995 Lerdorf uverejnil zdrojové kódy k PHP a umožnil tak vývojárom z celého sveta aby používali PHP podľa vlastných potrieb. Týmto krokom taktiež umožnil užívateľom podieľať sa na odstraňovaní chýb a všeobecnom zlepšovaní PHP.

### 2.3.2 Súčasnosť PHP

PHP od čias svojho vzniku prešlo pomerne dlhú vývojovú cestu, ktorá obnášala mnoho zmien. Pri prechodoch na vyššiu verziu sa často stávalo, že aplikácie navrhnuté v staršej verzii prestali fungovať, keďže sa často krát s príchodom novej verzie pomerne zásadne menila špecifikácia PHP. Celkovo prešlo PHP piatimi majoritnými verziami, pričom väčšina z nich obsahovala niekoľko minoritných verzií, ktoré prinášali menšie zlepšenia a často aj opravy chýb. Aktuálne poslednou stabilnou oficiálne vydanou verzou je verzia

5.4.0, vydaná v marci 2012. Verzia priniesla vstavaný web server ako aj mnohé zlepšenia existujúcich funkcionalít spolu so zvýšením výkonnosti a znížením pamäťových nárokov.

### 2.3.3 Syntax PHP

PHP vykonáva iba kód oddelený od ostatnej časti dokumentu značkami `<?php` na začiatku a `?>` na konci PHP kódu. Akýkoľvek kód zapísaný mimo týchto značiek PHP nevykoná a je webovým prehliadačom zobrazený ako obyčajný text. Veľkou výhodou PHP je jeho možnosť byť súčasťou klasických HTML dokumentov. Jediným rozdielom je, že takýto dokument musí mať koncovku `.php`. Ak by mal súbor koncovku `.html`, PHP prekladač by ho bral ako čistý HTML dokument a aj v prípade zapísaného PHP kódu v takomto dokumente, by tento nebol webovým serverom vykonaný.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>PHP Test</title>
  </head>
  <body>
    <?php
    echo 'Hello World';
    ?>
  </body>
</html>
```

Uvedená jednoduchá ukážka vnoreného PHP do HTML kódu by po spracovaní webovým serverom vygenerovala čistý HTML kód, ktorý by vyzeral nasledovne.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>PHP Test</title>
  </head>
  <body>
    Hello World
  </body>
</html>
```

## 2.4 MySQL

MySQL je vo svete najpoužívanejší relačný databázový systém. Jeho pomenovanie vzniklo spojením mena dcéry zakladateľa Michaela Wideniusa - My a skratkou SQL, ktorá je oficiálnou skratkou pre Structured Query Language, ktorý je štandardizovaným a veľmi rozšíreným jazykom pre prístup k databázam. MySQL v súčasnosti podporuje a ďalej vyvíja firma Oracle. MySQL databázy si našli svoje stabilné miesto hlavne pri tvorbe webových aplikácií, keďže umožňujú veľmi rýchlo a prehľadne ukladať aj veľké množstvá dát a následne s nimi pracovať. Dáta sú ukladané do tabuliek, pričom jeden záznam tvorí jeden riadok tabuľky. Medzi tabuľkami v MySQL databáze môžeme definovať rôzne vzťahy - relácie, na základne takzvaných kľúčov. Neprehliadnuteľnou výhodou MySQL je fakt, že sa jedná o open-source projekt, šírený pod GPL licenciou. To znamená, že kód MySQL je voľne šíriteľný a užívatelia ho môžu ľubovoľne upravovať. Medzi ďalšie výhody MySQL patrí určite aj cenová dostupnosť, nakoľko MySQL poskytuje svoju plne funkčnú verziu MySQL Community Server úplne zadarmo. V ponuke sú tiež komerčné platené licencie pod názvom MySQL Enterprise Server.

### 2.4.1 Syntax MySQL príkazov

Pri práci s MySQL databázou máme k dispozícii množstvo príkazov, ktoré nám z databázy vrátia požadovaný výsledok. Medzi najpoužívanejšie príkazy pri tvorbe webových aplikácií patria určite tieto:

#### 2.4.1.1 *CREATE TABLE*

Príkaz `CREATE TABLE` slúži na vytvorenie novej tabuľky v databáze.

```
CREATE TABLE nazov_tabulky
(
nazov_stlpca1 datovy_typ,
nazov_stlpca2 datovy_typ,
...
)
```

### 2.4.1.2 *INSERT*

Príkaz INSERT slúži na vkladanie nových dát do existujúcej tabuľky v databáze.

```
INSERT INTO nazov_tabulky  
VALUES (hodnota_stlpca1, hodnota_stlpca2, ...)
```

### 2.4.1.3 *UPDATE*

Príkaz UPDATE slúži na úpravu existujúceho záznamu v tabuľke.

```
UPDATE nazov_tabulky  
SET stlpec1=hodnota1, stlpec2=hodnota2, ...
```

### 2.4.1.4 *DELETE*

Príkaz DELETE slúži na vymazanie záznamu z tabuľky.

```
DELETE FROM nazov_tabulky  
WHERE stlpec=hodnota
```

### 2.4.1.5 *SELECT*

Príkaz SELECT slúži na výber dát uložených v tabuľke.

```
SELECT nazov_stlpca  
FROM nazov_tabulky
```

### 2.4.1.6 *WHERE*

Príkaz WHERE slúži ako doplnenie ostatných MySQL príkazov, ktorý umožňuje špecifikovať konkrétnu podmienku na hodnotu konkrétneho stĺpca. Tento príkaz sa vykoná len na tých riadkoch tabuľky, ktoré spĺňajú podmienku uvedenú za príkazom WHERE.

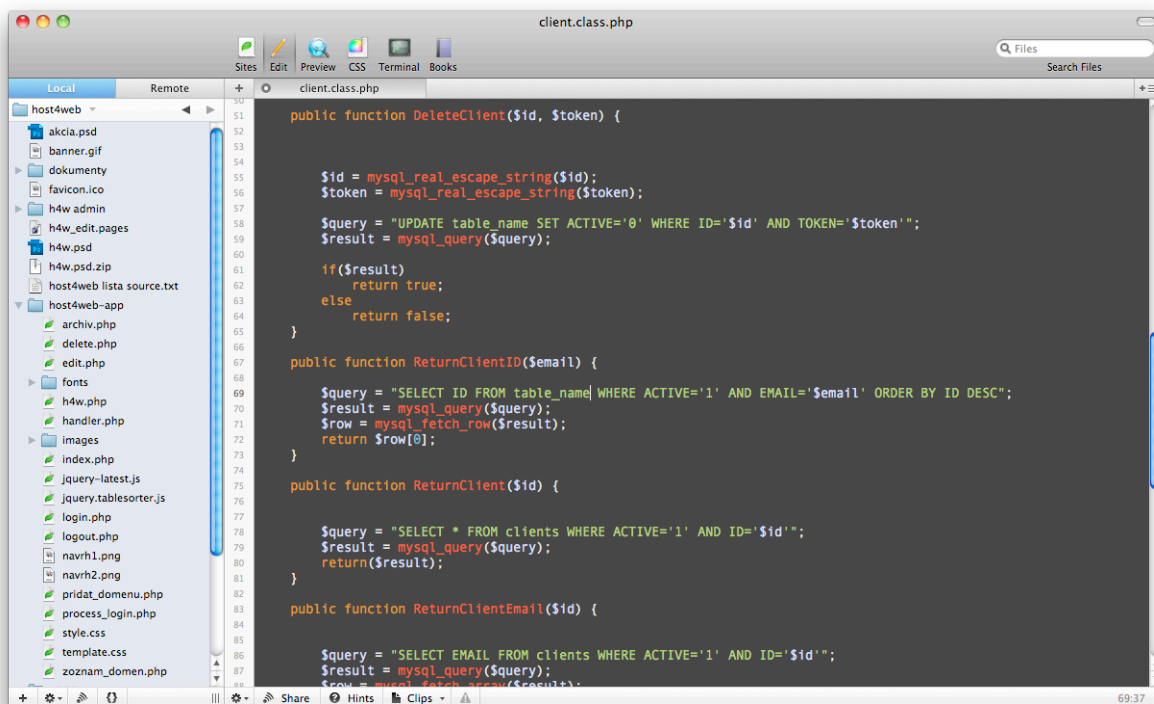
```
SELECT nazov_stlpca  
FROM nazov_tabulky  
WHERE nazov_stlpcaX=hodnota
```

### 3 POUŽITÉ PROGRAMOVÉ VYBAVENIE

Pri voľbe programov ktoré použijem pri tvorbe fakturačného systému som prihliadal na osobné preferencie a tak isto fakt, že vývoj prebiehal na platforme Mac OS X od firmy Apple. Použité programy sú pre bežných užívateľov platformy Windows zrejme pomerne neznáme, no drvivá väčšina užívateľov pracujúcich so systémom Mac ich určite dobre pozná. Obidva základné programy, ktoré som použil pri práci sú vyvinuté spoločnosťou Panic.

#### 3.1 Coda

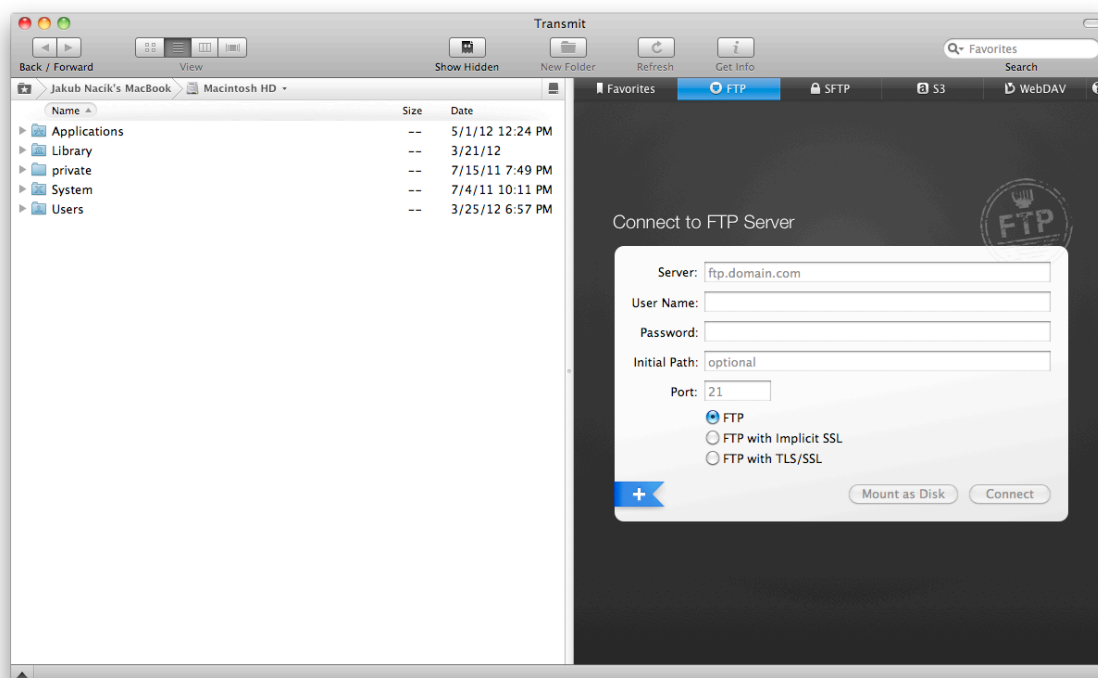
Editor Coda je jedným z produktov firmy Panic, ktorá sa špecializuje na tvorbu softvéru pre webových vývojárov na platforme Mac OS X. Coda ponúka pre vývojára takzvaný vývoj v jednom okne. To znamená že pri tvorbe webovej aplikácie nenúti jej používateľa preklikávať do iného okna, nakoľko mu poskytuje všetko, čo k práci potrebuje. Coda obsahuje textový editor, ktorý dokáže zvýrazňovať syntax širokého spektra programovacích jazykov a toto zvýrazňovanie je plne nastavovateľné užívateľom, ďalej ponúka zabudovaný FTP klient, interaktívny editor CSS a možnosť živého náhľadu rozpracovaného súboru v prehliadači. Prostredie Cody je veľmi intuitívne a prehľadné.



Obr. 1 Prostredie editoru Coda

## 3.2 Transmit

Transmit je FTP klient pre operačný systém Mac OS X a je rovnako ako Coda produkt firmy Panic. Transmit ponúka veľmi prehľadné používateľské rozhranie a možnosť spravovať aj veľké množstvo FTP kont. Umožňuje jednotlivé FTP kontá pripojiť tak, že sa v operačnom systéme chovajú ako ďalší pevný disk, takže je možné k súborom uloženým na FTP pristupovať priamo cez systém, nie iba cez okno programu. Tak isto umožňuje byť v jednom čase pripojený na viaceré FTP kontá a prepínať medzi nimi vo forme záložiek, aké poznáme z napríklad webových prehliadačov.



Obr. 2 Rozhranie FTP klienta Transmit

## **II. PRAKTICKÁ ČÁST**

## 4 REALIZÁCIA APLIKÁCIE

Mojou úlohou bolo navrhnúť nový proces objednávaní webhostingových služieb, ktorý by zautomatizoval čo najviac krokov v procese zriadení novej služby, ako aj v procese predlžovania platnosti už existujúcej služby. Bolo potrebné navrhnúť jednotlivé kroky a premyslieť konkrétny postup na základe ktorého bude fungovať nový fakturačný systém. Medzi požiadavky na fakturačný systém patrili hlavne prehľadná evidencia a správa všetkých doménových aj webhostingových služieb, klientov, ktorí tieto služby využívajú, ďalej automatické generovanie predfaktúr a odosielanie elektronickou poštou klientovi ako aj generovanie ostrých faktúr po prijatej platbe a ich opätovné doručenie klientovi. S tým úzko automatický dohľad systému nad splatnosťami predfaktúr a prípadne rozposielanie upomienok za neuhradené predafktúry elektronickou poštou. Ako doplnková funkcionalita sú základné štatistiky na základe dát uložených v databázach. Tieto umožňujú kedykoľvek zistiť základné súhrnné údaje o prevádzkovaných službách v reálnom čase.

### 4.1 Štruktúra systému

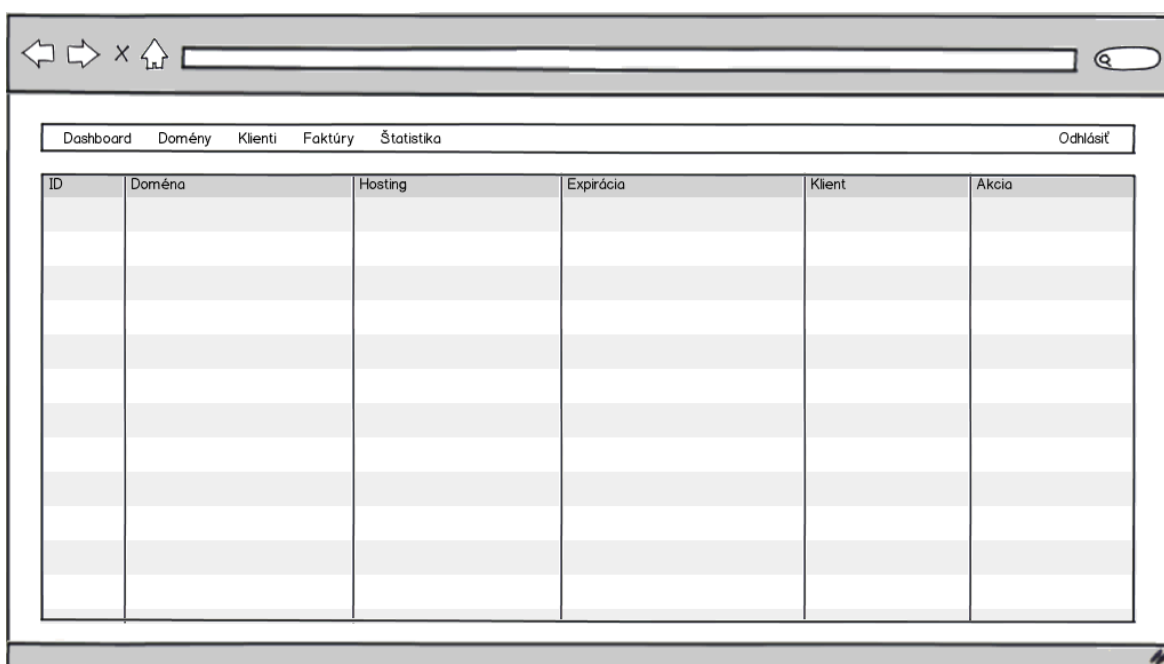
Štruktúra systému je celkovo rozdelená na 5 celkov, kde každý z nich spravuje príslušné údaje.

Rozdelenie sekcií systému:

- **Dashboard** - zobrazuje informácie o nových prijatých objednávkach, ktoré čakajú na schválenie pracovníkom technickej podpory ako aj informácie o službách, ktorým uplynul dátum splatnosti a je potrebné ich na serveri odstaviť.
- **Domény** - zoznam všetkých webhostingových služieb, ktoré sú aktívne využívané zákazníkmi. Každá webhostingová služba je viazaná na konkrétne doménové meno. V tejto sekcii má užívateľ prehľadne uvedené všetky kľúčové informácie o každej službe.
- **Faktúry** - evidencia vystavených faktúr. Umožňuje všetky faktúry plnohodnotne spravovať a poskytuje prehľad o prijatých a očakávaných platbách.
- **Klienti** - zoznam všetkých klientov, ktorí využívajú webhostingové služby. Zoznam obsahuje podrobné kontaktné a fakturačné údaje pre každého klienta.
- **Štatistika** - výpis základných štatistických údajov na základe systémových dát.

## 4.2 Návrh užívateľského administračného rozhrania

Pri návrhu rozhrania som bral do úvahy dve možnosti rozloženia prvkov na ploche stránky. Prvou možnosťou bolo použiť vertikálne hlavné menu na ľavom okraji stránky a pravú centrálnu časť pre prácu s údajmi. Druhou možnosťou bolo použiť horizontálne menu na hornej časti stránky a pod ním centrálnu časť na prácu s údajmi. Priklonil som sa k druhej možnosti, nakoľko som videl potenciál v možnosti spraviť šablónu stránky dynamickú na šírku a vždy tak využiť maximum dostupnej šírky okna webového prehliadača.



Obr. 3 Návrh štruktúry administračného rozhrania

Štruktúra systému je takmer vo všetkých sekciách veľmi podobná. Úlohou každej sekcie je prehľadne zoskupiť množstvo informácií v nej a čo možno najrýchlejšie a najjednoduchšie umožniť pracovníkovi nájsť potrebnú informáciu. Preto sú sekcie tvorené prevažne rozsiahlou tabuľkou dát, ktorá v jednotlivých stĺpcoch umožňuje rýchly prístup k najdôležitejším informáciám o konkrétnom zázname, bez nutnosti jeho rozkliknutia. Nad súhrnnou tabuľkou má vždy užívateľ možnosť prídania nového záznamu do príslušnej sekcie. Úplne posledný stĺpec tabuľky na pravej strane vždy slúži ako stĺpec s akciami, akými sú hlavne editácia a vymazanie záznamu, ale aj iné, špecifické akcie pre jednotlivé sekcie systému.

Farebná kombinácia administratívneho systému bola zosúladená s firemnými farbami, v duchu ktorých sa nesie aj rozhranie hlavnej webovej stránky spoločnosti.

ID	Doména	Hosting	Vytvorené	Obdobie	Hosting do	Doména do	Klient	Akcia
120068	esgun.sk	MiNi	25.05.2012	Rok	25.05.2013	21.07.2012	ESGun s.r.o.	<a href="#">Upraviť</a> <a href="#">Vymazať</a>
120067	solarastronomy.sk	MiNi	25.05.2012	Rok	25.05.2013	21.07.2012	Klára Baloghová	<a href="#">Upraviť</a> <a href="#">Vymazať</a>
120066	sykoraelektrodom.sk	STANDART	25.05.2012	Rok	25.05.2013	25.05.2013	Sykora Elektrodom	<a href="#">Upraviť</a> <a href="#">Vymazať</a>
120065	autodielymgzillina.sk	STANDART	25.05.2012	Rok	25.05.2013	25.05.2013	Autodiely MG	<a href="#">Upraviť</a> <a href="#">Vymazať</a>
120064	akyo.sk	Iba doména	24.05.2012	Rok	24.05.2013	-	AKR1 s.r.o.	<a href="#">Upraviť</a> <a href="#">Vymazať</a>
							Bc.Zuzana Maleriziová	<a href="#">Upraviť</a>

Obr. 4 Ukážka zrealizovaného rozhrania fakturačného systému

Ako možno vidieť z priloženého obrázku, rozhranie väčšinou pracuje s takým množstvom dát, ktoré presahuje maximálnu zobrazovanú výšku na bežných monitoroch. To núti užívateľa často rolovať smerom nadol, aby sa dostal k hľadaným údajom. Keď však chce po zrolovaní do nižších častí stránky prejsť do inej sekcie, aby sa dostal k hlavnému menu, ktoré je na úplnom vrchu rozhrania, musel by užívateľ opäť zbytočne rolovať smerom nahor. Z toho dôvodu som sa rozhodol hlavné menu spraviť fixné a zobrazené vždy na vrchu stránky, takže aj keď dôjde k rolovaniu smerom nadol, menu ostane na svojej pozícii a uľahčí tak prechod do inej sekcie systému. Túto vlastnosť som docielil prídelením fixnej pozície elementu hlavného menu v CSS.

HTML kód hlavního menu:

```
<div class="menu">
  <ul class="lave">
    <a href="/"><li>
      Dashboard
    </li></a>
    <a href="/domains"><li>
      Domény
    </li></a>
    <a href="/invoices"><li>
      Faktúry
    </li></a>
    <a href="/clients"><li>
      Klienti
    </li></a>
    <a href="/stats"><li>
      štatistika
    </li></a>
  </ul>
  <ul class="prave">
    <a href="/logout"><li>
      Odhlásiť
    </li></a>
  </ul>
</div>
```

Element DIV s triedou menu má pridelený CSS kód s nastavením fixnej pozície:

```
.menu {
  font-family:"Droid Sans";
  font-size:10px;
  font-weight:300;
  text-transform:uppercase;
  text-shadow:none;
  width:100%;
  background-color:#000;
  opacity: 0.835;
  height:39px;
  position:fixed;
  left:0;
  top:0;
  z-index:8000;
}
```

### 4.3 Návrh štruktúry databáz

Návrh databázovej štruktúry patrí medzi veľmi dôležité súčasti v procese tvorby webových aplikácií a je potrebné venovať mu dostatočnú pozornosť. Je dôležité dbať na správnu logiku rozloženia a prepojenia jednotlivých tabuliek, ako aj na správny výber dátových typov v jednotlivých stĺpcoch tabuliek. Zle navrhnutá databázová štruktúra sa môže veľmi nepriaznivo prejaviť na výkonnosti webovej aplikácie. S narastajúcim množstvom dát v tabuľkách sa dôležitosť databázovej štruktúry prejavuje oveľa zreteľnejšie.

Databáza fakturačného systému obsahuje celkovo 8 tabuliek, pričom samotná funkčnosť aplikácie využíva 6 tabuliek.

#### 4.3.1 Tabuľka "users"

Tabuľka "users" ukladá dáta o užívateľoch systému - pracovníkoch technickej podpory.

Tab. 1 users

Field	Type	Collation	Attributes	Null	Default	Extra
<b>id</b>	int(11)			No	None	AUTO_INCREMENT
<b>username</b>	varchar(30)	utf8_general_ci		No	None	
<b>password</b>	varchar(128)	utf8_general_ci		No	None	
<b>email</b>	varchar(255)	utf8_general_ci		No	None	

- **Stĺpec "id"** je primárnym kľúčom tabuľky "users", slúži ako jedinečný identifikátor záznamov v tabuľke a jeho hodnota sa automaticky inkrementuje s každým novým záznamom.
- **Stĺpec "username"** ukladá prihlasovacie meno každého užívateľa. Vďaka kombinácii prihlasovacieho mena a hesla, je možné autorizovať užívateľov pre vstup do systému.
- **Stĺpec "password"** ukladá hash užívateľského hesla. Hash je vygenerovaný 256 bytovým SHA algoritmom.
- **Stĺpec "email"** ukladá kontaktný e-mail pre každého užívateľa, ktorý slúži napríklad pre kontaktovanie systémovým administrátorom.

### 4.3.2 Tabuľka "clients"

Tabuľka "clients" ukladá dáta o firemných zákazníkoch. Každý zákazník tu má uložené jeho kontaktné a fakturačné údaje.

Tab. 2 clients

Field	Type	Collation	Attributes	Null	Default	Extra
<b>ID</b>	int(11)			No	None	AUTO_INCREMENT
<b>NAME</b>	varchar(60)	utf8_general_ci		No	None	
<b>WEBADMIN</b>	varchar(30)	utf8_general_ci		No	None	
<b>ADDRESS</b>	varchar(60)	utf8_general_ci		No	None	
<b>TOWN</b>	varchar(60)	utf8_general_ci		No	None	
<b>ZIP</b>	varchar(10)	utf8_general_ci		No	None	
<b>ICO</b>	varchar(10)	utf8_general_ci		No	None	
<b>DIC</b>	varchar(10)	utf8_general_ci		No	None	
<b>IC_DPH</b>	varchar(12)	utf8_general_ci		No	None	
<b>EMAIL</b>	varchar(60)	utf8_general_ci		No	None	
<b>PHONE</b>	varchar(30)	utf8_general_ci		No	None	
<b>CONTACT_NAME</b>	varchar(60)	utf8_general_ci		No	None	
<b>SKNIC</b>	varchar(10)	utf8_general_ci		No	None	
<b>NOTE</b>	text	utf8_general_ci		No	None	
<b>TOKEN</b>	varchar(50)	utf8_general_ci		No	None	
<b>ACTIVE</b>	tinyint(1)			No	None	

- **Stĺpec "ID"** je primárnym kľúčom tabuľky "clients", slúži ako jedinečný identifikátor záznamov v tabuľke a jeho hodnota sa automaticky inkrementuje s každým novým záznamom.
- **Stĺpec "NAME"** uchováva meno klienta v prípade fyzickej osoby alebo názov firmy v prípade právnickej osoby.
- **Stĺpec "WEBADMIN"** ukladá login užívateľa do externého administračného systému webhostingových služieb, ktorý poskytuje prevádzkovateľ servera. Tento login je každému novému klientovi vytvorený pri prvej objednávke a následne si všetky ďalšie objednávané služby klient spravuje pod spoločným loginom.
- **Stĺpec "ADDRESS"** ukladá ulicu a číslo domu z fakturačnej adresy klienta.
- **Stĺpec "TOWN"** ukladá mesto z fakturačnej adresy klienta.
- **Stĺpec "ZIP"** ukladá PSČ z fakturačnej adresy klienta.

- **Stĺpec "ICO"** ukladá identifikačné číslo obchodníka v prípade, že je klientom právnická osoba.
- **Stĺpec "DIC"** ukladá jeho daňové identifikačné číslo v prípade, že je klientom právnická osoba.
- **Stĺpec "IC\_DPH"** ukladá identifikačné číslo pre DPH v prípade, že je klientom právnická osoba registrovaná ako platiteľ dane z pridanej hodnoty.
- **Stĺpec "EMAIL"** ukladá kontaktnú e-mailovú adresu klienta. Na túto adresu sú doručované všetky elektronické správy z fakturačného systému.
- **Stĺpec "PHONE"** ukladá kontaktné telefónne číslo na klienta.
- **Stĺpec "CONTACT\_NAME"** ukladá meno kontaktnej osoby na klienta. Opodstatnenie má hlavne pri právnických osobách.
- **Stĺpec "SKNIC"** ukladá identifikátor pre registráciu .sk domén, ktorý pridáva na základe žiadosti správcu slovenskej národnej domény, spoločnosť SK-NIC, a.s.
- **Stĺpec "NOTE"** ukladá prípadnú poznámku pracovníka k danému klientovi.
- **Stĺpec "TOKEN"** slúži ako bezpečnostný identifikátor jednotlivých akcií. Jedná o náhodný 128 bitový hash, pomocou ktorého sa overuje oprávnenie jednotlivých operácií, akými je napríklad vymazanie klienta z databázy.
- **Stĺpec "ACTIVE"** ukladá hodnotu 1 alebo 0, podľa toho, či sa jedná o klienta aktívne využívajúceho niektorú zo služieb.

### 4.3.3 Tabuľka "domains"

V tejto tabuľke sú uložené dáta o všetkých webhostingových a doménových službách, ktoré sú momentálne využívané klientmi. Každá služba tvorí jeden záznam, pri ktorom sú uchované všetky dôležité informácie.

Tab. 3 domains

Field	Type	Collation	Attributes	Null	Default	Extra
<b>ID</b>	int(11)			No	None	AUTO_INCREMENT
<b>DOMAIN</b>	varchar(63)	utf8_general_ci		No	None	
<b>TLD</b>	varchar(10)	utf8_general_ci		No	None	
<b>TLD_ID</b>	int(11)			No	None	
<b>HOSTING</b>	tinyint(1)			No	None	
<b>HOSTINGONLY</b>	tinyint(1)			No	0	
<b>DATE</b>	date			No	None	
<b>PERIOD</b>	tinyint(1)			No	None	
<b>EXPIRATION</b>	date			No	None	
<b>DOM_EXPIRATION</b>	date			No	None	
<b>CLIENT_ID</b>	int(11)			No	None	
<b>TOKEN</b>	varchar(50)	utf8_general_ci		No	None	
<b>CONFIRMED</b>	tinyint(1)			No	0	
<b>ACTIVE</b>	tinyint(1)			No	None	

- **Stĺpec "ID"** je primárnym kľúčom tabuľky "domains", slúži ako jedinečný identifikátor záznamov v tabuľke a jeho hodnota sa automaticky inkrementuje s každým novým záznamom.
- **Stĺpec "DOMAIN"** ukladá doménové meno bez koncovky.
- **Stĺpec "TLD"** ukladá koncovku k doménovému menu.
- **Stĺpec "TLD\_ID"** ukladá ID koncovky, ktoré je definované v tabuľke "tlds", ktorá eviduje zoznam všetkých dostupných domén v ponuke.
- **Stĺpec "HOSTING"** ukladá ID webhostingového balíka z ponuky. Všetky webhostingové balíky sú uložené v tabuľke "services".
- **Stĺpec "HOSTINGONLY"** ukladá informáciu, či sa jedná iba o webhostingovú službu, alebo firma spravuje aj klientovu doménu.
- **Stĺpec "DATE"** ukladá dátum vytvorenia a teda aj začatia platnosti služby.

- **Stĺpec "PERIOD"** ukladá informáciu, na aké obdobie má klient predplatenú službu.
- **Stĺpec "EXPIRATION"** ukladá dátum, dokedy je predplatená webhostingová služba.
- **Stĺpec "DOM\_EXPIRATION"** ukladá dátum, dokedy je predplatená doména k webhostingu.
- **Stĺpec "CLIENT\_ID"** ukladá ID klienta z tabuľky "clients", ktorý je vlastníkom služby.
- **Stĺpec "TOKEN"** slúži ako bezpečnostný identifikátor jednotlivých akcií. Jedná o náhodný 128 bitový hash, pomocou ktorého sa overuje oprávnenie jednotlivých operácií, akými je napríklad vymazanie služby z databázy.
- **Stĺpec "CONFIRMED"** ukladá informáciu, či bola služba overená pracovníkom. Overenie služby prebieha jednorázovo, vždy pri novej objednávke.
- **Stĺpec "ACTIVE"** ukladá informáciu, či sa jedná o aktívnu službu, alebo službu už deaktivovať - po expirácii.

#### 4.3.4 Tabuľka "invoices"

Eviduje všetky predfaktúry a faktúry spolu s ich vygenerované systémom kľúčovými informáciami akou je napríklad evidencia splatnosti faktúr.

Tab. 4 invoices

Field	Type	Collation	Attributes	Null	Default	Extra
VS	int(6)			No	None	
<u>ORDER_NO</u>	int(11)			No	None	
CLIENT_ID	int(11)			No	None	
TYPE	varchar(2)	utf8_general_ci		No	PF	
DATEFROM	date			No	None	
DATETO	date			No	None	
PAYMENT	varchar(10)	utf8_slovak_ci		No	None	
TOKEN	varchar(50)	utf8_general_ci		No	None	

- **Stĺpec "VS"** ukladá variabilný symbol - číslo ostrej faktúry, ktorý sa zhoduje s číslami faktúr evidovaných vo firemnom účtovníctve. Generuje sa spolu s ostrou faktúrou, po zaevidovaní úrady predfaktúry.
- **Stĺpec "ORDER\_NO"** je primárnym kľúčom tabuľky "invoices" a jeho hodnota sa vždy zhoduje s ID služby v tabuľke "domains", pre ktorú je daná faktúra vystavená.
- **Stĺpec "TYPE"** ukladá informáciu, či sa jedná o predfaktúru čakajúcu na úhradu alebo o už uhradenú ostrú faktúru.
- **Stĺpec "DATEFROM"** ukladá dátum vystavenia faktúry.
- **Stĺpec "DATETO"** ukladá dátum splatnosti faktúry.
- **Stĺpec "PAYMENT"** ukladá informáciu o počte dní do splatnosti, prípadne počte dní po splatnosti faktúry. V prípade uhradenej faktúry nesie hodnotu "X".
- **Stĺpec "TOKEN"** slúži ako bezpečnostný identifikátor jednotlivých akcií. Jedná o náhodný 128 bitový hash, pomocou ktorého sa overuje oprávnenie jednotlivých operácií, akými je napríklad vymazanie faktúry z databázy.

#### 4.3.5 Tabuľka "invoices\_subjects"

Je podradenou tabuľkou k tabuľke "invoices" a eviduje položky jednotlivých faktúr.

Tab. 5 invoices\_subjects

Field	Type	Collation	Attributes	Null	Default	Extra
<b>ID</b>	int(11)			No	None	AUTO_INCREMENT
<b>ORDER_NO</b>	int(11)			No	None	
<b>SERVICE</b>	varchar(100)	utf8_slovak_ci		No	None	
<b>DATEFROM</b>	varchar(12)	utf8_general_ci		No	None	
<b>DATETO</b>	varchar(12)	utf8_general_ci		No	None	
<b>PRICE</b>	float			No	None	
<b>PRICEVAT</b>	float			No	None	

- **Stĺpec "ID"** je primárnym kľúčom tabuľky "invoices\_subjects", slúži ako jedinečný identifikátor záznamov v tabuľke a jeho hodnota sa automaticky inkrementuje s každým novým záznamom.
- **Stĺpec "ORDER\_NO"** ukladá hodnotu primárneho kľúča "ORDER\_NO" z tabuľky "invoices" a tým nesie informáciu, ku ktorej faktúre je položka priradená.
- **Stĺpec "SERVICE"** ukladá názov položky.
- **Stĺpec "DATEFROM"** ukladá dátum, od ktorého platí daná položka.
- **Stĺpec "DATETO"** ukladá dátum, do ktorého platí daná položka.
- **Stĺpec "PRICE"** ukladá cenu položky bez DPH.
- **Stĺpec "PRICEVAT"** ukladá cenu položky s DPH.

#### 4.3.6 Tabuľka "pricelist"

Jedná sa o cenník všetkých aktuálne poskytovaných služieb. Ceny z tejto tabuľky sa pri generovaní faktúr priradujú k jednotlivým položkám a tak je možné vypočítať celkovú cenu na faktúrach.

Tab. 6 pricelist

Field	Type	Collation	Attributes	Null	Default	Extra
<b>ID</b>	int(11)			No	None	AUTO_INCREMENT
<b>NAME</b>	varchar(30)	utf8_general_ci		No	None	
<b>PERIOD</b>	tinyint(1)			No	None	
<b>PRICE</b>	float			No	None	
<b>PRICEVAT</b>	float			No	None	

- **Stĺpec "ID"** je primárnym kľúčom tabuľky "pricelist", slúži ako jedinečný identifikátor záznamov v tabuľke a jeho hodnota sa automaticky inkrementuje s každým novým záznamom.
- **Stĺpec "NAME"** ukladá názov služby.
- **Stĺpec "PERIOD"** ukladá obdobie, na ktoré je daná služba dostupná za uvedenú cenu.
- **Stĺpec "PRICE"** ukladá cenu služby za príslušné obdobie bez DPH.
- **Stĺpec "PRICEVAT"** ukladá cenu služby za príslušné obdobie s DPH.

## 4.4 Funkčné pozadie systému

### 4.4.1 Objektové PHP

Keďže je predpoklad, že sa systém bude do budúcnosti rozširovať, pri jeho návrhu som sa snažil dbať o čo najlepšiu robustnosť a modularitu, aby si prípadné zásahy do systému nevyžadovali jeho masívne úpravy. Z toho dôvodu som použil objektovo orientované PHP a základnú funkčnú časť postavil na triedach prislúchajúcich k jednotlivým sekciám. Celkovo systém v jeho súčasnej podobe využíva 5 tried a každá obsahuje potrebné funkcie, ktoré sa s dátami v danej triede vykonávajú.

Vytvorené triedy v PHP:

- **Client** - trieda obsahuje funkcie pracujúce s evidenciou klientov, ich pridávaním, mazaním a vypisovaním.
- **Domain** - trieda obsahuje funkcie pracujúce s evidenciou domén a k nim prislúchajúcich webhostingových služieb.

- **Email** - trieda obsahuje funkcie, ktoré zabezpečujú odosielanie elektronickej pošty klientom. Hlavne správy s predafktúrami, faktúrami ako aj výzvami na blížiacu sa expiráciu preplatených služieb.
- **Invoice** - trieda obsahuje funkcie na generovanie predfaktúr a faktúr do PDF ako aj obslužné funkcie pre správu faktúr a evidenciu platieb. Na generovanie PDF súborov je použitá PHP knižnica HTML2PDF. Tá vytvára PDF dokumenty podľa predvytvorenej HTML šablóny.
- **Stats** - trieda obsahuje funkcie pre výpis základných štatistík v systémovej sekcii Štatistika.

#### 4.4.2 Načítavanie podstránok pomocou funkcie include()

Pre každú sekciu systému existuje na serveri jedna podstránka. Všetky podstránky majú spoločnú hlavnú stránku **index.php**. V prípade zavolania odkazu na niektorú z podstránok sa tento volá v tvare **index.php?page=nazov\_sekcie**. V súbore `index.php` je skript, ktorý cez funkciu `$_GET` načíta z adresy hodnotu parametra `page`, overí existenciu požadovanej podstránky v preddefinovanom poli podstránok, vyhledá v príslušnej zložke na serveri súbor s týmto názvom a ten pomocou funkcie **include()** vloží do súboru `index.php` a zobrazí výsledok v podobe rozkliknutej podstránky. Ak sa požadovaná stránka nenájde alebo parameter `page` neobsahuje žiadnu hodnotu, skript načíta úvodnú stránku `dashboard.php`.

```
...
$page = $_GET['page'];
$action = $_GET['action'];

if (isset($page))
{
    $pages = array('dashboard',
                  'domains',
                  'clients',
                  'invoices',
                  'stats');
    $pocet = count($pages);

    for($i = 0; $i <= $pocet; $i++)
    {
        if($page == $pages[$i]) {
            include "../pages/" . $page . ".php";
        }
    }
} else {
    include "../pages/dashboard.php";
}
...
```

Ako je možné vidieť v ukážke skriptu, okrem parametra page načítava cez funkciu `$_GET` aj parameter action. Ten slúži na špecifikovanie zobrazenia, ktoré sa má na danej podstránke vykonať. Napríklad v prípade pridávania novej domény do databázy by zavolaný odkaz vyzeral nasledovne: `index.php?page=domains&action=new-domain`.

Aby nedochádzalo k načítavaniu súborov podstránok mimo súboru `index.php`, použil som jednoduchý systém ochrany načítavaných súborov. Na začiatku súboru `index.php` je definovaná konštanta `protectInclude`, ktorej je priradená hodnota 1.

```
define('protectInclude', '1');
...
```

A následne overenie či bola definovaná konštanta `protectInclude` v každom súbore, ktorý je načítavaný ako podsúbor k hlavnému `index.php`.

```
<? if (!defined('protectInclude')) { die('Error!'); exit;
} ?>
...
```

Ak nebola definovaná konštanta `protectInclude`, vypíše sa na obrazovku `Error!` a zastaví sa vykonávanie skriptu funkciou **die()**.

#### 4.4.3 Pekné URL pomocou `mod_rewrite`

Vývoj v oblasti webových aplikácií ide stále dopredu a jednou zo stále obľúbenejších noviniek je používanie pekných odkazov v adresnom riadku. Pri bežných webových stránkach to má pozitívny vplyv na optimalizáciu pre vyhľadávače. Pri webových aplikáciách ide skôr o lepšiu prehľadnosť aplikácie a akcií, ktoré v nej vykonávame.

Základným princípom tejto technológie je nahradenie odkazov v tvare napríklad `index.php?page=domains&action=new-domain` za odkaz v tvare `domains/new-domain`, ktorý je na prvý pohľad oveľa zreteľnejší a prehľadnejší. Toto nám umožňuje súbor **.htaccess**, umiestnený v koreňovom adresári na serveri, v ktorom je možné nakonfigurovať akým spôsobom sa majú parametre z "krajšej" varianty odkazu dopĺňať na pozadí aplikácie do pôvodného tvaru.

Pre tento fakturačný systém vyzerá súbor `.htaccess` nasledovne:

```
RewriteEngine on
RewriteBase /

RewriteRule ^login$ /login.php [L]
RewriteRule ^login\/$ /login.php [L]
RewriteRule ^logout$ /logout.php [L]
RewriteRule ^logout\/$ /logout.php [L]

RewriteRule ^show-pdf /pages/show-pdf.php [L]

RewriteRule ^([\-,a-z,0-9]+)/([\-,a-z,0-9+]$
/?page=$1&action=$2 [L]
RewriteRule ^([\-,a-z,0-9+)/([\-,a-z,0-9+)\/$
/?page=$1&action=$2 [L]
```

#### 4.4.4 Automatické operácie

Pri fungovaní fakturačného systému sú určité operácie, ktoré si nevyžadujú zásah pracovníka a môžu prebiehať úplne samostatne. Na to, aby mohla nejaká akcia vo webovej aplikácii prebehnúť je potrebné vykonať príslušný skript. To nám umožňuje funkcia časovaného spúšťania skriptov s názvom CRON. CRON zabezpečuje priamo webový server a jeho konfigurácia prebieha cez administračné rozhranie webhostingu.

čas	<input type="text"/>
typ	http link
http link/skript	<input type="text"/>
poslať výstup na email	nikdy
emailová adresa	<input type="text"/>
<input type="button" value="vytvoriť"/>	

Obr. 5 Nastavenie časovača skriptov CRON

Nastavenie CRONu je pomerne jednoduché. Je potrebné zvoliť čas, v ktorom sa má požadovaný skript vykonávať. Čas sa nastavuje v štandardnom crontab unixovom formáte [minúty] [hodiny] [deň v mesiaci] [mesiac] [deň v týždni]. Následne je potrebné zadať cestu k skriptu ktorý sa má vykonávať. Ďalej môžeme zvoliť, či sa ma prípadný výstup skriptu odoslať na zvolenú e-mailovú adresu.

Fakturačný systém vykonáva tieto automatizované operácie:

- **odpočítavanie dní do splatnosti faktúry** - skript, ktorý znižuje hodnotu o 1 v tabuľke "invoices" a stĺpci "PAYMENT" pri všetkých doposiaľ neuhradených predfaktúrach
- **rozposielanie upomienok pri predfaktúrach po splatnosti** - skript, ktorý prechádza dni do splatnosti na jednotlivých predfaktúrach a pokiaľ je daná predfaktúra po splatnosti, pošle na kontaktnú e-mailovú adresu upomienku
- **predlžovanie služieb s blížiacou sa expiráciou** - ak je dátum platnosti služby menej ako 30 dní od aktuálneho dátumu, vygeneruje skript predfaktúru na ďalšie obdobie a zašle ju klientovi na kontaktnú e-mailovú adresu

#### 4.4.5 Prihlasovanie užívateľov

Keďže sú všetky dáta ukladané v databáze systému internými firemnými dátami je prístup k nim samozrejme povolený len oprávneným osobám. Preto je nutné sa pred vstupom do systému prihlásiť pomocou svojich prihlasovacích údajov, konkrétne kombináciou loginu a hesla, ktoré sú uložené v MySQL databáze v tabuľke "users".

Overovanie užívateľských dát a prihlasovanie do systému zabezpečuje funkcia **Login()**.

```
function Login($username, $password)
{
    $password = hash('sha256', $username . hash('sha256',
$password));

    include './includes/connection.php';
    $query = "SELECT password FROM users WHERE username =
'$username'";
    $result = mysql_query($query);
    $userData = mysql_fetch_array($result, MYSQL_ASSOC);

    if(mysql_num_rows($result) < 1) {
        return false;
    }

    if($password != $userData['password']) {
        return false;
        mysql_close();
    } else {
        ValidateUser($username);
        return true;
        mysql_close();
    }
}
```

Funkcia má ako vstupné parametre premenné \$username a \$password, ktoré užívateľ vyplní v prihlasovacom formulári. Z premennej \$password vytvorí hash, ktorý porovná s hashom uloženým v databáze. Ak sa tieto údaje zhodujú, je zavolaná funkcia **ValidateUser()**, ktorá do \$\_SESSION uloží informáciu o tom, že daný užívateľ je oprávnený vstúpiť do systému a pracovať s ním.

```
function ValidateUser($username) {
    session_regenerate_id();
    $_SESSION['valid'] = 1;
    $_SESSION['username'] = $username;
}
```

Pre overenie, či je užívateľ prihlásený a teda oprávnený vstúpiť do systému slúži funkcia **isLoggedIn()**, ktorá vracia hodnotu TRUE alebo FALSE, podľa toho, či existuje `$_SESSION["valid"] = 1`. Táto funkcia je zavolaná na začiatku súboru `index.php` ako aj ostatných sekcií systému a v prípade že vráti hodnotu FALSE, je užívateľ presmerovaný na stránku s prihlásením.

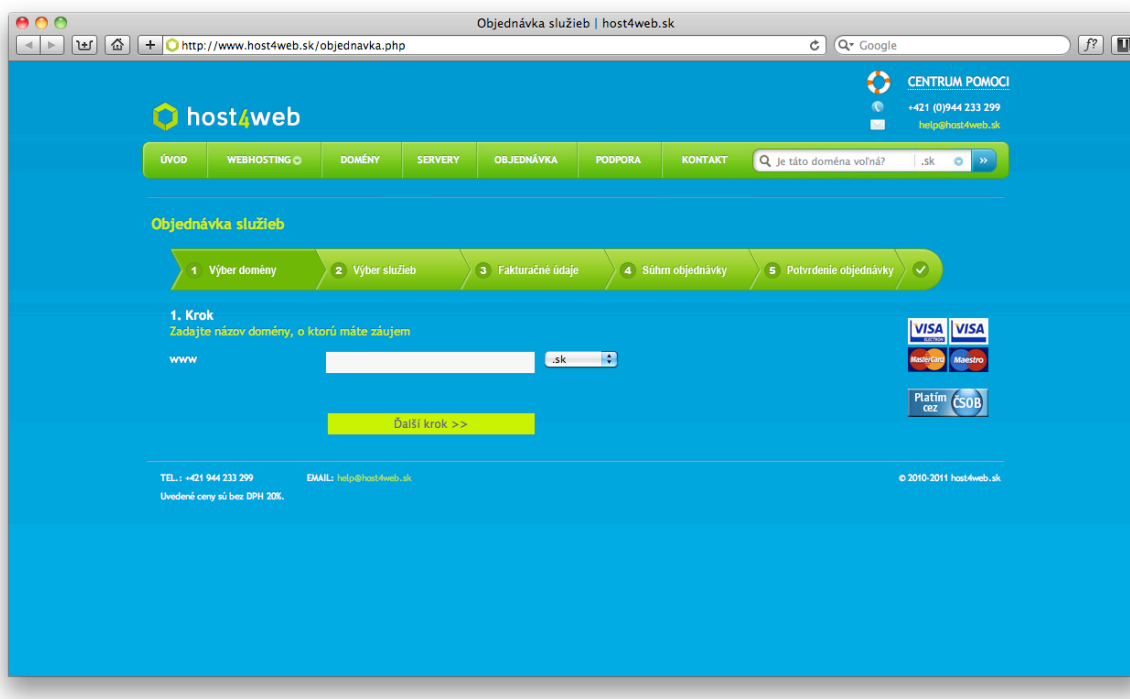
```
function isLoggedIn()  
{  
    if($_SESSION['valid'])  
        return true;  
    return false;  
}
```

## 4.5 Fungovanie systému v praxi

Fakturačný systém bol krátko po svojom vývoji nasadený do reálnej praxe. Svojou funkcionalitou zastrešuje kompletnú evidenciu klientov, služieb ktoré využívajú a evidenciu faktúr, na základe ktorej sú vedené aj firemné účtovné záznamy. S návrhom systému boli navrhnuté aj určité pracovné postupy, na základe ktorých celý systém funguje.

### 4.5.1 Objednávanie nových služieb

Súčasťou fakturačného systému je aj objednávkový formulár, ktorý je implementovaný do hlavnej firemnej webstránky. Cez tento formulár noví ale aj stáli klienti špecifikujú svoje požiadavky a odosielajú elektronickou formou objednávky služieb.



Obr. 6 Objednávkový formulár

Celý proces objednávania je rozdelený na 4 kroky. Každý krok je zameraný na iný typ dát, ktoré záujemcovia vyplňajú. Medzi jednotlivými krokmi je možné postupovať dopredu a dozadu. Aby sa údaje vyplnené v jednotlivých krokoch uchovávali, ukladajú sa po prechode na ďalší krok do premenných `$_SESSION`, kde sú uchované počas práce s webstránkou až do zatvorenia okna prehliadača.

Jednotlivé kroky objednávkového formulára sú nasledovné:

- **1. krok - špecifikácia domény** - keďže sa každá webhostingová služba viaže na nejakú doménu, zadáva sa táto doména pri každej objednávke aj v prípade, že si registráciu a správu samotnej domény zabezpečuje sám klient.
- **2. - výber webhostingového balíka služieb** - na výber sú konkrétne 3 balíky služieb, klient má možnosť výberu, o ktorý balík má záujem.
- **3. - kontaktné a fakturačné údaje** - klient vyplní údaje, na ktoré mu bude vygenerovaná faktúra za služby a kontaktné údaje, ktoré budú slúžiť na komunikáciu s firmou. Pre existujúcich klientov je tu možnosť vyplniť iba prihlasovacie meno do administratívneho rozhrania webhostingových služieb a

kontaktnú e-mailovú adresu. Ostatné údaje sa im doplnia automaticky na základe záznamu v databáze klientov.

- **4. súhrn objednávky** - prehľad všetkých údajov špecifikovaných v objednávke spolu s výpočtom celkovej ceny objednávky.

Po odoslaní objednávky sa všetky údaje zaznamenajú do databáze vykonaním funkcie **AddDomain()** z triedy Domain.

```
public function AddDomain($domain, $tld, $tldid, $hosting,
    $hostingonly, $date, $domaindate, $period, $userid)
{
    ...

    $query = "INSERT INTO domains (DOMAIN, TLD, TLD_ID,
    HOSTING, HOSTINGONLY, DATE, PERIOD, EXPIRATION,
    DOM_EXPIRATION, CLIENT_ID, TOKEN, ACTIVE)
    VALUES ('$domain', '$tld', '$tldid', '$hosting',
    '$hostingonly', '$date', '$period', '$expiration',
    '$domainexpiration', '$userid', '$token', '1')";

    $result = mysql_query($query);

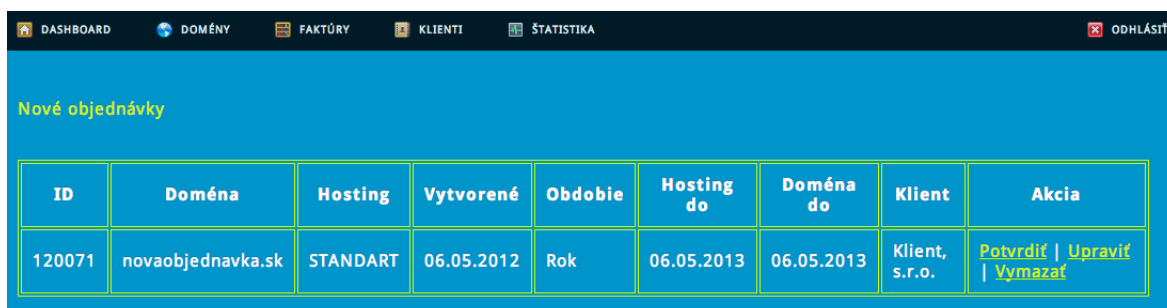
    if($result)
        return true;
    else
        return false;

}
```

Po odoslaní objednávky sa na e-mailovú adresu klienta odošle potvrdzujúca správa o prijatí novej objednávky a na administrátorskú e-mailovú adresu sa odošle informácia o prijatí novej objednávky.

#### 4.5.2 Potvrdenie objednávky pracovníkom

Vo fakturačnom systéme sú nové objednávky zobrazené priamo na dashboarde po prihlásení. Úlohou pracovníka je overiť údaje vyplnené v objednávke, zriadiť objednané služby na webovom serveri a prístupové údaje k nim doručiť klientovi.



The screenshot shows a dashboard with a navigation bar at the top containing 'DASHBOARD', 'DOMÉNY', 'FAKTÚRY', 'KLIENTI', 'ŠTATISTIKA', and 'ODHLÁSIŤ'. Below the navigation bar, the title 'Nové objednávky' is displayed. A table with 9 columns is shown, containing one row of data for a new order.

ID	Doména	Hosting	Vytvorené	Obdobie	Hosting do	Doména do	Klient	Akcia
120071	novaobjednavka.sk	STANDART	06.05.2012	Rok	06.05.2013	06.05.2013	Klient, s.r.o.	Potvrdiť   Upraviť   Vymazať

Obr. 7 Nová objednávka vo fakturačnom systéme

Na potvrdenie objednávky slúži akcia "Potvrdiť", ktorá je zobrazená pri každej novej objednávke. Po kliknutí na tlačidlo "Potvrdiť" sa zobrazí formulár, do ktorého pracovník doplní prístupové údaje do administračného rozhrania webhostingu a potvrdí objednávku. Po jej potvrdení sa vykoná niekoľko funkcií, ktoré upravia stav objednávky v databáze. Na základe špecifikácie služieb uvedených v zázname je vygenerovaná predfaktúra a jej PDF verzia doručená na e-mailovú adresu klienta.



The screenshot shows a confirmation form titled 'Zoznam domén'. It contains several input fields with pre-filled values and a confirmation button.

ID: 120071  
 Doména: novaobjednavka.sk  
 Klient: Klient, s.r.o.  
 E-mail: klient@klient.sk  
 WebAdmin: klient  
 Heslo: vygenerovane\_heslo  
 Potvrdiť objednávku

Obr. 8 Potvrdzovanie novej objednávky pracovníkom

### 4.5.3 Evidencia úhrady predfaktúry

Po potvrdení objednávky je vygenerovaná predfaktúra, ktorá sa následne zobrazuje v sekcii Faktúry a má nastavenú základnú splatnosť 7 dní. V prípade, že pracovník zaznamená e-mailovú notifikáciu z banky o prijatej platbe, má možnosť označiť danú predfaktúru ako uhradenú, k čomu slúži funkcia **Pay()**. Tá upraví záznam v databáze, zaeviduje úhradu, vygeneruje ostrú faktúru ako daňový doklad a tú odošle na e-mailovú adresu klientovi.

```
public function Pay($orderno, $vs) {  
  
    $datefrom = date("Y-m-d");  
    $dateto = strtotime(date("Y-m-d",  
strtotime($datefrom)) . " +7 day");  
    $dateto = date("Y-m-d", $dateto);  
  
    $query = "UPDATE invoices SET VS='$vs', TYPE='FA',  
DATEFROM='$datefrom', DATETO='$dateto', PAYMENT='X' WHERE  
ORDER_NO='$orderno'";  
    $result = mysql_query($query);  
  
    $this->GeneratePDF($orderno);  
  
    if($result)  
        return true;  
    else  
        return false;  
}
```

Po tejto akcii je už ďalší cyklus domény plne automatizovaný a pracovníci sa o nič nemusia starať, keďže systém sleduje expiráciu jednotlivých služieb a automaticky ich predlžuje na nové obdobie. Predĺžené služby sa objavia v zozname faktúr ako novo vystavené predfaktúry a úlohou pracovníkov je opäť len evidencia úhrady.

#### 4.5.4 Štatistiky

Funkcionalita systému je doplnená aj o sekciu Štatistika, kde je vždy možné vidieť aktuálne štatistické údaje dát v systéme. Okrem základných dát, ako napríklad počet domén, počet hostingov a ich rozdelenie podľa jednotlivých balíkov je sekcia so štatistikami doplnená aj o dva grafy zobrazujúce pomer jednotlivých webhostingových balíkov a pomer medzi spravovanými doménami a hostingovými službami. Pre tieto grafy som použil knižnicu grafov voľne dostupnú cez Google API. Grafy sú dynamicky generovaná JavaScriptom na základe vstupných dát.

```

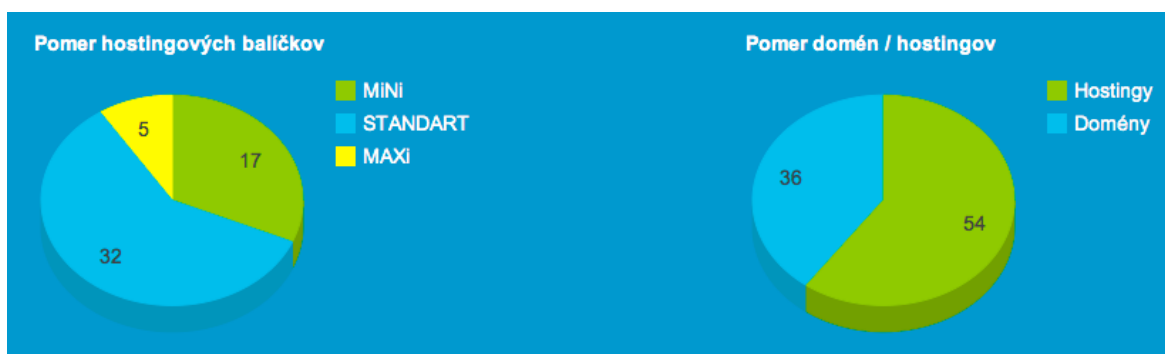
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>

<script type="text/javascript">
  google.load("visualization", "1",
{packages:["corechart"]});
  google.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Balíček');
    data.addColumn('number', 'Počet');
    data.addRows(3);
    data.setValue(0, 0, 'MiNi');
    data.setValue(0, 1, <? echo $Stats->ReturnMini(); ?>);
    data.setValue(1, 0, 'STANDART');
    data.setValue(1, 1, <? echo $Stats->ReturnStandart();
?>);
    data.setValue(2, 0, 'MAXi');
    data.setValue(2, 1, <? echo $Stats->ReturnMaxi(); ?>);

    var chart = new
google.visualization.PieChart(document.getElementById('hos
tingove_baliky'));
    chart.draw(data, {backgroundColor: '#0186c0',width:
500, height: 300, title: 'Pomer hostingových balíčkov',
pieSliceText: 'value', is3D: true, tooltipText:
'percentage', colors:['#8EBF04','#2aafe3','#ffff00'],
pieSliceTextStyle: {color: '#333333', fontSize: 14},
titleTextStyle: {color: 'white', fontSize: 14},
legendTextStyle: {color: 'white', fontSize: 14}});
  }
</script>

```

Uvedený JavaScript nám na stránke vygeneruje nasledovné koláčové grafy.



Obr. 9 Grafy v sekcii Štatistika

## ZÁVĚR

Hlavným cieľom tejto bakalárskej práce bolo vytvoriť systém, ktorý zjednoduší každodennú evidenciu množstva dát o firemných zákazníkoch a službách, ktoré využívajú a oslobodí tak pracovníkov technickej podpory od pracnej evidencie a fakturácie. Pracovníci sa tak môžu viac venovať technickej podpore zákazníkov a riešeniu ich problémov, pričom na nevyhnutné formálne úkony im slúži prehľadná a svojou štruktúrou jednoduchá webová aplikácia.

Pri tvorbe systému som začal základnou analýzou problematiky, ktorá mi umožnila navrhnúť štruktúru aplikácie. Následne som zvolil potrebné technológie, ktoré som popísal v teoretickej časti tejto práce. V praktickej časti som popísal jednotlivé technické postupy, ktoré som v aplikácii vytvoril.

V súčasnej dobe je celá aplikácia aktívne využívaná v praxi a jej demoverzia je prístupná na adrese <http://demo.host4web.sk>.

## **ZÁVĚR V ANGLIČTINĚ**

The main objective of this bachelor thesis was to create system, which will make every day evidence of many company clients and services data much easier and will make company employees free from hard evidence and invoicing. Employees can spend much more time with technical support for customers and solving their problems, while for necessary formal steps they can use simple web application with transparent structure.

I have started creation with basic analysis of issue, which helps me to propose application structure. Then I have chosen needed technologies, described in theoretical part of this thesis. In practical part I have described technical procedures, which I have created.

At present, all web application is in active use and it's demo version is available on <http://demo.host4web.sk>.

## SEZNAM POUŽITÉ LITERATURY

- [1] PROKOPOVÁ, Zdenka. *Databázové systémy MySQL+PHP*. Zlín: FAI UTB Zlín, 2006. ISBN 80-7318-486-9.
- [2] LUBOMÍR, Lacko. *PHP a MySQL - hotová řešení*. Brno: Computer Press, 2006. ISBN 80-251-1249-7.
- [3] W3C.org: HTML & CSS. [online]. [cit. 2012-05-26]. Dostupné z: <http://www.w3.org/standards/webdesign/htmlcss>
- [4] MySQL.com: About MySQL. [online]. [cit. 2012-05-26]. Dostupné z: <http://www.mysql.com/about/>
- [5] ULLMAN, Larry. *PHP a MySQL: názorný průvodce tvorbou dynamických WWW stránek*. Brno: Computer Press, 2004, 534 s. ISBN 80-251-0063-4.
- [6] WELLING, Luke. *MySQL: průvodce základy databázového systému*. Brno: Computer Press, 2005, 255 s. ISBN 80-251-0671-3.
- [7] SCHNEIDER, Robert D. *MySQL: oficiální průvodce tvorbou, správou a laděním databázi*. Praha: Grada Publishing, 2006, 372 s. ISBN 80-247-1516-3.
- [8] PHP.net: History of PHP. [online]. [cit. 2012-05-26]. Dostupné z: <http://sk.php.net/manual/en/history.php.php>

## **SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

HTML Hyper Text Markup Language.

CSS Cascading Style Sheets.

PHP Hypertext Preprocessor.

PDF Portable Document Format.

SQL Structured Query Language.

W3C World Wide Web Consortium.

SHA Secure Hash Algorithm.

URL Uniform Resource Locator.

FTP File Transfer Protocol.

**SEZNAM OBRÁZKŮ**

<i>Obr. 1 Prostredie editoru Coda</i> .....	20
<i>Obr. 2 Rozhranie FTP klienta Transmit</i> .....	21
<i>Obr. 3 Návrh štruktúry administračného rozhrania</i> .....	24
<i>Obr. 4 Ukážka zrealizovaného rozhrania fakturačného systému</i> .....	25
<i>Obr. 5 Nastavenie časovača skriptov CRON</i> .....	38
<i>Obr. 6 Objednávkový formulár</i> .....	41
<i>Obr. 7 Nová objednávka vo fakturačnom systéme</i> .....	43
<i>Obr. 8 Potvrdzovanie novej objednávky pracovníkom</i> .....	43
<i>Obr. 9 Grafy v sekcii Štatistika</i> .....	45

**SEZNAM TABULEK**

<i>Tab. 1 users</i> .....	27
<i>Tab. 2 clients</i> .....	28
<i>Tab. 3 domains</i> .....	30
<i>Tab. 4 invoices</i> .....	32
<i>Tab. 5 invoices_subjects</i> .....	33
<i>Tab. 6 pricelist</i> .....	34

## **SEZNAM PŘÍLOH**

P1: ZDROJOVÝ SQL KÓD PRE VYTVORENIE TABULIEK V MYSQL DATABÁZE

## **PŘÍLOHA P I: ZDROJOVÝ SQL KÓD PRE VYTVORENIE TABULIEK V MYSQL DATABÁZE**

```
CREATE TABLE IF NOT EXISTS `clients` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `NAME` varchar(60) NOT NULL,  
  `WEBADMIN` varchar(30) NOT NULL,  
  `ADDRESS` varchar(60) NOT NULL,  
  `TOWN` varchar(60) NOT NULL,  
  `ZIP` varchar(10) NOT NULL,  
  `ICO` varchar(10) NOT NULL,  
  `DIC` varchar(10) NOT NULL,  
  `IC_DPH` varchar(12) NOT NULL,  
  `EMAIL` varchar(60) NOT NULL,  
  `PHONE` varchar(30) NOT NULL,  
  `CONTACT_NAME` varchar(60) NOT NULL,  
  `SKNIC` varchar(10) NOT NULL,  
  `NOTE` text NOT NULL,  
  `TOKEN` varchar(50) NOT NULL,  
  `ACTIVE` tinyint(1) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `domains` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `DOMAIN` varchar(63) NOT NULL,  
  `TLD` varchar(10) NOT NULL,
```

```
`TLD_ID` int(11) NOT NULL,  
`HOSTING` tinyint(1) NOT NULL,  
`HOSTINGONLY` tinyint(1) NOT NULL DEFAULT '0',  
`DATE` date NOT NULL,  
`PERIOD` tinyint(1) NOT NULL,  
`EXPIRATION` date NOT NULL,  
`DOM_EXPIRATION` date NOT NULL,  
`CLIENT_ID` int(11) NOT NULL,  
`TOKEN` varchar(50) NOT NULL,  
`CONFIRMED` tinyint(1) NOT NULL DEFAULT '0',  
`ACTIVE` tinyint(1) NOT NULL,  
PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `invoices` (  
  `VS` int(6) NOT NULL,  
  `ORDER_NO` int(11) NOT NULL,  
  `CLIENT_ID` int(11) NOT NULL,  
  `TYPE` varchar(2) NOT NULL DEFAULT 'PF',  
  `DATEFROM` date NOT NULL,  
  `DATETO` date NOT NULL,  
  `PAYMENT` varchar(10) CHARACTER SET utf8 COLLATE utf8_slovak_ci NOT  
  NULL,  
  `TOKEN` varchar(50) NOT NULL,  
  PRIMARY KEY (`ORDER_NO`),  
  KEY `ORDER_NO` (`ORDER_NO`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `invoices_subjects` (
```

```
  `ID` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `ORDER_NO` int(11) NOT NULL,
```

```
  `SERVICE` varchar(100) CHARACTER SET utf8 COLLATE utf8_slovak_ci NOT NULL,
```

```
  `DATEFROM` varchar(12) NOT NULL,
```

```
  `DATETO` varchar(12) NOT NULL,
```

```
  `PRICE` float NOT NULL,
```

```
  `PRICEVAT` float NOT NULL,
```

```
  PRIMARY KEY (`ID`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `pricelist` (
```

```
  `ID` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `NAME` varchar(30) NOT NULL,
```

```
  `PERIOD` tinyint(1) NOT NULL,
```

```
  `PRICE` float NOT NULL,
```

```
  `PRICEVAT` float NOT NULL,
```

```
  PRIMARY KEY (`ID`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `services` (
```

```
  `ID` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `HOSTING_ID` tinyint(4) NOT NULL,
```

```
`NAME` varchar(20) COLLATE utf8_slovak_ci NOT NULL,  
PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_slovak_ci  
AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `tlds` (  
  `ID` int(2) NOT NULL AUTO_INCREMENT,  
  `NAME` varchar(10) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```

```
CREATE TABLE IF NOT EXISTS `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(30) NOT NULL,  
  `password` varchar(128) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT;
```