

**UNIVERZITA PARDUBICE
ÚSTAV ELEKTROTECHNIKY
A INFORMATIKY**

STUDENTSKÉ WEB HOSTINGOVÉ CENTRUM

BAKALÁŘSKÁ PRÁCE

**AUTOR PRÁCE: Josef Bačkovský
VEDOUCÍ PRÁCE: Ing. Lukáš Čegan**

2007

**UNIVERSITY OF PARDUBICE
INSTITUTE OF ELECTRICAL ENGINEERING
AND INFORMATICS**

STUDENT WEB HOSTING CENTER

BACHELOR WORK

**AUTHOR: Josef Bačkovský
SUPERVISOR: Ing. Lukáš Čegan**

2007

Vysokoškolský ústav: Ústav elektrotechniky a informatiky
Katedra/Ústav: Ústav elektrotechniky a informatiky
Akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pro: Bačkovský Josef

Studijní program: Informační technologie

Studijní obor: Informační technologie

Název tématu: Studentské web hostingové centrum

Zásady pro zpracování:

Teoretická část bude obsahovat přehled a zhodnocení dostupných webových technologií a výběr nejvhodnějších aplikací pro vytvoření studentského web hostingového centra. Implementační část bude spočívat v instalaci a konfiguraci vybraného webového serveru s podporou vybrané skriptovací technologie a instalaci databázového systému. Dále bude naprogramována webová aplikace, která umožní registraci studentů v tomto centru a která automatickým způsobem přidělí studentům diskový prostor na serveru spolu s vytvořenou databází.

Seznam odborné literatury:

- AULDS, CH. *Linux – administrace serveru Apache*. Grada, 2003.
- DOSTÁLEK, L., KABELOVÁ, A. *Velký průvodce protokoly TCP/IP a systémem DNS*. Computer Press, Praha, 1999.
- GUNDAVARAM, S. *CGI programování*. Computer Press, 1998.
- LEE, J., WARE, B. *Open Source – vývoj webových aplikací (Linux, Apache, MySQL, PHP a Perl)*. Computer press, 2003.
- POŠMURA, V. *Apache, Příručka správce WWW serveru*. Computer Press, 2002.
- SATRAPA, P. *Perl pro zelenáče*. Neokortex, 2000.

Rozsah: 30 stran

Vedoucí práce: Ing. Čegan Lukáš

Vedoucí katedry (ústavu): prof. Ing. Pavel Bezoušek, CSc.

Datum zadání práce: 30. 11. 2006

Termín odevzdání práce: 18. 5. 2007

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 18.05.2007

Josef Bačkovský
(vlastnoruční podpis)

ABSTRAKT

Cílem této práce je vytvoření studentského web hostingového centra a jeho uvedení do provozu. V první části jsou popsány používané technologie v oblasti web hostingu, dále jsou v ní popsány kritéria, podle kterých lze web hostingy porovnávat a přehled dostupných web hostingů. V druhé části je popsána volba technologií, které použiji a odůvodnění těchto voleb. Poslední část se zabývá samotnou realizací projektu. Obsahuje popis konfigurace jednotlivých použitých technologií a implementaci mnou tvořené aplikace pro správu web hostingového centra.

OBSAH

1. ÚVOD	11
2. PODSTATA HOSTINGU.....	12
2.1 Web hosting.....	12
2.2 Server hosting	12
3. POROVNÁNÍ DOSTUPNÝCH WEB HOSTINGŮ.....	13
3.1 Velikost prostoru na serveru	13
3.2 Způsob přístupu k nahrávání dat.....	13
3.3 E-mail.....	14
3.4 Serverové skripty a databáze.....	14
3.5 Technická podpora	14
3.6 Administrační rozhraní.....	14
3.7 Traffic	15
3.8 Rychlost serveru	15
3.9 Dostupnost a konektivita web hostingů	15
4. VOLBA WEB HOSTINGU.....	16
4.1 Volba bezplatného web hostingů	16
4.2 Volba komerčního web hostingů	16
4.3 Porovnání bezplatných a komerčních web hostingů	17
5. ARCHITEKTURA.....	18
5.1 Databáze	19
5.2 FTP	19
5.3 SMTP.....	19
5.4 Webový server.....	20
5.4.1 Historie	20
5.4.2 Platformy a rozmístění	21
5.4.3 Webové aplikace.....	21
5.4.4 Shrnutí.....	22
5.5 Virtuální server Apache.....	22

6.	TECHNOLOGIE WEB HOSTINGOVÉHO CENTRA.....	23
6.1	Apache	23
6.2	PHP	23
6.3	MySQL	23
6.4	PhpMyAdmin.....	24
6.5	ProFTPD	24
7.	PŘEDPOKLÁDANÉ KOMPLIKACE.....	24
7.1	Automatická konfigurace Apache.....	24
7.2	Konfigurace FTP serveru.....	25
7.3	Bezpečnost.....	25
8.	REGISTRAČNÍ APLIKACE	25
9.	AUTOMATICKÁ KONFIGURACE APACHE	26
9.1	VirtualDocumentRoot.....	27
9.2	VirtualDocumentRootIP	28
9.3	VirtualScriptAlias.....	28
9.4	VirtualScriptAliasIP.....	28
10.	NASTAVENÍ DNS SERVERU	29
11.	MYSQL A PHPMYADMIN.....	30
12.	NASTAVENÍ FTP SERVERU.....	32
13.	TVORBA APLIKACE V JAZYCE PHP	34
13.1	Uživatelská část aplikace.....	35
13.1.1	index.php	37
13.1.2	registruj.php	37
13.1.3	login.php	38
13.1.4	odhlásit.php.....	39
13.1.5	zmenit_heslo.php	39
13.1.6	zapomenute_heslo.php.....	39
13.1.7	Správce souborů.....	40
13.2	Administrační část aplikace.....	40
13.2.1	overovani.php	41
13.2.2	index.php	41
13.2.3	zobrazit_uzivatele.php	41

14. ZABEZPEČENÍ	41
14.1 Přístup do databáze	42
14.2 Zabezpečení FTP	42
14.3 Zabezpečení psané aplikace	42
14.3.1 Nepřihlášení uživatelé.....	42
14.3.2 Přihlášení uživatelé	42
14.3.3 Správce hostingového centra	43
14.4 Ostatní zabezpečení	43
15. ZÁVĚR	44

SEZNAM OBRÁZKŮ A TABULEK

1. Obr. 1) Architektura serveru.....	str. 18
2. Obr. 2) Architektura web hostingového centra.....	str. 26
3. Obr. 3) Komunikace s databází	str. 31
4. Obr. 4) Adresářová struktura	str. 35
5. Obr. 5) Struktura uživatelské části aplikace	str. 36
6. Obr. 6) Struktura administrační části aplikace	str. 40
7. Tab. 1) Porovnání dostupných web hostingů	str. 17

1. ÚVOD

V této bakalářské práci se budu zabývat problematikou Web hostingů. Cílem mé práce je vytvoření plnohodnotného web hostingového centra a jeho spuštění. Systém by měl sloužit studentům Univerzity Pardubice pro prezentaci jejich osobních internetových stránek.

V první části své bakalářské práce nejprve vysvětlím podstatu web hostingů a způsob jeho funkce. Také v ní popíši kritéria pro volbu a porovnání kvality web hostingů a uvedu zásadní rozdíly mezi komerčním a bezplatným web hostingem, podle uvedených kritérií popíši příklad zvolení bezplatného i komerčního web hostingů ze současně dostupných hostingů. Porovnám jejich služby a uvedu výhody a nevýhody. Dále pak uvedu přehled používaných technologií.

V druhé části uvedu přehled technologií, které jsem zvolil pro tvorbu mého web hostingového centra. Tato část obsahuje především předpokládaný postup tvorby a také předpokládané komplikace v průběhu realizace projektu.

V třetí a poslední části budu popisovat konečnou tvorbu mého web hostingového centra. Popis postupu konfigurace jednotlivých použitých technologií, celkovou implementaci systému pro správu centra a řešení bezpečnostních problémů.

2. Podstata hostingu

2.1 Web hosting

Web hosting je umístění webových stránek a dalších služeb (e-mail, FTP, statistiky atd.) na výkonném a spolehlivém počítači (serveru), který je připojen do sítě internet. Rychlost jeho připojení vysoce převyšuje běžné připojení k internetu. Na serveru má každý zákazník vytvořená práva pro různé služby a rezervovaný diskový prostor (řádově desítky či stovky MB). Tento server má svoji vlastní IP adresu v číselné podobě. Bylo by velice nepraktické si pamatovat číselnou podobu adres, a proto se používá DNS server (Domain Name Server). Doména je základní adresní jednotka na internetu reprezentující konkrétní IP adresu. Doména prvního řádu má tvar .doména (.cz). Doména druhého řádu má tvar doména.cz a je placená. Doména třetího řádu, ve tvaru doména.poskytovatel.cz nebo poskytovatel.cz/doména, je zpravidla zdarma. DNS server má databázi jmen a jim příslušejících IP adres. Zadáme-li do prohlížeče adresu ve tvaru www.mojestranka.cz, prohlížeč vyšle požadavek pro DNS server, který mu poskytne informaci, že se jedná o IP adresu 81.35.92.76 a prohlížeč se na ni připojí.

2.2 Server hosting

Alternativou k web hostingu je server hosting. Jedná se o umístění či pronájem počítačového serveru, který je umístěn v prostorách poskytovatele této služby. Server hosting se využívá pro projekty u kterých je zapotřebí využití dalších služeb, nejen www. Zákazník si tak pronajme celý server a může ho používat a konfigurovat dle svého uvážení. Zpravidla má k server administrátorský přístup a může na něm provozovat různé aplikace, nebývá nějak omezen ve způsobu využití serveru. Server hosting je oproti web hostingu značně dražší jeho cena se zpravidla pohybuje od dvou do deseti tisíc korun ročně.

3. Porovnání dostupných web hostingů

V této části se zaměřím na porovnání běžně dostupných web hostingů, a to jak komerčních, tak i bezplatných.

Web hosting je služba s různými parametry. Podle poskytovaných služeb se web hostingy liší cenou. Od web hostingů poskytujících pouze základní služby, které jsou zdarma, až po web hostingy s velkým sortimentem služeb, jejichž cena může být až několik tisíc korun ročně.

Dle povahy projektu, který chceme realizovat, se volí vhodný web hosting.

3.1 Velikost prostoru na serveru

Prvním kritériem pro volbu web hostingů je velikost prostoru na serveru. Bezplatné web hostingy disponují velikostí prostoru od 10 MB do 100 MB. Takový prostor je vhodný pro prezentace, které nejsou příliš náročné na velikost. Rozhodně není vhodný pro stránky s větším množstvím multimediálních dat. Prostor na placených web hostingách se pohybuje od základních 100 MB do několika GB.

3.2 Způsob přístupu k nahrávání dat

Dalším kritériem pro volbu web hostingů je způsob přístupu k nahrávání dat. Je vhodné, když web hosting umožňuje nahrávání dat přes FTP i přes webové rozhraní. Přístup přes FTP nám umožňuje pracovat s prostorem na serveru stejně jako s běžným diskem a taková práce je výrazně rychlejší. Webové rozhraní je vhodné zejména pro drobné úpravy. Například v případě, že je zapotřebí provést modifikaci z počítače, který nemá potřebný software pro FTP přístup.

3.3 E-mail

Dalším kritériem pro volbu web hostingu je možnost vlastních e-mailů (např. jméno@doména.cz) a také souvisejících služeb jako například antispamu, antiviru atd. Většina bezplatných web hostingů neposkytuje vlastní e-mailové adresy.

3.4 Serverové skripty a databáze

Velice důležitým kritériem je podpora skriptovacích aplikačních jazyků a databází. Podpora skriptů ASP, PERL a dalších není samozřejmostí. Stejně tak i databáze MySQL, Oracle, MS SQL a další nemusí web hosting obsahovat. Pro některé méně náročné projekty nejsou nezbytné. Většina bezplatných web hostingů podporuje jazyk PHP a umožňuje nám využití jedné databáze MySQL o maximální velikosti 5 MB. U komerčních web hostingů je samozřejmě kvalita služeb vyšší.

3.5 Technická podpora

Podstatná je také podpora, kterou nám web hosting poskytuje. Je vhodné zvolit web hosting s podporou 24 hodin 7 dní v týdnu. Pokud narazíme na problém, například nefunkční e-mail nebo zapomenuté heslo a další, pak tuto službu určitě oceníme. Bezplatné web hostingy mají obvykle podporu pouze přes e-mail. Kvalitnější komerční web hostingy nabízejí také podporu telefonickou.

3.6 Administrační rozhraní

Web hosting by měl obsahovat administrační prostředí, které nám umožní snadno a přehledně nastavovat parametry našeho web hostingu jako hesla, vytváření e-mailů, subdomén a další. U bezplatných web hostingů jsou tyto možnosti značně omezené. Oproti tomu u placených web hostingů je takovéto prostředí již poměrně dost rozšířené.

3.7 Traffic

Traffic je objem dat, které jsou od serveru požadovány za určitý časový úsek. Rozhodně bychom měli volit web hosting, který má neomezený Traffic, pokud se jedná o komerční službu. U bezplatného web hostingu lze předpokládat, že se jedná o projekt menšího rozsahu, tudíž i objem vyžádaných dat bude menší. V takovém případě omezený Traffic s největší pravděpodobností nebude překážkou.

3.8 Rychlost serveru

Rychlost odezvy serveru by měla být co nejvyšší. Také záleží na tom, v jak rychlém připojení internetu se server nachází. V případě, že některá z uvedených vlastností má nevhodné parametry, budou se návštěvníkům naše stránky zobrazovat dlouhou dobu, což jistě sníží jejich zájem o naše služby. U bezplatných web hostingů tyto parametry nebývají nejlepší z důvodů velkého vytížení serverů. I u komerčních web hostingů musíme vybírat pečlivě.

3.9 Dostupnost a konektivita web hostingu

Měli bychom volit web hosting, který bude umístěn na kvalitním serveru, rychlém a spolehlivém připojení k internetu, které umožní co nejvíce se přiblížit 100 % dostupnosti a nestane se tak, že naše stránky nebude možné zobrazit. Ani u komerčních web hostingů nelze počítat s tím, že mají všechny kvalitní konektivitu a proto je musíme volit pečlivě.

Toto jsou základní kritéria, podle kterých lze porovnávat a volit web hosting, který bude pro nás nejvhodnější. Vždy existuje více variant. Jistou pomoc při výběru a rozhodování, který web hosting zvolit, vám poskytnou různá fóra.

4. Volba web hostingu

Podle uvedených kritérií jsem zvolil jeden bezplatný a jeden komerční web hosting, které jsou v současné době na našem trhu dostupné.

4.1 Volba bezplatného web hostingu

Podle uvedených kritérií jsem zvolil web hosting internet centrum www.ic.cz. Tento bezplatný web hosting poskytuje dle mého názoru relativně kvalitní služby, jako diskový prostor 250 MB s vlastním rozdělením mezi MySQL, e-mail a www adresáři, podporu PHP4 a PHP5, e-mailovou schránku s antivirem a antispamfiltrem, přístup k poště přes webmail, POP3, POP3s, IMAP a IMAPs. Databáze MySQL, PostgreSQL a SQL Lite. Automatické zálohování databází. Tento web hosting dále umožňuje administraci pgsq a MySQL databáze, sledování aktuálně zabraného diskového prostoru, ankety a mailform, statistik přístupů, webFTP atd. Existují i bezplatné web hostings, které poskytují web hosting zdarma s doménu druhého řádu, zákazník zaplatí pouze poplatek za provoz správci domény. Jeden z nich je například www.kgb.cz.

4.2 Volba komerčního web hostingu

Z komerčních web hostingů jsem si vybral web hosting C4. Diskový prostor 1 GB. 100 e-mailových schránek, přístup k poště přes webmail, antispam, antivir, výběr e-mailů přes POP, IMAP a odesílání mailů přes SMTP. 100 subdomén, neomezený počet web aliasů, skriptovací jazyk PHP, možnost přidání skriptů do CRONU (pravidelné plánované spouštění). FTP přístup, bezpečný přístup přes SSL protokol, přístup k logům (access log, error log), možnost vlastních 3xx, 4xx a 5xx chybových stránek. 10 MySQL databází, administrace přes webové rozhraní phpMyAdmin. Podpora 24 hodin 7 dní v týdnu e-mailem a telefonická podpora.

4.3 Porovnání bezplatných a komerčních web hostingů

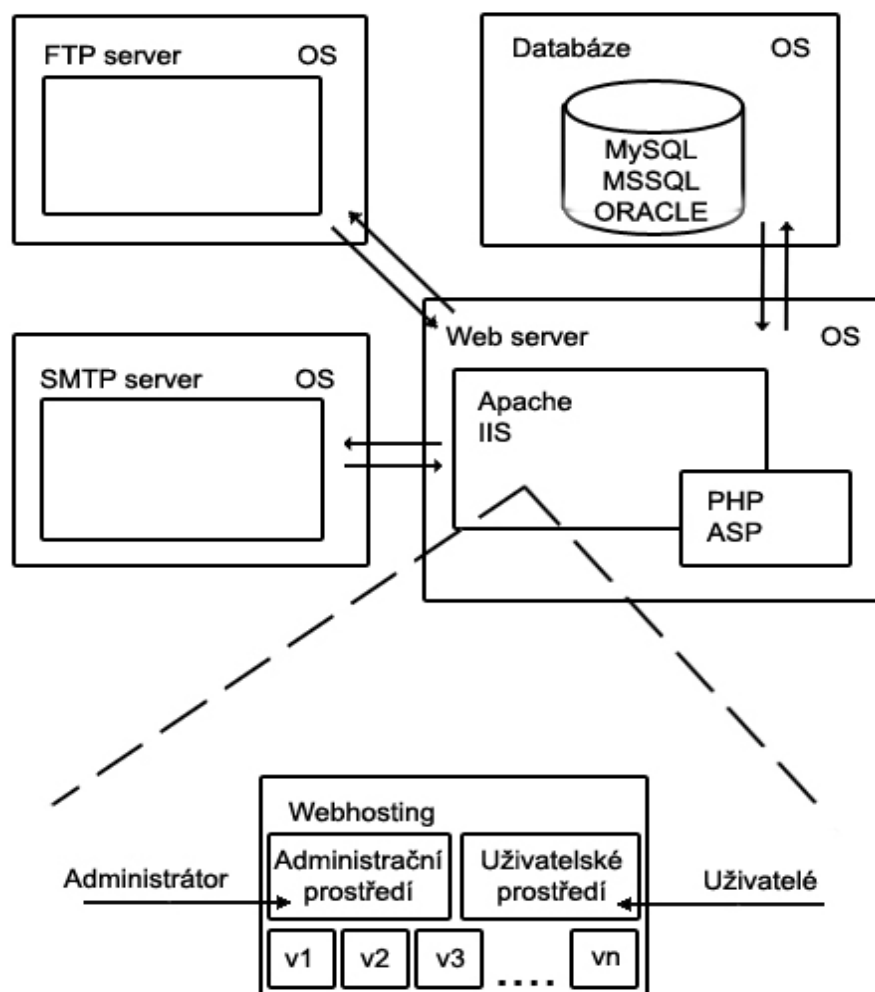
Největší nevýhodou bezplatného web hostingu v porovnání s komerčním je dle mého soudu dostupnost a konektivita. Stránky na bezplatném web hostingu se zpravidla načítají mnohem delší dobu, než na komerčním web hostingu. Také technická podpora není tak kvalitní. Ale existuje velké množství projektů, pro které tyto nedokonalosti nejsou až tak závažným problémem. Značnou nevýhodou pro některé projekty je také doména třetího řádu, která nepůsobí příliš profesionálně. Hlavní výhodou placeného web hostingu v porovnání s bezplatným je rychlost serveru, konektivita a kvalitní technická podpora. Pro některé projekty je také velkou výhodou podpora CRONU.

Název	Prostor [MB]	Doména II III	Mail	FTP	PHP	DB	Reklama	Cena [Kč/rok]
Bezplatné								
ic.cz	250	ne ano	ano	ano	ano	ano	ano	0
aspweb.cz	30	ne ano	ano	ano	ne	ano	ne	0
fbi.cz	200	ano ano	ano	ano	ano	ano	ano	0
PHP5.cz	10	ne ano	ne	ano	ano	ano	ne	0
Webzdarma	50	ne ano	ano	ano	ano	ano	ano	0
Komerční								
C4	1000	ano ano	ano	ano	ano	ano	ne	1200
Web4ce	5000	ano ano	ano	ano	ano	ano	ne	1008
VSHosting	2000	ano ano	ano	ano	ano	ano	ne	1188
Corecluster	310000	ano ano	ano	ano	ano	ano	ne	3 468
Savana	1600	ano ano	ano	ano	ano	ano	ne	600

Tab. 1) Porovnání dostupných web hostingů

- **Doména II** - Hostování stránek na doméně druhého řádu
(www.mojestranky.cz)
- **Doména III** - Hostování stránek na doméně druhého řádu
(www.mojestranky.cz)
- **Mail** - Ke službě je poskytován e-mail
- **FTP** - Stránky lze nahrávat přes protokol FTP
- **PHP** - Podpora PHP
- **DB** - Lze využívat databázi
- **Reklama** - Na stránkách se zobrazuje reklama provozovatele
- **Cena** - Cena ročního pronájmu web hostingu

5. Architektura



Obr. 1) Architektura serveru

Základem architektury web hostingu jsou jednotlivé servery Apache, databází, FTP a SMTP. Tyto servery mohou pracovat na různých platformách, nebo-li operačních systémech (OS). Mohou být umístěny na jednom nebo více fyzických serverech.

5.1 Databáze

Databáze je báze dat. Jedná se o uspořádanou množinu informací, která je uložena na paměťovém médiu. Tyto data jsou spravována databázovým serverem. Klienti k datům přistupují pomocí aplikací využívajících jazyka SQL (Structured Query Language). V širším smyslu se do databáze zahrnují i softwarové prostředky pro manipulaci, přístup a úpravy dat. Existuje množství databázových systémů (MySQL, MSSQL, ORACLE atd.).

5.2 FTP

FTP (File Transfer Protocol) je protokol určený pro přenos dat mezi počítači. Na počítačích mohou běžet různé operační systémy. Nejčastěji se využívá ke sdílení dat (hudba ,videa atd.) a především spravování účtů internetových stránek. Použijeme-li kvalitního FTP klienta (např. Total Comander), můžeme pracovat s daty na server stejně jako s daty na pevném disku našeho počítače.

5.3 SMTP

SMTP (Simple Mail Transfer Protocol) je protokol určený pro přenos zpráv elektronické pošty (e-mail). Přenos zpráv protokol zajišťuje přímým spojením mezi odesílatelem a adresátem. Doručená zpráva je adresátovi uložena do e-mailové schránky. Adresát si doručenou zprávu zobrazí nezávisle na stavu odesílatele.

5.4 Webový server

Server http, nebo též webový server je software, který odesílá obsah prohlížečům v síti internet prostřednictvím http protokolu. Existuje více jak 20 http serverů. My se budeme zabývat jen těmi nejpoužívanějšími. V oblasti serveru poskytujících webové stránky je na prvním místě Apache s podílem 58.70 %, na druhém místě je Internet Information Services (IIS) od Microsoftu s podílem 31.09 %. Ostatní servery mají dohromady jen 9.40 %.

5.4.1 Historie

Název vychází z anglického „A patchy server“ (záplatovaný server). Apache http server je webový server s otevřeným kódem pro Linux, BSD, Microsoft Windows a další platformy. Vývoj Apache začal v roce 1993. Původní jméno projektu bylo NCSA HTTPd. Po roce práce opustil vývojářský tým hlavní programátor, což vedlo ke zpomalení vývoje a posléze k jeho úplnému zastavení. V tu dobu však již NCSA HTTPd používali správci webových serverů. Začali k němu dodávat vlastní záplaty (pache). Později byla založena e-mailová konference. Sběr úprav a jejich distribuce začala být koordinována. První veřejná verze vydána v dubnu 1995 měla označení 0.6.2. Poté proběhlo kompletní přepsání kódu, který již neobsahoval nic z původního NCSA HTTPd a založení Apache Group, která se stala základem vývojářského týmu. Aktuální verze Apache je 2.2.

Internet Information Services (IIS) od Microsoftu je sbírka služeb, která zahrnuje servery pro Web, FTP, SMTP a Network News Transfer Protokol (NNTP). IIS byl poprvé distribuován s Windows NT 3.51. Podobně jako Apache i IIS prošel rozsáhlým vývojem a aktuální verze se od svých předchůdců velice odlišuje. Přestože IIS není otevřený kód, jeho licence je velice liberální, co se týče rozmístění a použití. Aktuálně používaná verze IIS je 6.0.

5.4.2 Platformy a rozmístění

Apache byl vytvořen pro UNIX a dlouhou dobu byl s touto platformou spjat pro výkon a spolehlivost. Apache 1.3 je poslední verzí výhradně pro UNIX. Apache 2.0 je verze, která byla napsána s nezávislostí na OS a přitom se snažila o nejlepší využívání vlastností OS. Vývojáři k dosažení této vlastnosti využívají Apache Portable Runtime (APR) a Multi-Processing Modules (MPMs). MPMs jsou moduly specifické pro každou platformu.

Microsoft vyvinul IIS pro jedinou platformu, a sice pro Windows. Tento přístup má výhodu bližší spolupráce se základem Windows OS, což může zlepšit výkon IIS. Také IIS může pracovat s některými klíčovými elementy, jako například HTTP.sys.

5.4.3 Webové aplikace

V dnešní době většina webových stránek používá dynamické prvky. Apache používá Common Gateway Interface (CGI) pro podporu webových aplikací. Pokaždé, když URL obsahuje CGI, spustí CGI program, který snižuje výkon. Z tohoto důvodu vytvořili vývojáři moduly pro začlenění webových aplikačních programovacích jazyků. Díky modulárnímu základu Apache podporuje mnoho programovacích jazyků a v případě potřeby může podporovat i nové jazyky.

Počínaje verzí 3.0, IIS podporuje Active Server Pages. ASP je plnohodnotné programovací prostředí webových aplikací, které se běžně používá. IIS podpora pro ASP je kvalitnější než podpora přidaného modulu.

IIS může také podporovat i jiné programovací jazyky. Nejen ASP, ale také třeba PHP a Perl. Přesto na serverech používajících IIS je spíše preferován ASP, zatímco širší řada jazyků (například PHP, Perl, C++ atd.) se používá na serverech Apache.

5.4.4 Shrnutí

Jedna ze zásadních otázek, kterou si položíme při porovnávání Apache a IIS je otázka, který webový server je výkonnější. Tímto problémem se již zabývalo více studií, ale výsledky nejsou jednoznačné. Lze říci, že na výkon má větší vliv to, jak je web psán, než-li to, zda je umístěn na serveru používajícím Apache nebo IIS.

Apache i IIS nabízejí řadu pokročilých a podobných technických vlastností. Jednou z nich je například podpora používání tzv. filtrů. Jedná se o rozšíření základu webového serveru pro podporu programovacích jazyků. Používáním filtrů oba webové servery v podstatě odstraní snížení výkonu způsobené použitím externích jazykových překladačů do spustitelných jazyků jako ASP, PHP a Perl.

Vzhledem k faktu, že IIS je psané pouze pro platformu Windows a nikoli jako Apache pro více platform by se mohlo zdát, že pouze IIS má bližší spolupráci na úrovni jádra OS. Ale jsou programy (například phhttpd), jejichž instalace může nabídnout vyšší výkon zvláště pro statický obsah. Hlavním rozdílem, je že phhttpd není standardní součástí Apache, nýbrž jen jeho doplněk.

Oba dva webové servery nejsou tak odlišné jak by se mohlo na první pohled zdát. Oba nabízejí podobné služby a výhody. Apache i IIS mají liberální licenční podmínky, jsou poměrně lehce konfigurovatelné a ovladatelné. Nejedná se tedy o otázku, který z nich je lepší, ale spíše o otázku, který z nich lépe vyhovuje požadavkům našeho projektu.

5.5 Virtuální server Apache

Apache je schopen obsluhovat několik webových serverů na jednom fyzickém serveru. Takovéto servery se nazývají virtuální. Toho se využívá v projektech jako je tento. Jednotlivé virtuální servery se chovají jako plnohodnotný server Apache. V našem případě každému registrovanému uživateli vytvoříme virtuální server, na kterém bude moci provozovat svou internetovou prezentaci.

6. technologie web hostingového centra

Pro tvorbu mého web hostingového centra použiji následující technologie Apache, PHP, MySQL, phpMyAdmin, ProFTPD. V další části mé práce uvedu důvody, pro které jsem zvolil právě tyto technologie.

6.1 Apache

Apache web server jsem se rozhodl použít, protože je to nejrozšířenější web server, který poskytuje všechny služby potřebné k tvorbě hostingového centra. S tímto serverem mám také již zkušenosti, což usnadní mou práci. Pro tento server existuje velké množství rozšíření v podobě volně dostupných modulů. I Apache jako takový je volně dostupný.

6.2 PHP

PHP skriptovací programovací jazyk jsem vybral především pro mé zkušenosti s ním. Existuje více programovacích jazyků, které by pro tento projekt byly vhodné. PHP je značně rozšířený jazyk s velkou podporou. Používá se k tvorbě dynamických webových stránek, tudíž má všechny potřebné instrukce i pro můj projekt.

6.3 MySQL

Databáze MySQL je volně dostupná a značně rozšířená. Neposkytuje sice tak rozsáhlé možnosti, jako například databáze Postgres nebo Oracle. Ale pro tento projekt jsou jeho služby dostačující. Jedná se o poměrně jednoduchý databázový systém, jehož používání je značně rozšířené pro menší a střední projekty. Přesně takové, které předpokládám, že budou vytvářet uživatelé mého hostingového centra. Díky jednoduchosti tohoto databázového systému jeho používání nečiní problémy ani méně zkušenějším uživatelům.

6.4 PhpMyAdmin

PhpMyAdmin je program napsaný v jazyce PHP pro jednoduchou správu databáze MySQL. Umožňuje přehledné a snadné vytváření, rušení a úpravy jednotlivých databází. Dále umožňuje provádět příkazy SQL a spoustu dalších funkcí. Jedná se o nejrozšířenější nástroj pro správu databází. Má podporu pro 52 jazyků. Jeho volba, je vzhledem k použití technologie PHP a MySQL jednoznačná.

6.5 ProFTPD

ProFTPD je kvalitní, snadno konfigurovatelný a bezpečný FTP server. Tento FTP server jsem si zvolil především pro jeho bezpečnost a také pro existenci modulu podporující vytváření FTP účtů podle dat načtených z databáze MySQL. Vzhledem k tomu, že informace o uživateli mého hostingového centra budou uloženy v databázi MySQL, umožní mi tento FTP server vytvářet účty bez nutnosti přidávání jednotlivých účtů do hlavního konfiguračního souboru a následně restartovat FTP server pro jejich aktivaci.

7. Předpokládané komplikace

V této fázi práce předpokládám několik hlavních problémů.

7.1 Automatická konfigurace Apache

První z nich bude automatická konfigurace virtuálních serverů Apache. Pokud bych požil základní direktivy konfigurace virtuálních serverů v hlavním konfiguračním souboru Apache httpd.conf, musel bych každý z virtuálních serverů nakonfigurovat ručně a poté provést restart serveru Apache. Toto řešení vzhledem k předpokládanému množství uživatelů je velice nepraktické a časově náročné pro administrátora hostingového centra. Řešením tohoto problému bude pravděpodobně některý z rozšiřujících modulů pro Apache.

7.2 Konfigurace FTP serveru

Dalším problémem bude konfigurace FTP serveru. Stejně jako při konfiguraci virtuálních serverů Apache, tak i při použití základních direktiv FTP serveru by musel administrátor pro každého nového uživatele web hostingu nakonfigurovat účet pro FTP server a poté restartovat tento server. Opět se jedná o způsob velice nepraktický a časově náročný. Pro odstranění tohoto problému použijí modul pro FTP server `mod_sql_mysql`, který provádí automatickou konfiguraci podle dat získaných z databáze MySQL bez nutnosti restartovat tento server.

7.3 Bezpečnost

Předpokládám také drobné problémy ohledně bezpečnosti web hostingového centra. Především v nastavení FTP serveru, aby uživatelé neměli přístup k datům jiných uživatelů nebo dokonce k datům administrátora. Také zabezpečení registračních a konfiguračních částí webového rozhraní uživatelů.

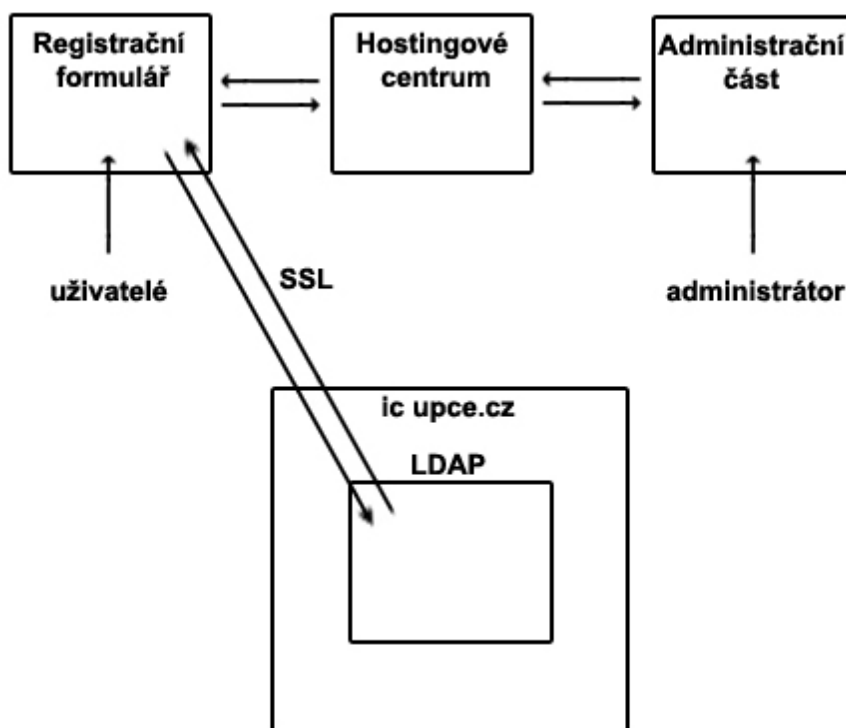
8. Registrační aplikace

Základem je web hostingové centrum, které obsahuje registrační prostředí pro uživatele a administrační prostředí. V registračním formuláři uživatelé vyplní potřebné údaje. Po odeslání údajů aplikace prověří, zda uživatel vyplnil všechny potřebné údaje ve správném formátu. Aplikace zkontroluje, zda uživatelské jméno není již obsazeno a pokud je volné, provede registraci uživatele do databáze a nastavení potřebných konfigurací pro FTP server, Apache a MySQL.

Administrační prostředí je určeno především pro potřeby přidání funkcí, které nepodporuje aplikace phpMyAdmin. V základu bude obsahovat pouze přehledný výpis uživatelů a možnost odstranění jejich účtů. PhpMyAdmin poskytuje všechny základní funkce potřebné pro

administraci uživatelů a bylo by tudíž zbytečné vytvářet veškeré funkce v našem administračním prostředí znovu.

Naznačená komunikace registračního formuláře s databází LDAP nebude součástí této práce. Jedná se pouze o možné rozšíření. Toto rozšíření by bylo vloženo mezi vyplnění registračního formuláře a provedení vlastní registrace do databáze. Před vlastní registrací by aplikace zjistila, zda uživatel je studentem Univerzity Pardubice, vysláním požadavku pro LDAP a pouze v případě kladné odpovědi by proběhla samotná registrace.



Obr. 2) Architektura web hostingového centra

9. Automatická konfigurace Apache

Jak jsem již uvedl hlavním problémem v konfiguraci serveru Apache je jeho automatická konfigurace a schopnost pracovat s nově registrovanými uživateli bez nutnosti restartovat server. Tento problém se dá vyřešit použitím některých modulů určených právě pro jednoduché a spolehlivé vytváření virtuálních serverů. Existuje větší množství

takovýchto modulů, více či méně vyhovujících našemu projektu. Po seznámení se s problematikou a jednotlivými moduly jsem se rozhodl použít modul `mod_vhost_alias`. Základní funkcí tohoto modulu je zjednodušení práce při vytváření virtuálních serverů. Tento modul umožňuje vytvářet virtuální servery dvěma základními způsoby. První způsob je vytváření podle hostitelského jména vytvářeného virtuálního serveru a druhým způsobem je jejich vytváření podle adresy IP. Modul má čtyři základní direktivy.

9.1 VirtualDocumentRoot

Direktiva `VirtualDocumentRoot` umožňuje nastavit kořenový adresář dokumentů pro virtuálního hostitele pomocí interpolované adresářové cesty.

Zápis:

```
VirtualDocumentRoot interpolovaná_adresářová_cesta
```

Pro názornost uvedu příklad. Pokud do hlavního konfiguračního souboru `httpd.conf` vložíme direktivu:

```
VirtualDocumentRoot /www/%0/htdocs
```

Bude požadavek <http://www.domena.cz/index.html> přijatý serverem přeložen na požadavek do adresáře [www/www.domena.cz/htdocs/index.html](http://www.domena.cz/htdocs/index.html). Hodnota `%0` je nahrazena úplným jménem hostitele (v našem případě www.domena.cz).

Lze použít i jednotlivé části hostitelského jména. Hostitelské jméno je děleno na části oddělené tečkou. K vytvoření interpolované adresářové cesty lze využít několik dalších znaků. Tyto znaky určují, o kterou část oddělenou tečkami se jedná. Pro určení zprava doleva je jejich zápis následující: `%1` určuje první část (`www`), `%2` určuje druhou část (`doména`), `%3` třetí část (`cz`) atd. Lze použít také určení zleva doprava přidáním před číslo určující, o kterou část se jedná znaménko mínus (`%-1`). Další možností je určení jen určitého počtu částí. Pomocí

%N.P, kde N určuje, o kterou část se jedná a P určuje počet znaků, který se má použít (např. %1.2 jsou první dva znaky ww).

9.2 VirtualDocumentRootIP

Direktiva VirtualDocumentRootIP je velice podobná předchozí direktivě s tím rozdílem, že místo hostitelského jména vytvářeného serveru je překládána adresa IP. Zápis direktivy a její použití jsou stejné jako u předchozí. Tento způsob má význam v případě, že každý virtuální hostitel má vlastní adresu IP.

9.3 VirtualScriptAlias

Direktiva VirtualScriptAlias umožňuje definovat náhradní jméno adresáře skriptů pomocí použití interpolované adresářové cesty.

Zápis:

```
VirtualScriptAlias interpolovaná_adresářová_cesta
```

Příklad zápisu:

```
VirtualScriptAlias /cgi-bin/ /www/%0/cgi-bin
```

Požadavek http://www.domena.cz/cgi-bin/jmeno_skriptu přijatý serverem bude přeložen na požadavek do adresáře www/www.domena.cz/cgi-bin/jmeno_skriptu.

9.4 VirtualScriptAliasIP

Direktiva VirtualScriptAliasIP je velice podobná předchozí direktivě s tím rozdílem, že místo hostitelského jména vytvářeného serveru je překládána adresa IP. Zápis direktivy a její použití jsou stejné jako u předchozí. Tento způsob má význam v případě, že každý virtuální hostitel má vlastní adresu IP.

Díky tomuto modulu tedy můžeme vytvářet virtuální hostitele jen tím, že v daném adresáři vytvoříme podadresář a modul sám zajistí správný překlad požadavku. Pokud mají všechny virtuální servery, na které je Apache dotázán, stejnou adresu IP, Apache určí do kterého virtuálního serveru má být požadavek přesměrován z hlavičky přijatého požadavku. Použití direktiv vytvořených pro práci s IP adresami je možné jen v případě, že pro každý virtuální server bude přidělena jiná adresa IP.

Pro tento projekt použijeme direktivy `VirtualDocumentRoot` a `VirtualScriptAlias`.

Budeme mít vyhrazenou doménu `mojestranky.hosting.upceuei.cz`. V našem případě nastavení direktiv bude následující:

```
VirtualDocumentRoot
/StHosting/vhost/%2+/%1.1/%1/htdocs

VirtualScriptAlias /cgi-bin/
/StHosting/vhost/%2+/%1.1/%1/cgi-bin
```

Adresář `/www/vhost/` je základní umístění všech virtuálních hostingů. Parametr `%2+` označuje vše od druhé části až na konec adresy (v našem případě `hosting.upceuei.cz`). Vzhledem k předpokládanému velkému počtu uživatelů by bylo nepřehledné mít všechny virtuální servery v jednom adresáři. Další parametr `%1.1`, je první písmeno z první části (m z `mojestranky`), díky tomu budou virtuální servery rozděleny podle prvního písmena z této části. Parametr `%1` určuje celou první část (`mojestranky`). V každém virtuálním serveru budou dva základní adresáře `htdocs` pro obsah stránek a `cgi-bin` jako adresář skriptů.

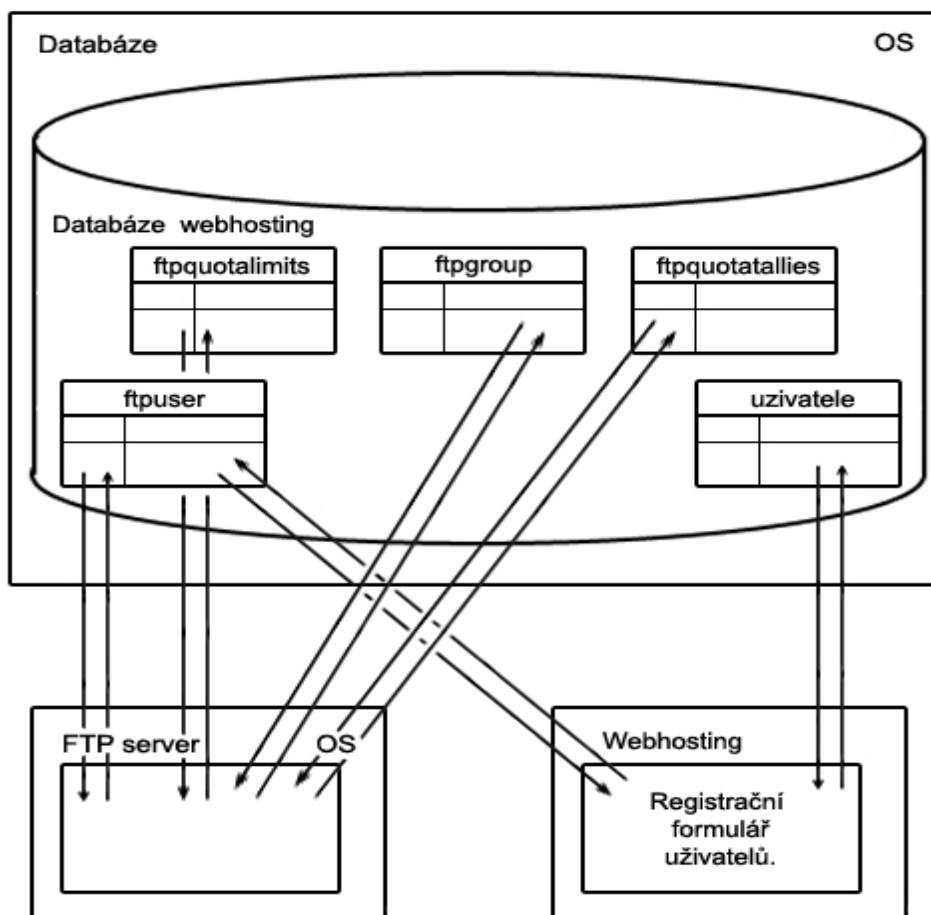
10. Nastavení DNS serveru

Pro tento projekt bylo zapotřebí nastavit DNS server tak aby veškeré požadavky na doménu `cokoli.hosting.upceuei.cz` zasílal na server s určitou IP adresou. Pro potřeby vývoje stačilo přidat do konfiguračního

souboru hosts jednotlivé domény a jim odpovídající IP adresy. Pokud do prohlížeče zadáme adresu ve tvaru něco.hosting.upceuei.cz, systém se nejprve podívá do souboru hosts a teprve pokud se zde záznam s tímto názvem nevyskytuje, ptá se serveru DNS. Tímto způsobem je řešen například localhost. Tento převod domén na adresy IP funguje samozřejmě pouze na počítači, na kterém je takto nakonfigurován soubor hosts. Z tohoto důvodu toto řešení nelze používat i v provozu hostingového centra. Pro běžný provoz je zapotřebí nakonfigurovat DNS server a zónové soubory. Vzhledem k tomu, že nastavení pro provozní podmínky vyžaduje oprávnění super uživatele na serveru, nastavení DNS server jsem neprováděl já sám, ale nastavil ho správce serveru.

11. MySQL a phpMyAdmin

Pro phpMyAdmin nebyla nutná žádná zvláštní nastavení. Pouze pro přístup skrze něj do databáze MySQL bylo zapotřebí vytvářet uživatele pomocí skriptů registračního formuláře. Samotná aplikace hostingového centra využívá jedné databáze, kterou jsem nazval web hosting.



Obr. 3) Komunikace s databází

Na Obr. 3 je zobrazena tato databáze, její tabulky a přístupy FTP serveru a registračního formuláře, aplikace k datům v tabulkách.

Tabulky obsluhující FTP server:

- **ftpuser** – Obsahuje údaje pro tvorbu uživatelských účtů FTP.
- **ftpquotalimits** – Obsahuje nastavení quot pro skupiny uživatelů nebo pro jednotlivce.
- **ftpgroup** – Obsahuje údaje o uživateli a skupině, pod kterou je FTP účet spuštěn na serveru.
- **ftpquotatallies** – Je tabulka využívaná FTP serverem pro nastavení quot, její obsah mění pouze FTP server.

Tabulky obsluhující registrační formulář:

- **uzivatele** – Tato tabulka obsahuje pouze údaje, které využívá hostingové centrum.
- **ftpuser** – Registrační formulář přidává jednotlivé uživatele FTP serveru.

Také bylo zapotřebí vytvořit uživatele, pod kterým se přihlašuje formulář pro registraci a další skripty. Z důvodu maximalizace bezpečnosti hostingového centra bylo nezbytné vytvářet účty pro přístup do databáze MySQL jen s nezbytným minimem potřebných oprávnění pro daný účel. Takže například účet pro přístup registračního formuláře je vytvořen pouze s právy pro vkládání záznamů do tabulek a nelze pomocí něj záznamy odstraňovat.

12. Nastavení FTP serveru

Jak jsem již uvedl největším problémem v nastavení FTP serveru je automatické vytváření FTP účtu pro přístup uživatelů bez nutnosti zasahování správce hostingu. Pro řešení tohoto problému jsem zvolil použití FTP serveru ProFTPD a jeho modulu `mod_sql_mysql`. Tento modul umožňuje vytváření účtů bez nutnosti konfigurování každého z nich a restartování serveru. Účty FTP vytváří pomocí dat uložených v databázi MySQL. Nastavení FTP serveru není složité, v tomto projektu obsahuje několik standardních direktiv, které zde nebudu popisovat a přejdu rovnou k nastavení modulu pro tvorbu FTP účtů podle dat uložených v databázi.

```
Include /etc/proftpd.d/*.conf
```

Nejdříve je nutné vložit direktivu, která načte daný modul.

```
SQLConnectInfo databasename@host user password
```

Připojení do databáze MySQL. Databasename je název databáze, do které se FTP server připojuje, host určuje, ke kterému serveru, user je uživatel, pod kterým se přihlašuje a password je heslo pro přístup do databáze.

```
SQLUserInfo webhosting userid passwd uid gid homedir
```

Volba tabulky a jejích sloupců. Web hosting je název tabulky, ve které jsou údaje uloženy. Sloupec userid je jméno uživatele FTP serveru, passwd obsahuje heslo pro FTP přístup. Uid určuje, pod kterým uživatelem na serveru je FTP spojení spuštěno, gid určuje pod kterou skupinou. Homedir je domovský adresář připojení, nad tento adresář nemá uživatel přístup.

```
SQLGroupInfo ftpgroup groupname gid members
```

Určení skupin, pod kterými běží FTP spojení. Groupname obsahuje název skupiny, gid její identifikační číslo a members její členy.

```
SQLHomedirOnDemand off
```

Pokud je tento parametr nastaven na on, vytvoří FTP server adresář pokud neexistuje. V našem případě jej nevytváří.

```
SQLLog PASS updatecount SQLNamedQuery updatecount  
UPDATE "count=count+1, accessed=now() WHERE  
userid='%u'" ftpuser
```

Připočítává přihlášení k FTP účtu, jejich počet je uložen v tabulce

```
SQLLog STOR,DELE modified SQLNamedQuery modified  
UPDATE "modified=now() WHERE userid='%u'" ftpuser
```

Mění datum poslední změny uložené v tabulce ftpuser.

```
SQLNamedQuery get-quota-limit SELECT "name,  
quota_type, per_session, limit_type, bytes_in_avail,  
bytes_out_avail, bytes_xfer_avail, files_in_avail,
```



```
files_out_avail, files_xfer_avail FROM ftpquotalimits
WHERE name = '%{0}' AND quota_type = '%{1}'"
```

```
SQLNamedQuery get-quota-tally SELECT "name,
quota_type, bytes_in_used, bytes_out_used,
bytes_xfer_used, files_in_used, files_out_used,
files_xfer_used FROM ftpquotatallies WHERE name =
'"{0}"' AND quota_type = '{1}'"
```

```
SQLNamedQuery update-quota-tally UPDATE
"bytes_in_used = bytes_in_used + %{0}, bytes_out_used
= bytes_out_used + %{1}, bytes_xfer_used =
bytes_xfer_used + %{2}, files_in_used = files_in_used
+ %{3}, files_out_used = files_out_used + %{4},
files_xfer_used = files_xfer_used + %{5} WHERE name =
'"{6}"' AND quota_type = '{7}'"
```

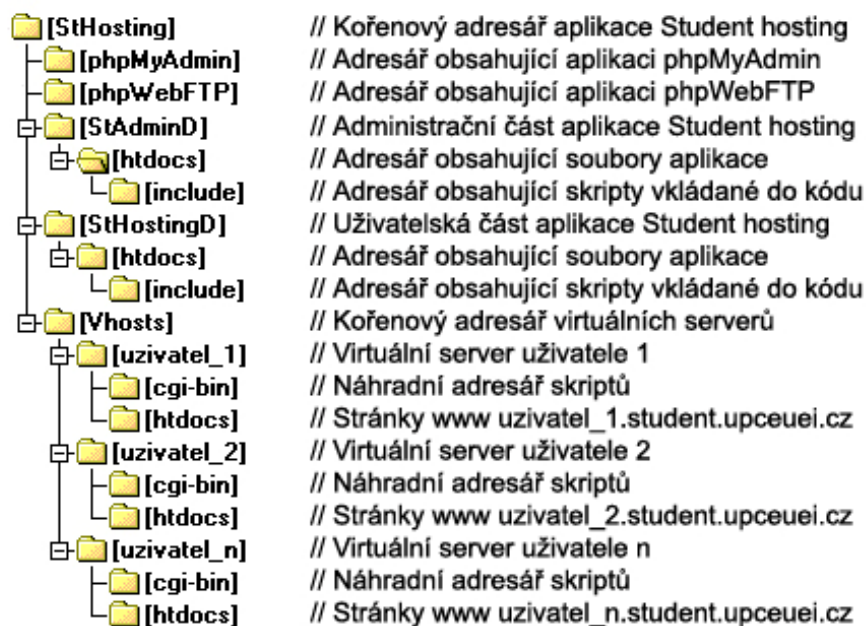
```
SQLNamedQuery insert-quota-tally INSERT "%{0}, %{1},
%{2}, %{3}, %{4}, %{5}, %{6}, %{7}" ftpquotatallies
QuotaLimitTable sql:/get-quota-limit QuotaTallyTable
sql:/get-quota-tally/update-quota-tally/insert-quota-
tally
```

Tato část nastavení FTP serveru zjišťuje, zda pro daného uživatele či skupinu jsou nastaveny quoty v tabulce ftpquotalimits. Pokud ano, připočítává velikost přenesených dat a odečítá velikost smazaných. Tyto údaje udržuje v záznamu příslušného uživatele v tabulce ftpquotatallies. Pokud uživatel překročí quotu, nedovolí mu FTP server nahrát další obsah, dokud neuvolní potřebné místo.

13. Tvorba aplikace v jazyce PHP

Při tvorbě aplikace jsem nejprve vytvořil adresářovou strukturu. Kořenovým adresářem pro tento projekt bude adresář StHosting. Ten obsahuje veškeré mnou vytvářené a používané adresáře. Adresář phpMyAdmin obsahuje tuto aplikaci pro práci s databází MySQL, v StAdminD je umístěna administrační část aplikace pro správu

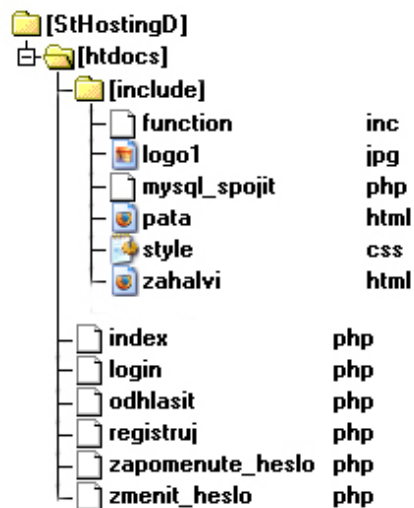
uživatelů. Adresář StHostingD obsahuje uživatelskou část této aplikace. Uživatel má možnost spravovat svá data pomocí FTP klienta a také pomocí online aplikace phpWebFTP, umístěné ve stejnojmenném adresáři. Poslední hlavní adresář je Vhosts. Ten obsahuje adresáře jednotlivých uživatelů, které se díky nastavení serveru Apache budou chovat jako servery.



Obr. 4) Adresářová struktura

13.1 Uživatelská část aplikace

Nejprve vytvořím některé soubory pro často prováděné činnosti, které budou následně vkládány na potřebná místa. Tyto soubory jsou uloženy v adresáři /StHostingD/htdocs/include.



Obr. 5) Struktura uživatelské části aplikace

Prvním z nich je soubor `function.inc`. Jak již název napovídá jedná se o soubor obsahující mnohou definované funkce. Soubor `mysql_spoj.php` obsahuje skripty pro připojení k databázi MySQL, jak je patrné z jeho názvu. Informace, na který počítač, do jaké databáze, pod jakým uživatelským jménem a heslem se má skript připojit jsou nadefinovány v tomto skriptu pevně. Po jeho zavolání se pokusí skript připojit k databázovému systému, pokud se mu připojení nezdaří, vypíše chybové hlášení (Nelze se připojit k databázi). Jestli-že připojení proběhne bez problému, pokusí se zvolit danou databázi, dojde-li ke komplikacím, upozorní nás skript chybovým hlášením (Nelze otevřít databázi). Proběhne-li připojení i volba databáze v pořádku, skript skončí a nevypisuje žádné hlášení.

Dalšími soubory umístěnými do adresáře `/StHostingD/htdocs/include` budou `zahalvi.html` a `pata.html`. Přestože tyto soubory mají příponu `html`, obsahují i skripty v jazyce PHP. Příponu `html` mají z důvodu přehlednosti. Celá tato aplikace je psána tak, že každý soubor, jako například `index.php` obsahuje pouze vlastní tělo, záhlaví a `pata` jsou vkládány pomocí direktivy `include`. Soubor `zahalvi.html` obsahuje potřebné direktivy pro nastavení vlastností obsahu, jako například umístění souboru s nastavením vzhledu `styl.css`. Nastavení znakové sady, informace pro vyhledávače atd.

Soubor `pata.html` je vkládán na konec souborů s vlastním obsahem. Jeho součástí je také položka menu, ve které jsou umístěny všechny odkazy na další potřebné části aplikace. Skripty v souboru `pata.html` zjišťují jedná-li se o uživatele již přihlášeného či nikoli. Pro oba dva typy uživatelů existují odlišná menu. Pro nepřihlášeného uživatele se v menu zobrazí odkazy na registrační formulář, přihlašovací formulář a odkaz zapomněli jste heslo pro zaslání nového hesla na e-mail zadaný při registraci. Uživatel, který je již přihlášený se zobrazí odkazy na změnu hesla, odhlášení, `phpMyAdmin` a úpravu obsahu jeho webových stránek.

Nyní přejdu k popisu jednotlivých souborů uložených v adresáři `/StHostingD/htdocs`. Všechny tyto soubory obsahují direktivy `include` `zahnavi.html`, `include pata.html` a `require_once mysql_spojic.php`, pokud soubor obsahuje skripty pracující s databází.

13.1.1 index.php

Tento soubor obsahuje základní informace o web hostingu, pravidla jeho používání a kontaktní údaje. Je zde nastaven název stránky předaný souboru `zahnavi.html` pomocí proměnné `$navez_stranky`.

13.1.2 registruj.php

Logicky jsem jako další soubor vytvořil `registruj.php` umístěný stejně jako `index.php` v adresáři `/StHostingD/htdocs`. Tento soubor obsluhuje celý proces registrace. Mimo standardně vložených souborů `zahnavi.html` a `pata.html` je zde vytvořen registrační formulář pomocí direktivy `form`. Tento formulář obsahuje zadání jednotlivých uživatelských dat potřebných pro registraci. Obsahuje tlačítko `registrovat`, které odešle zadaná data zpět na tento soubor (`registrace.php`). Tedy jak registrační formulář, tak i jeho zpracování je umístěno v jednom souboru. Podle toho, zda bylo stisknuto tlačítko `registrovat` soubor pozná, jestli má vypsat nový registrační formulář a nebo začít jeho zpracování. V případě, že jsme vyplnili registrační formulář a klikli na tlačítko `registrovat`, skript provede kontrolu

jednotlivých údajů zadaných do formuláře. Kontroluje jestli všechny zadané údaje mají dostatečný počet písmen a neobsahují problematické znaky. U emailu navíc kontroluje jeho správný formát (obsahuje-li znaky @ .). Poslední kontrolou je formát hesla a souhlasí-li s potvrzením hesla. Pokud některá z výše uvedených kontrol neproběhne bez problémů je zavolán formulář pro vyplnění údajů a uživateli je navíc zobrazeno chybové hlášení, určující, který údaj byl zadán chybně. Všechny zadané informace kromě hesel si formulář pamatuje a díky tomu uživatel nemusí znovu zadávat veškeré informace, pouze opraví chybné. Při úspěšném zadání všech údajů a v případě, že uživatelské jméno není již obsazeno, aplikace přidá uživatele do databáze, vytvoří pro něj novou databázi MySQL a nastaví oprávnění pro přístup do databáze. Dále vytvoří v adresáři /StHosting/vhost uživatelský adresář, který má stejný název jako uživatelské jméno. V něm jsou vytvořeny další dva adresáře /htdocs a /cgi-bin. Adresář /htdocs je určen pro uživatelův obsah internetové prezentace a /cgi-bin je náhradní adresář skriptů.

13.1.3 login.php

Soubor login.php obsahuje skripty pro přihlašování již registrovaných uživatelů. I tento soubor obsahuje přihlašovací formulář a zároveň jeho zpracování. Po vyplnění přihlašovacího formuláře a následném kliknutí na tlačítko přihlásit odešle soubor login.php údaje sám sobě. Skript pro přihlášení zkontroluje, zda uživatel zadal všechny potřebné údaje. Pokud proběhla kontrola v pořádku pomocí direktivy require_once mysql_spojit.php se aplikace připojí k databázi MySQL. Do databáze MySQL je zaslán požadavek o vrácení řádku z tabulky, ve kterém uživatelské jméno a heslo odpovídají údajům zadaným ve formuláři login.php. V případě, že zadané údaje neodpovídají žádnému záznamu v tabulce uživatelů, zobrazí se uživateli přihlašovací formulář a chybové hlášení, že uživatelské jméno a heslo neodpovídá. Pokud uživatelem zadané údaje souhlasí se záznamem v databázi jsou tyto údaje přiděleny proměnným typu session, které jsou uloženy na serveru a slouží pro přenášení dat mezi jednotlivými stránkami. Tudíž z nich lze

vyčíst je-li uživatel přihlášen. Jejich platnost je ukončena příkazem v kódu a nebo uzavřením prohlížeče. Poté aplikace přesměruje uživatele na index.php.

13.1.4 odhlásit.php

Soubor odhlásit.php zjistí pomocí `$_SESSION['jmeno']` je-li uživatel přihlášen, pokud ano, zruší tyto proměnné a vypíše hlášení uživateli, že byl úspěšně odhlášen.

13.1.5 zmenit_heslo.php

Zmenit_heslo.php má stejnou strukturu jako předešlé soubory. Obsahuje jednoduchý formulář pro zadání nového hesla a jeho potvrzení. Po kliknutí na tlačítko změnit heslo odešle informace zpět na sám sobě, dojde ke kontrole, zda je nové heslo zadané ve správném tvaru a souhlasí s potvrzením nového hesla. Pokud ano, změní heslo v databázi. Pokud heslo není zadáno ve správném tvaru a nebo nesouhlasí s potvrzením nového hesla, zobrazí se uživateli formulář pro změnu hesla a chybová hláška udávající k jaké chybě došlo.

13.1.6 zapomenute_heslo.php

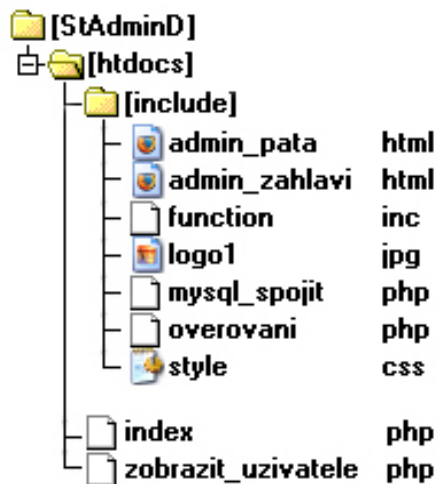
Pro nepřihlášené uživatele, kteří jsou již registrovaní, ale zapomněli své heslo je vytvořen soubor zapomenute_heslo.php. Tento soubor obsahuje jednoduchý formulář, do kterého žadatel o zapomenuté heslo zadá své uživatelské jméno, proběhne kontrola, zda v databázi uživatelů je záznam o existenci takového uživatele. Pokud ano je vygenerováno nové heslo automaticky, toto heslo je uloženo do databáze uživatelů a je odeslán email na adresu uvedenou při registraci uživatele, který uživatele informuje o změně hesla a obsahuje nové heslo. Samozřejmě že uživatel může své heslo poté změnit pomocí odkazu změnit heslo, jakmile je přihlášen pomocí vygenerovaného hesla. Pokud během generování nebo změny hesla dojde ke komplikacím je uživatel informován chybovým hlášením obsahujícím informace o komplikacích.

13.1.7 Správce souborů

Pro správu vlastní webové prezentace může uživatel použít FTP klienta (Total Commander, Krusader atd.), nebo online aplikaci phpWebFTP. Rozhodl jsem se použít tuto aplikaci, protože pokud bych vytvářel vlastní nedosáhnou takové úrovně propracovanosti a kvality služeb. Lze předpokládat, že ji uživatelé použijí pro úpravu obsahu z počítačů, kde nelze použít klasického FTP klienta. Při konfiguraci phpWebFTP jsem upravil pouze volbu jazyku na český a server, ke kterému se má aplikace připojit na mnou používaný. Po přihlášení uživatele v mnou psané aplikaci a kliknutí na odkaz Správce souborů, dojde k přesměrování do adresáře phpWebFTP. Pomocí proměnných session jsou phpWebFTP předány informace potřebné k přihlášení přes FTP a provede se přihlášení.

13.2 Administrační část aplikace

Adresářová struktura administrační části je shodná s uživatelskou částí.



Obr. 5) Struktura administrační části aplikace

Kořenovým adresářem pro administrační část aplikace je /StHosting/StAdminD. Administrační část aplikace obsahuje některé soubory stejné nebo velice podobné jako uživatelská. Jedná se o mysql_spojil.php, který je stejný jako u uživatelské části, pouze používá přístup do databáze přes jiného uživatele. Také soubory

zahlaví_admin.html a pata_admin.html jsou velice podobné souborům použitím v uživatelské části.

13.2.1 overovani.php

Prvním ze souborů tvořených pro administrační část aplikace je overovani.php, jedná se o skript, který zpřístupní administrační část pouze administrátorovy hostingového centra na základě jeho uživatelského jména a hesla.

13.2.2 index.php

Dalším souborem je index.php, který jako všechny soubory obsahuje direktivy include zahlaví_admin.html a pata_admin.html. Také pomocí této direktivy proběhne ověření administrátora. Obsahuje základní informace o administraci této aplikace. Pokud ověření neproběhne správně vyzve aplikace administrátora k zadání správného přihlašovacího jména a hesla.

13.2.3 zobrazit_uzivatele.php

Soubor zobrazit_uzivatele.php je určen pro přehled registrovaných uživatelů a manipulaci s jejich účty. Stejně jako soubor index.php ověří jedná-li se o administrátora oprávněného provádět změny. Pokud ano, pak v tabulce o nastaveném počtu zobrazení provede výpis uživatelů s možnostmi úprav jejich účtů. U každého vypsaného uživatelského účtu je tlačítko smazat, kterým se provede kompletní vymazání účtu. Odstranění adresářové struktury, odebrání práv pro přístup do databáze, smazání databáze, smazání účtu pro přístup přes FTP a odstranění informací o účtu z tabulky uživatelé.

14. Zabezpečení

Zabezpečení web hostingového centra je velice rozsáhlá problematika, která by vydala na samostatnou bakalářskou práci. Při

řešení tohoto problému jsem se alespoň snažil provést co nejlepší zabezpečení základních věcí.

14.1 Přístup do databáze

Z hlediska zabezpečení databází je podstatné, aby uživatelé měli co nejmenší možná práva. Netýká se to pouze uživatelů, kteří se registrují na hostingovém centru, ale také účtů, přes které jsou databáze a uživatelské účty vytvářeny při registraci. Tento účet má proto nastaveny práva pouze na tvorbu databází a účtů, ale ne na jejich odstraňování. Účty registrovaných uživatelů jsou vytvářeny s právy pro přístup pouze do jejich vlastní databáze.

14.2 Zabezpečení FTP

Jediným potřebným zabezpečením z hlediska FTP je omezení přístupu uživatele pouze na jeho vlastního adresáře. Tento problém již řeší sám o sobě program ProFTPD a jeho modul `mod_sql_MySQL`.

14.3 Zabezpečení psané aplikace

Z hlediska zabezpečení psané aplikace lze všechny uživatele rozdělit do tří skupin.

14.3.1 Nepřihlášení uživatelé

První skupinou jsou uživatelé přistupující na stránky hostingového centra, kteří nejsou přihlášení. Zabezpečení pro tyto uživatele spočívá v tom, že je jim zabráněno vstupovat do zabezpečené části bez řádné registrace a přihlášení.

14.3.2 Přihlášení uživatelé

Zabezpečení přístupu do části pro přihlášené uživatele je realizováno pomocí proměnných `session`. Po přihlášení se do těchto proměnných uloží údaje o uživateli.

```
if (isset($_SESSION['jmeno']))
```

 – Pokud je tato podmínka splněna znamená to, že session jmeno existuje a tudíž se jedná o registrovaného uživatele. Pokud neexistuje session jmeno je jasné že uživatel není přihlášen. Podle toho se zobrazují například položky menu atd. Vzhledem k tomu, že proměnné session se přenáší mezi stránkami, na které se prohlížeč přepne, lze takto realizovat zabezpečení celé části a ne pouze jedné stránky.

14.3.3 Správce hostingového centra

Pro zajištění zabezpečení u administrátorské části hostingového centra jsem zvolil metodu porovnání zadaných přihlašovacích údajů s nastaveními údaji přímo v těle skriptu.

```
$overeno = FALSE;  
  
if ( ($_SERVER['PHP_AUTH_USER'] == 'jmeno') AND  
($_SERVER['PHP_AUTH_PW'] == 'heslo') ) { $overeno =  
TRUE; }
```

Na začátku každého souboru administrační části nastavím proměnnou \$overeno na false, tím zajistím, že v případě potenciálního útoku nemůže útočník nastavit tuto proměnnou trvale na hodnotu true. Tento způsob zabezpečení je vhodný v případě, že není zapotřebí odlišit více administrátorů. Také je nezbytně nutné chránit zdrojové kódy programu, protože obsahují přímo přístupová hesla.

14.4 Ostatní zabezpečení

Další důležitou částí zabezpečení je oddělení virtuálních serverů uživatelů od virtuálních serverů pro phpMyAdmin, phpWebFTP a samotné aplikace hostingů. Problémem je nastavení php_openbasedir, což je nastavení nejvyššího adresáře, do kterého mají přístup skripty php jednotlivých uživatelů. Protože pro všechny virtuální servery vytvořené pomocí mod_vhost_alias je nastaven php_openbasedir /StHosting/Vhost musí mít části zajišťující funkci a správu hostingového centra umístěné

mimo tento adresář. Pokud by tomu tak nebylo, je zde velké bezpečnostní riziko. Tento problém lze vyřešit použitím jiného modulu Apache a nebo doplněním kódu modulu. Mě se však tento problém vyřešit nepodařilo. Pro všechny mnou psané skripty manipulující s adresáři či soubory jsem používal pouze funkce přístupující přes FTP, díky tomu lze na serveru zakázat funkci php copy. Tato funkce je na většině běžně dostupných web hostingách zakázána, protože představuje bezpečnostní riziko. Pokud ji zakážeme, vyřeší se tak i problém s php_openbasedir, protože nelze kopírovat obsah adresářů jinak než přes FTP. Přes FTP nelze vstoupit nad rámec vlastního adresáře.

15. Závěr

V první části této bakalářské práce jsem se zabýval uvedením do problematiky web hostingu. Uvedl jsem v ní kritéria, podle kterých lze porovnávat jednotlivé web hostingové služby. Dále pak porovnání dostupných web hostingových služeb, komerčních i bezplatných a volbou jedné varianty v každé kategorii.

V další části jsem popisoval předpokládaný postup mé práce, volbu technologií a výskyt možných komplikací. Uvedl jsem v ní také popis dostupných technologií použitelných při tvorbě web hostingového centra. Přibližnou mnou vytvořenou architekturu a popis její funkce.

V poslední části jsem se zabýval konečnou implementací programu na spravování web hostingového centra, jeho vnitřní strukturou a vysvětlením funkce jednotlivých částí. Popisoval jsem možnosti technologií serveru Apache a mnou zvolená řešení. Dále také použití databázového systému MySQL a jeho konfiguraci. Volbu vhodného FTP serveru ProFTPD s modulem mod_sql_MySQL, který zajišťuje automatické vytváření FTP účtů. Důvody, které mě vedly k použití aplikace phpWebFTP, která umožňuje uživatelům on-line správu jejich dat. Značný prostor jsem v této části věnoval také otázce bezpečnosti,

popisu mnou použitých bezpečnostních opatření a také popisem slabých míst systému.

Závěrem své práce bych také rád poděkoval vedoucímu mé práce Ing. Lukáši Čeganovi za rady, které mi poskytl a za čas, který mi věnoval.

SEZNAM POUŽITÉ LITERATURY

1. MOHAMMED, K. *Apache server 2*. 1. vyd. Brno: Computer Press, 2004. ISBN 80-251-0319-6
2. ULLMAN, L. *SQL Hotová řešení*. 1. vyd. Brno: Computer Press, 2003. ISBN 80-7226-975-5
3. LACKO, L. *PHP a MySQL*. 1. vyd. Brno: Computer Press, 2004. ISBN 80-251-0063-4
4. Apache.org. [online]. c2006 [cit. 2007-05-05]. Dostupný z WWW: <<http://www.apache.org/>>
5. The PHP Group. *PHP: Hypertext Preprocessor*. [online]. c2001-2007, 14.5.2007 [cit. 2007-05-15]. Dostupný z WWW: <<http://www.php.net/>>
6. ProFTPD.org. [online]. c2006 [cit. 2007-05-05]. Dostupný z WWW: <<http://www.proftpd.org/>>
7. ProWebFTP.com. [online]. c2006 [cit. 2007-05-05]. Dostupný z WWW: <<http://www.prowebftp.com/>>
8. Virtual Hosting With Proftpd And MySQL. *Howto forge* [online]. 2005 [cit. 2007-05-05]. Dostupný z WWW: <http://www.howtoforge.com/proftpd_mysql_virtual_hosting>
9. PURYEAR, Dustin. Apache vs. IIS. TechX World [online]. 2007 [cit. 2007-05-05]. Dostupný z WWW: <<http://techxworld.com/community/blogs/features/archive/2007/02/25/apache-vs-iis.aspx>>.

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	Studentské web hostingové centrum
Autor práce	Bačkovský Josef
Obor	Informační technologie
Rok obhajoby	2007
Vedoucí práce	Ing. Lukáš Čegan
Anotace	Cílem této práce je vytvoření studentského web hostingového centra a jeho uvedení do provozu. Jsou v ní popsány používané technologie v oblasti web hostingu, kritéria, podle kterých lze web hostingy porovnávat a přehled dostupných web hostingů. Obsahuje také popis konfigurace jednotlivých použitých technologií, popis implementační části vytvářené aplikace v jazyce php a základní bezpečnostní mechanismy.
Klíčová slova	Apache, IIS, phpMyAdmin, ProFTPD, MySQL, SMTP, phpWebFTP, web hosting, server hosting, hosting, server, DNS, mod_vhost_alias, mod_sql_MySQL,