

E-mailový klient přes WWW rozhraní

Tomáš Bubela

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Vložit oficiální zadání diplomové/bakalářské práce

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 08. 06. 2005

.....

Tomáš Bubela

Abstrakt

Webmail je aplikace napsaná v PHP4. Hlavními funkcemi jsou odesílání, přijímání a práce s emaily a složkami. Podporuje protokoly POP3 a SMTP. Je založena na databázi MySQL 4.1. Zobrazení je možné pomocí jakéhokoli webového prohlížeče, který podporuje HTML 4.0 a kaskádové styly CSS2.

Summary

Webmail is application written in PHP4. Main functions are sending, receiving and working with emails and folders. It supports protocol POP3 and SMTP. It is based on database MySQL 4.1. Viewing is possible via any web browser, which supports HTML 4.0 and cascading style sheets CSS2.

OBSAH

ÚVOD	7
I. TEORETICKÁ ČÁST	8
1 PHP	10
1.1 Historie PHP	10
1.1.1 PHP/FI	10
1.1.2 PHP 3	10
1.1.3 PHP 4	11
1.1.4 PHP 5	12
2 MYSQL	13
2.1 Co je to databáze?	13
2.2 Co je to SQL?	13
2.3 Co je to MySQL?	14
2.4 Jak to vše souvisí s PHP?	15
3 CSS (CASCADING STYLE SHEETS)	16
3.1 Historie CSS	16
3.2 Strategie použití CSS	18
II. PRAKTICKÁ ČÁST	21
4 ADMINISTRÁTORSKÁ SEKCE	22
4.1 Struktura databáze	22
4.2 Popis tabulek databáze	22
4.2.1 cis_fakulta	22
4.2.2 cis_role	23
4.2.3 cis_styl	23
4.2.4 mailboxes	23
4.2.5 messages	24
4.2.6 user	25
4.3 Nastavení programu (config.php)	25
4.4 Funkce (functions.php)	26
4.5 Soubory s jazyky (czech.php, english.php, language.php)	27
4.6 Přihlášení (login.php)	27
4.7 Seznam emailů (list.php)	28
4.7.1 Zjištění nových emailů	29
4.8 Psaní emailů (compose.php)	29
4.9 Prohlížení emailu (view.php)	30
4.10 Nastavení složek (folder.php)	30
4.11 Nastavení stylů (styl.php)	31
4.12 Nastavení programu (settings.php)	31
4.13 Odhlášení (logout.php)	31
4.14 Přejít na další rok	31
5 UŽIVATELSKÁ SEKCE	32
5.1 Přihlášení uživatele	32
5.2 Zobrazení seznamu zpráv	32
5.3 Čtení zpráv	34
5.4 Psaní zpráv	35
5.5 Nastavení složek	35
5.5.1 Přejmenování složky	36
5.5.2 Smazání složky	36
5.5.3 Vytváření složky	36
5.6 Nastavení stylů	37

5.7	Nastavení programu	37
5.8	Odhlášení uživatele.....	38
ZÁVĚR.....		39
SEZNAM POUŽITÝCH ZDROJŮ		40
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		42
SEZNAM OBRÁZKŮ.....		43
SEZNAM PŘÍLOH.....		44

ÚVOD

Mým úkolem bylo zpracovat emailového klienta přes webové prostředí, aby se uživatel mohl připojit na školní email odkudkoli, kde je dostupný internet. K řešení tohoto problému jsem zvolil kombinaci jazyků PHP a MySQL za přispění kaskádových stylů CSS2, JavaScriptu a komponent PEAR.

Hlavní funkce měly být podobné emailovým klientům jako jsou například Microsoft Outlook, Mozilla Thunderbird nebo Pegasus Mail. Hodně jsem se těmito programy nechal inspirovat a do aplikace jsem zahrnul funkce přijímání a odesílání emailu, třídění emailů do složek, práci se složkami a změnu stylů.

Protože jsem zjistil, že emailové servery jsou poněkud pomalé, využil jsem uložení dat do databázového prostředí MySQL. Databáze umožňují snadnou správu emailů, různé kritéria pro třídění nebo třeba vyhledávání.

Práci jsem se snažil psát spíše ve stylu manuálu. V teoretické části jsem napsal něco o PHP jazyku SQL a kaskádových stylech CSS2. Praktickou část jsem rozdělil na dvě sekce. První je brána z pohledu administrátora. Obsahuje popis hlavních nastavení, strukturu databáze, popis funkcí, které se v programu vyskytují a popis jednotlivých skriptů s jejich vysvětlením. Na druhou sekci jsem se snažil podívat z pohledu méně zkušeného uživatele, spíše začátečníka. Pokusil jsem se, snad názorně, vysvětlit základní funkčnost.

I. TEORETICKÁ ČÁST

1 PHP

1.1 Historie PHP

1.1.1 PHP/FI

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval 'Personal Home Page Tools'. Protože byla třeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro Web. Rasmus se rozhodl uvolnit zdrojový kód PHP/FI pro všechny, takže kdokoli ho může používat, stejně jako opravovat chyby a vylepšovat kód.

PHP/FI, což znamená Personal Home Page / Forms Interpreter, obsahovalo něco ze základní funkcionality PHP, jak ho známe dnes. Mělo proměnné perlovského typu, automatickou interpretaci formulářových proměnných a syntaxi vloženou do HTML. Syntaxe samotná byla podobná jazyku Perl, přestože mnohem omezenější, jednodušší a v něčem nekonzistentní.

V roce 1997 se PHP/FI 2.0, druhá implementace psaná v C, stala kultovní záležitostí pro (odhadem) tisíce uživatelů po celém světě, a s přibližně 50.000 doménami oznamujícími nainstalované PHP/FI, což čítalo zhruba 1 % všech domén na Internetu. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

PHP/FI 2.0 bylo oficiálně uvolněno až v listopadu 1997, poté co strávilo většinu svého života v betaverzích. Krátce nato bylo následováno první alfa verzí PHP 3.0.

1.1.2 PHP 3

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný celek, poté co shledali PHP/FI 2.0 výrazně "poddimezované" pro vývoj svých aplikací. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI,

rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0, a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. K poskytnutí pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.

Nový jazyk byl uvolněn pod novým názvem, který odstranil implikaci omezeného osobního použití, kterou neslo označení PHP/FI 2.0. Byl nazván pouze 'PHP', což je rekurzivní akronym - PHP: Hypertext Preprocessor.

Na konci roku 1998 vyrostlo PHP do rozsahu instalací v řádu (odhadem) desítek tisíc uživatelů a stovek tisíc Webů. V době svého vrcholu bylo PHP 3.0 instalováno na přibližně 10 % všech WWW serverů na Internetu.

PHP 3.0 bylo oficiálně uvolněno v červnu 1998, poté co strávilo cca 9 měsíců ve veřejném testování.

1.1.3 PHP 4

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přespání jádra PHP. Cílem návrhu bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze PHP. Takové aplikace byly schopny pracovat s PHP 3.0 (díky novým možnostem a podpoře široké škály databází a API od jiných tvůrců), ale PHP 3.0 nebylo navrženo pro efektivní práci tak náročných aplikací.

Nový engine, nazvaný 'Zend Engine' (sestaven z jejich křestních jmen, Zeev a Andi), úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze, přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

1.1.4 PHP 5

PHP 5 zlepšuje PHP 4 zejména v těchto důležitých oblastech:

- objektově orientované programování
- databáze
- XML

Objektově orientované programování v PHP 5

V PHP5 je nový objektový model. Manipulace PHP 5 s objekty je kompletně přepsána, má lepší výkonnost a více možností. Byly přidány (upraveny) následující funkce:

- konstruktory
- destruktory
- přidání viditelnosti do proměnných a metod
- statické proměnné a metody
- konstanty v objektech
- abstraktní třídy
- interfaces – rozhraní
- přetěžování metod `__get`, `__set`, `__call`
- iterace přes datové členy
- metoda `__toString` pro převod do řetězce
- magické metody `__sleep` a `__wakeup`
- klíčové slovo `final`
- klonování objektů
- reflection API pro zpětnou analýzu objektů

Co se týká objektů, je tedy novinek více, než dostatek. Při pročítání manuálu se člověk neubrání dojmu, že právě v oblasti objektově orientovaného programování jsou největší změny v PHP 5 oproti PHP 4. S objekty se v PHP 5 zkrátka zachází úplně jinak. PHP 5 je momentálně poslední uvolněnou verzí PHP.

2 MYSQL

2.1 Co je to databáze?

Databázi si můžeme představit jako prostor, do kterého se ukládají všechny potřebné údaje. Zpracováním a přístupem údajů v databázi bývá pověřen program, kterému se říká DBMS (DataBase Management System), nebo česky SŘBD (Systém Řízení Báze Dat). Celé je to tedy zařízeno tak, že veškeré zpracování, ukládání a získávání dat je prováděno DBMS programem. Jakákoli aplikace, která databázi používá vždy přistupuje přes tento DBMS program.

Mezi DBMS patří řada programů v nejrůznějších cenových kategoriích. Většina profesionálních a komerčních DBMS nejsou programy nikterak levné a jejich cena se pohybuje ve velmi vysokých částkách. Naštěstí existují v kategorii DBMS i programy, které jsou v nízké cenové relaci, nebo zcela zdarma. Představitelem těchto DBMS systémů jsou například MySQL, PostgreSQL, nebo Firebird.

Naprostá většina dnešních DBMS staví na tzv. relačním modelu dat. V tomto modelu jsou veškerá data uspořádána do databázových tabulek. Každá tabulka zpravidla uspořádává údaje o určitém objektu. Databázová tabulka je uspořádána do řádků a sloupců. Například budeme-li mít tabulku se seznamem osob, bude každý řádek odpovídat jedné osobě. Sloupce pak obsahují konkrétní údaje o osobách, v našem případě by to třeba mohlo být: Jméno, Rodné číslo, Adresa, Telefon.

2.2 Co je to SQL?

Jak už jsem zmínil, přístup k údajům uloženým v databázi obstarává DBMS program. Aby mohly jiné aplikace používat také údaje z databáze, musí DBMS program nabízet rozhraní, pomocí kterého budou moci ostatní aplikace s DBMS spolupracovat.

Při práci s DBMS programem se používá osvědčený model klient/server. V roli serveru vystupuje DBMS program nabízející své služby. DBMS program je tak nepřetržitě spuštěn a čeká na požadavky od klientů - jiných aplikací. Na tyto požadavky pak odpovídá a reaguje na ně. Protože DBMS program vystupuje v roli serveru, říká se mu také databázový server.

Pro zápis požadavků na databázový server se dnes nejčastěji používá SQL jazyk. SQL je anglická zkratka znamenající Structured Query Language. Tento jazyk dnes používá naprostá většina databázových serverů. Proto se takovým serverům říká zjednodušeně SQL servery. Samotný SQL obsahuje vše potřebné pro práci s databázemi - ať už příkazy pro práci na vytvoření, rušení, modifikování tabulky, tak i pro práci s údaji samotnými – přidávání, změnu, rušení a vyhledávání údajů.

Jazyk SQL má za sebou poměrně dlouhý vývoj. Jeho prototyp by poprvé implementován v roce 1974 společností IBM. Od té doby má za sebou SQL bouřlivý vývoj, a dá se říci, že jazyk SQL se dodnes stále vyvíjí, hlavně co se týká přidávání nových technologií. Skutečným faktickým standardem se stal SQL až v roce 1986, kdy se dočkal standardizace díky organizaci ANSI. Od té doby se tedy SQL stal univerzálním nástrojem pro programování databází. Dnes se každý, kdo se dostává do styku s vývojem, či laděním databází, musí seznámit s jazykem SQL, který je prakticky u databází všudypřítomný.

Pomocí jazyka SQL tedy říkáme, co chceme v databázi provést. Jazyk SQL se skládá ze dvou základních částí, jednak z části DDL, a jednak z části DML. Příkazy DDL jsou zkratkou za Data Definition Language, a slouží k definici a modifikaci tabulek a dalších databázových objektů. Příkazy DML jsou zase zkratkou za Data Manipulation Language a slouží již opravdu pro manipulaci s databázovými údaji.

2.3 Co je to MySQL?

Jednoduše se dá říci, že MySQL je databázový systém. Takových systémů existuje celá řada, například Oracle, MS SQL, nebo třeba Sybase. MySQL je systém, který se etabloval především ve webových aplikacích, a který je dost preferovaný při spolupráci s PHP.

Samo o sobě patří MySQL spíše k těm jednoduchým databázovým systémům. Ve své podstatě je MySQL ořezaný o některé možnosti, kterou mají jiné databázové systémy. Důsledkem ořezání je nenáročnost MySQL na zdroje počítače a zvýšení rychlosti u některých operací. Jednoduše shrnuto, MySQL je malý, rychlý, jednoduchý a nenáročný databázový systém.

MySQL patří mezi databázové systémy, které vyvolávají mohutné diskuse pod články. Někteří lidé MySQL zatracují, jiní jí zase mohutně chválí. Pravdou je, že MySQL nemá takové schopnosti a možnosti jako mnohé velké databázové systémy, ale MySQL do

této oblasti ani nemíří. MySQL je projektován pro jednoduché databáze, byť třeba s obrovskou spoustou údajů.

2.4 Jak to vše souvisí s PHP?

V roli klienta pro databázový server může jako aplikace vystupovat i PHP. Tedy PHP se může spojit s databázovým serverem a využívat jeho služby, požadovat zápis, nebo získání určitých údajů z databáze. To znamená, že naše PHP skripty mohou obsahovat příkazy zapsané v jazyce SQL a PHP může zpracovávat výsledky SQL po provedení na databázovém serveru.

V praxi má každý SQL server svůj vlastní protokol, a proto PHP musí umět s každým SQL serverem komunikovat trochu jinak. Ale jak už jsem napsal, PHP umí komunikovat s mnoha SQL servery, mimo jiné třeba s Oracle, Sybase, Informix, MySQL, PostgreSQL.

3 CSS (CASCADING STYLE SHEETS)

3.1 Historie CSS

Oddělení struktury dokumentu od jeho layoutu (formátování) bylo cílem HTML od samých počátků v roce 1990. Již Tim Berners-Lee, který je považován za otce WWW, navrhl svůj první prohlížeč NeXT tak, že formátování dokumentu bylo řízeno jednoduchým zabudovaným stylovým předpisem. Berners-Lee však jazyk tohoto stylového předpisu nikdy nepublikoval a spoléhal na autory prohlížečů, že tuto koncepci sami dále rozvinou. K tomu však, bohužel, nedošlo. První komerční prohlížeč, NSCA Mozaic z roku 1993, naopak oklestil možnosti uživatelského či autorského stylování HTML dokumentů na minimum.

Jak popularita Webu stoupala, stále více autorů si stěžovalo, že nemohou vzhled svých HTML dokumentů nijak ovlivnit. To, co již v té době bylo samozřejmostí nejen u DTP programů, ale i u běžných textových editorů, tedy např. změna velikosti, barvy či řezu písma, nastavení barvy pozadí atd., v HTML dokumentech nebylo možné. Na tuto kritiku přišly skoro současně dvě reakce.

Těsně před konferencí o WWW konané v listopadu 1994 v Chicagu publikoval Hakon Wium Lie první koncept "Cascading HTML Style Sheets". Ačkoli návrh nebyl zdaleka ještě dokonalý, stal se základem, ze kterého vycházejí kaskádové styly dodnes, a podpořil ho tehdy i Dave Raggett, hlavní architekt HTML 3.0.

Jen o tři dny později, než Hakon Wium Lie publikoval svůj návrh, se objevila první verze prohlížeče Netscape, která na všeobecnou potřebu stylovat HTML dokumenty reagovala po svém. Místo implementace stylového jazyka totiž Netscape zabudoval základní formátovací prostředky přímo do jazyka HTML. Jak se záhy ukázalo, pro vývoj webu to znamenalo téměř katastrofu, neboť střet těchto dvou koncepcí zásadně poznamenal kompatibilitu a použitelnost webových dokumentů na mnoho let dopředu a jeho negativní důsledky neseme dodnes.

O kaskádových stylech se od vydání prvního návrhu ještě dlouze diskutovalo, přičemž hlavním předmětem diskusí nebyl ani tak vlastní jazyk, jako spíš koncepce, která poslední slovo o formátování dokumentu přenechávala uživateli, nikoli autorovi (uživatelské stylové předpisy). Zastánci této myšlenky, mezi které patřil i autor původního návrhu Hakon Wium Lie, však nakonec zvítězili a v prosinci 1996 spatřila světlo světa

Specifikace kaskádových stylů 1, již jako doporučení konsorcia W3C, které začalo pracovat v roce 1995 a kaskádové styly vzalo pod svá křídla jako nedílnou součást rozvoje jazyka HTML.

Konsorcium W3C, resp. jeho příslušná pracovní skupina, pak kaskádové styly dále rozvíjelo a rozvíjí dodnes. V květnu 1998 tak bylo vydáno doporučení Specifikace CSS 2, která je základem této knihy. Nyní probíhají práce na verzi CSS 3. Svůj vývoj prodělaly i implementace kaskádových stylů v prohlížečích. První komerční prohlížeč s alespoň částečnou podporou CSS 1 byl Internet Explorer 3 (IE3) společnosti Microsoft, vydaný v srpnu 1996. V té době však samotná Specifikace CSS 1 nedospěla do stádia oficiálního Doporučení W3C, a tak se implementace v IE3 se specifikací zcela neshoduje. IE3 má poměrně spolehlivou podporu pro vlastnosti barev, pozadí a textu, prakticky však opomíjí koncepci boxů.

Dalším prohlížečem na řadě byl Netscape Navigátor 4.0 (NN4). Jelikož byla společnost Netscape (na rozdíl od Microsoftu) od počátku ke kaskádovým stylům značně skeptická, je implementace v NN4 spíše důsledkem konkurenčního boje než pečlivé přípravy. Podpora CSS je tak v NN4 sice zdánlivě rozsáhlá, avšak velmi chybová, a některé z teoreticky podporovaných vlastností jsou prakticky nepoužitelné. Netscape navíc interně implementuje CSS prostřednictvím JavaScriptu, takže pokud je JavaScript v NN4 vypnut, nejsou aktivní ani kaskádové styly.

Od té doby jsou již nové implementace CSS záležitostí Microsoftu a od roku 1998 též prohlížeče Opera (od verze 3.5). Vývoj Netscapu se na dlouhá léta zastavil a byl oživen až po otevření zdrojových kódů a převzetím vývoje komunitou nezávislých vývojářů. Tak vzniklo vykreslovací jádro Gecko rozvíjené do současnosti jako otevřený kód (open source), které zpětně převzala společnost Netscape a zabudovala ho do svých nejnovějších verzí 6 a 7.

Ani přes dlouhou historii se zatím bohužel nepodařilo žádnému výrobcí dotáhnout implementaci CSS 2 do zdárného konce. Posledních pár let platí, že vždy ta nejaktuálnější verze z trojice hlavních prohlížečů (Microsoft Internet Explorer, otevřený kód Gecko/Mozilla a Opera) je s implementací nejdál. V době vydání této knihy je to tedy prohlížeč Mozilla ("výkladní skříň" jádra Gecko), kterému odpovídá Netscape 7. Publikována však již byla beta verze Opery 7, která Mozillu/NN7 možná předběhne, a otázkou je, čím nás překvapí Internet Explorer 7, který časem jistě také přijde.

3.2 Strategie použití CSS

Kaskádové styly lze použít různým způsobem a v různém rozsahu. V praxi se nejčastěji uplatňují tři strategie použití:

- obohacení formátovacích možností
- veškeré formátování s výjimkou rozmístění (layoutu) prvků
- formátování jen pomocí CSS, včetně rozmístění prvků

Žádnou z těchto strategií nelze automaticky označit za lepší či nejlepší. Záleží na podmínkách konkrétního projektu a úsudku webdesignéra, kterou ze strategií zvolí a proč. Abychom vám rozhodování usnadnili, podívejme se na jednotlivé strategie podrobněji. Všechny tři strategie jsou dále popsány ve své ideální podobě. Ve své práci však ve skutečnosti můžete použít různé kombinace a varianty podle vlastních potřeb.

Obohacení formátovacích možností

Jedná se zřejmě zatím o nejrozšířenější strategii využívání CSS. Setkáte se s ní dnes již téměř všude, především na komerčních stránkách s vysokou návštěvností (katalogy stránek, on-line verze deníků atd.).

Základem je tradiční HTML dokument využívající značky starších norem jazyka (HTML 3.2, 4.0), včetně značek formátovacích (**, *<i>*, *<i>* atd.). Rozvržení (layout) stránky je tvořeno tabulkami (značky *<table>*, *<tr>* a *<td>*), jejichž atributy většinou určují i barvu písma a pozadí (atributy *color* a *background* prvku *<td>*) jednotlivých částí stránky. K dosažení designérských cílů a shodného vzhledu v různých prohlížečích se často používá nejen odmítnutých (deprecated) či zastaralých značek a atributů, ale i prvků proprietárních, zavedených výrobcí jednotlivých prohlížečů, bez jakékoli opory ve standardech (např. atributy *marginwidth* a *marginheight* značky *<body>*, či atribut *height* značky *<table>*).

Kaskádové styly jsou použity pouze jako doplněk k dosažení těch formátovacích efektů, které standardními, ani proprietárními prostředky HTML vytvořit nelze. Nejčastěji se jedná např. o určení velikosti písma v pixelech, stylování formulářových prvků (barva, orámování a velikost tlačítek a vstupních polí) nebo dynamické efekty přechodu ukazatele myši nad odkazem (*a:hover*) či kombinující CSS s JavaScriptem (rozbalovací menu). Častá je též aplikace proprietárních vlastností CSS, např. stylování posuvníků zavedené v MS Internet Exploreru pro Windows od verze 5.5.

Při této strategii jsou stylové předpisy obvykle deklarovány v hlavičce dokumentu (embedded style sheet), značkou `<sty/e>`, nebo dokonce přímo v jednotlivých prvcích (atributem style). Využívá se však i tříd, a to nejčastěji pro definování velikosti, typu a barvy písma. Tyto třídy bývají deklarovány v externím stylovém předpisu a v dokumentu se pak explicitně vážou ke každému příslušnému prvku (zpravidla `<td>`), neboť dědičnost a kontextové selektory obvykle nelze využít z důvodu nezřetelné (či dokonce chybné) struktury obsahu dokumentu.

Výhody

- U webových dokumentů vytvořených tímto postupem lze dosáhnout shodného, resp. velmi podobného zobrazení ve všech zatím nejrozšířenějších prohlížečích, včetně prohlížečů zastaralých (MSIE4/Win, NN4), a to i v případě poměrně graficky náročného designu. S postupným mizením starších prohlížečů a naopak s nástupem moderních zařízení (palmtopy, mobilní telefony) se tato výhoda neustále zmenšuje.
- umožňuje postupný přechod na pokročilejší strategie využití CSS u dříve vytvořených stránek bez komplexního redesignu
- webdesignér nemusí ovládat do detailů kompletní problematiku kaskádových stylů, vystačí s několika jednoduchými prvky (vlastnosti font-family, font-size, co/or, selektor a:hover atd.) či hotovými řešeními (skripty na dynamická menu)
- v případě nasazení externích stylových předpisů a tříd (např. pro stylování písma) se lze již částečně těšit výhodám snadné správy stránek a hromadných změn stylu

Nevýhody

- Vytvořené webové dokumenty jsou obvykle nevalidní (odporující platným normám). To sice obecně nemusí nejrozšířenějším prohlížečům vadit, nicméně obtížně se pak kontroluje, zda zdrojový kód neobsahuje skutečně závažné chyby.
- mícháním formátovacích značek HTML a kaskádových stylů vzniká riziko defektního

zobrazení, někdy vedoucího až k nepřístupnosti (nečitelnosti) obsahu. Typickým příkladem

méně závažného defektu je nekonzistentní velikost písma v případě současného používání

značky `<font>` a CSS vlastnosti `font-size` s udanou velikostí v pixelech. Hrozbu znepřístupnění obsahu představuje např. nevhodná kombinace atributu `bgcolor` prvku

`<td>` s CSS vlastností `color` — prohlížeč nepodporující CSS (třeba NN4 s vypnutým

JavaScriptem) pak může zobrazit zcela nečitelnou kombinaci barvy písma a pozadí.

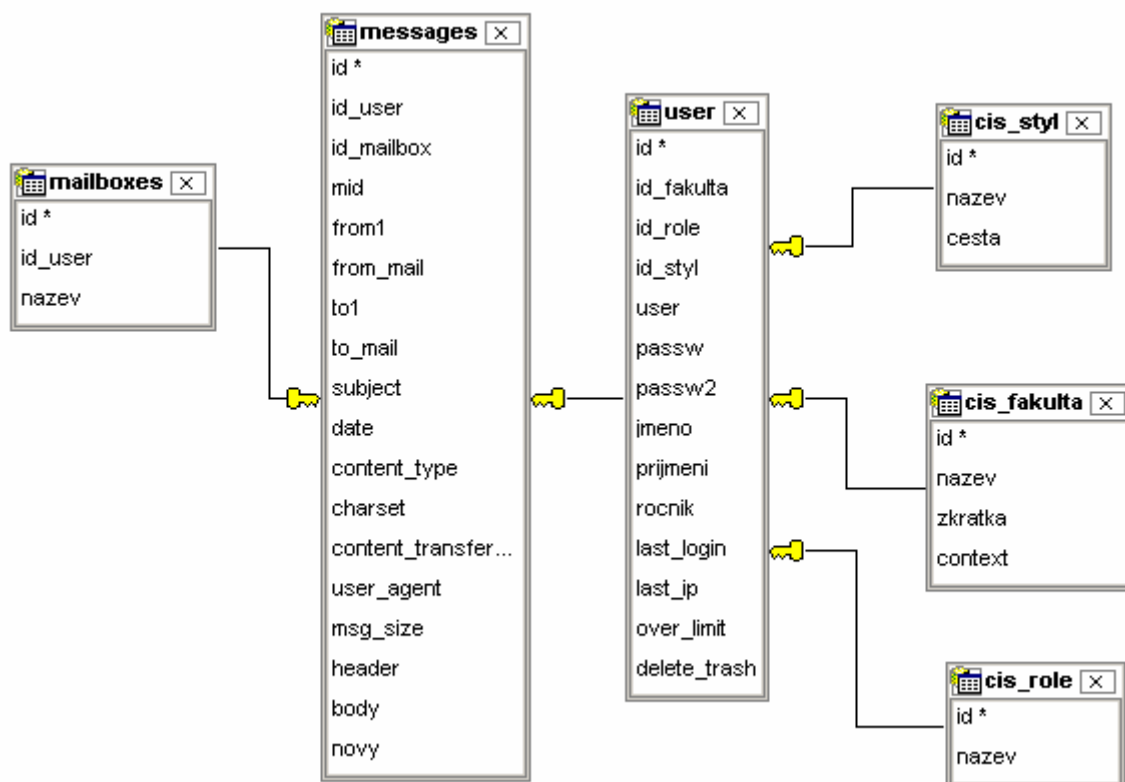
- tato strategie s sebou obvykle nese i hojně používání mnohonásobně vnořovaných tabulek nejen pro základní rozvržení stránky, ale i pro přesné umístění dílčích prvků (např. řezaná grafika) a doplňkových grafických efektů (orámování boxů atd.). Důsledkem je snížená přístupnost obsahu dokumentu v textových prohlížečích a pro hendikepované uživatele (zejména nevidomé). V kombinaci s pevně definovanými šířkami (celé stránky i jednotlivých prvků) navíc dochází ke znepřístupnění dokumentu i pro uživatele malých displejů (především palmtopů a mobilních telefonů).
- hlavní výhody CSS, tj. snadná správa dokumentu a přínosy oddělené struktury obsahu a formátování, lze využít jen ve velmi omezené míře. Dokumenty vytvořené touto metodou jsou nepřehledné a obtížně upravovatelné.
- datový objem zdrojového HTML kódu většinou mnohonásobně překračuje objem samotného obsahu a zpomaluje se tak načítání stránek, zejména při pomalejším připojení.

II. PRAKTICKÁ ČÁST

4 ADMINISTRÁTORSKÁ SEKCE

4.1 Struktura databáze

Struktura databáze je zobrazena na Obr. 1.



Obr. 1. Struktura databáze

4.2 Popis tabulek databáze

4.2.1 cis_fakulta

Jedná se o číselník, ve kterém jsou uchovány informace o fakultě.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
nazev	varchar(64)	Název fakulty
zkratka	varchar(10)	Zkratka fakulty
context	varchar(160)	Kontext fakulty

4.2.2 cis_role

Jedná se o číselník, ve kterém jsou uchovány informace o rolích pro pozdější použití do administrátorské sekce.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
nazev	varchar(32)	Název role (administrátor, uživatel, atd.)

4.2.3 cis_styl

Jedná se o číselník, ve kterém jsou uchovány informace o stylech.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
nazev	varchar(64)	Název stylu
cesta	varchar(250)	Cesta k souboru se stylem. Jedná se o relativní cestu, tj. <i>/css/default.css</i>

4.2.4 mailboxes

Tabulka, ve které jsou informace o složkách.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
id_user	int(9) unsigned	Id uživatele, kterému daná složka náleží
nazev	varchar(128)	Název složky

4.2.5 messages

V této tabulce jsou uchovány všechny emaily.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
id_user	int(9) unsigned	Id uživatele, kterému daná složka náleží
id_mailbox	int(9) unsigned	Název složky
mid	varchar(128)	Message-ID
from1	varchar(160)	Skutečné jméno odesilatele, od kterého email přišel
from_mail	varchar(160)	Odesílatelův email
to1	varchar(160)	Skutečné jméno příjemce, kterému byl email poslán
to_mail	varchar(160)	Příjemcův email
subject	varchar(160)	Předmět
date	datetime	Datum příchodu emailu
content_type	varchar(160)	Kódování textu
charset	varchar(160)	Znaková sada
content_transfer_encoding	varchar(160)	Kódování příloh
user_agent	varchar(160)	Název emailového klienta
msg_size	bigint(20)	Velikost zprávy
header	longblob	Hlavičky emailu
body	longblob	Celé tělo emailu
novy	char(1)	Indikuje, jestli je email již přečtený nebo ne
is_mime	char(1)	Indikuje, jestli se jedná o skutečný email nebo pouze uložený text pro pozdější zpracování

4.2.6 user

Tabulka, uchovávající informace o uživateli a jejich nastaveních.

Název	Typ	Popis
id	int(9) unsigned	Auto_increment id
id_fakulta	int(9) unsigned	Odkazuje do tabulky cis_fakulta
id_role	int(9) unsigned	Odkazuje do tabulky cis_role
id_styl	int(9) unsigned	Odkazuje do tabulky cis_styl
user	varchar(30)	Uživatelské jméno
passwd	varchar(50)	Heslo pro přístup do databáze
passwd2	varchar(50)	Heslo pro přístup na FTP server
jmeno	varchar(30)	Skutečné jméno uživatele
prijmeni	varchar(30)	Skutečné příjmení uživatele
rocnik	smallint(1)	Ročník, do kterého uživatel náleží. Hodnota 0 slouží pro učitele.
last_login	datetime	Datum posledního přihlášení
last_ip	varchar(64)	Poslední IP adresa, ze které se uživatel přihlašoval
over_limit	char(1)	Indikuje příznak, kdy uživatel překročí limit velikosti schránky nebo počtu zpráv
delete_trash	char(1)	Indikuje příznak, kdy chce uživatel při odhlašování vyprázdnit koš
delete_sent	char(1)	Indikuje příznak, kdy chce uživatel při odhlašování vymazat emaily ze složky Odeslané

4.3 Nastavení programu (config.php)

Většina nastavení lze provést v souboru *include/config.php*.

- **\$NazevMailu (string)**: Hodnota, která se bude zobrazovat v záhlaví programu.
- **\$SQLServerName (string)**: Název serveru, na kterém běží databáze MySQL.
- **\$SQLUserName (string)**: Uživatelské jméno pro přístup do databáze.
- **\$SQLPassword (string)**: Heslo pro přístup do databáze.

- **\$SQLDatabase (string)**: Název databáze.
- **\$SMTPServer (string)**: Název SMTP serveru.
- **\$SMTPPort (integer)**: Číslo SMTP portu. Standardně bývá nastavena hodnota 25.
- **\$POPServer (string)**: Název POP3 serveru.
- **\$POPPort (integer)**: Číslo POP3 portu. Standardně bývá nastavena hodnota 110.
- **\$Domain (string)**: Doména, ze které se odesílá mail (např.: “utb.cz”).
- **\$Copyright (string)**: Řetězec, ve kterém jsou informace o vlastníkovi licenčních práv.
- **\$EmailsPerPage (integer)**: Počet emailových zpráv na jedné stránce.
- **\$DobaOdlognuti (integer)**: Udává dobu automatického odhlášení uživatele při nečinnosti delší než \$DobaOdlognuti. Hodnota se zadává v sekundách.
- **\$MaxMessagesCount (integer)**: Maximální počet zpráv pro uživatele.
- **\$MaxMessagesSize (integer)**: Maximální celková velikost zpráv uživatele. Udává se v bytech.

4.4 Funkce (functions.php)

Soubor *include/functions.php* obsahuje funkce, které jsou často použity v programu. Jedná se o tyto funkce:

- **time strtotimefix(\$strtotime:string)**: opravuje chybu v PHP5, kdy se špatně přičítal čas k datumu.
- **integer get_messages_count(\$folder:string)**: vrací celkový počet zpráv v dané složce aktuálně přihlášeného uživatele.
- **integer get_new_messages_count(\$folder:string)** : vrací počet nových zpráv v dané složce aktuálně přihlášeného uživatele.
- **void get_folders(void)**: vytvoří názvy složek aktuálně přihlášeného uživatele.
- **get_messages(\$id_user:integer,\$id_mailbox:integer,\$od:integer,\$kolik:integer, \$order:string,\$direction:string)**
- **void create_menu(void)**: Vytvoří menu.
- **array parse_address(\$text:string)**: Analyzuje adresu emailu a vrátí pole s jednotlivými položkami:

- o `foo["name"]` – skutečné jméno uživatele.
 - o `foo["user"]` – uživatelské jméno.
 - o `foo["domain"]` – doména, ze které byl email odeslán.
- **string `dej_mime($jmeno_souboru:string)`**: Ze zadaného jména souboru vyseparuje příponu a vrátí její mime typ.
- **boolean `check_mail($email:string)`**: Provede kontrolu mailu. Vrací hodnotu `true` pokud mail odpovídá RFC822.

4.5 Soubory s jazyky (`czech.php`, `english.php`, `language.php`)

V těchto souborech se nachází překlady veškerých textů, které se v programu vyskytují. Program umožňuje přidávání dalších jazyků. Musíme pouze v souboru `include/language.php` přidat následující řádky do větvení `switch($_SESSION["ses_lang"])`:

```
1 case 'XX':  
2     include("lang/YYYY.inc");  
3 break;
```

kde `XX` je zkratka jazyka, která se předává v session (např.: `cz`, `en`) a `YYYY` je samotný soubor s jazykem, který má následující formát:

`$foo = 'Text';`

kde `$foo` je symbolický název proměnné. Např.: v souboru `lang/czech.inc` bude mít řádek s hláškou o úspěšném odhlášení následující tvar:

`$LogoutOK = 'Odhlášení proběhlo úspěšně';`

Ten samý řádek bude v souboru `lang/english.inc` vypadat následovně:

`$LogoutOK = 'Logout has been succesful';`

Ještě bude třeba umístit nad přihlašovací tabulku (`login.php`) vlaječku s příslušným jazykem. Obrázek s vlajkou by se měl vyskytovat v adresáři `images`.

4.6 Přihlášení (`login.php`)

Ověření uživatele se provádí dvěma kroky. Jednak ověření uživatele v databázi a jednak ověření existence účtu na školním serveru pomocí protokolu FTP. Ověření pomocí

protokolu FTP se provádí připojením na FTP server zadaným v proměnné **\$FTPServer** v *include/config.inc* a následným pokusem o přihlášení uživatele. Heslo budeme brát z tabulky *user* ze sloupce *passwd*. Toto heslo se posílá v nezašifrované podobě. Pokud je odpověď od serveru kladná, můžeme považovat uživatele za ověřeného.

Přihlašovací formulář odesílá data pomocí metody POST (heslo se posílá zašifrované v JavaScriptu pomocí metody MD5, která je uložena v souboru *javascript/md5.js*) funkci **AuthenticateUser(\$user,\$password,\$context)**, která se nachází v souboru *include/auth.php*. Ta provede srovnání s databází a pokud je přihlašovací jméno a heslo pořádku vrátí hodnotu 1. V každém jiném případě vrátí hodnotu 0. Pokud bylo přihlášení úspěšné nastaví se proměnné:

- **\$_SESSION['ses_user']**: jméno uživatele.
- **\$_SESSION['ses_passw']**: heslo zašifrované pomocí metody MD5.
- **\$_SESSION['ses_first_login']**: po přihlášení je 1, kvůli updatu IP adresy a času přihlášení.
- **\$_SESSION['ses_context']**: kontext uživatele.
- **\$_SESSION['ses_mail_store']**: pokud chceme nechávat emaily na POP3 serveru nastavíme hodnotu 1.

a provede se přesměrování na *list.php*. Pokud se přihlášení nezdaří provede se přesměrování na *login.php*.

4.7 Seznam emailů (list.php)

Po přihlášení se zjistí id uživatele a styl se kterým byl naposledy přihlášen. Pokud se uživatel připojuje poprvé nastaví se defaultní styl. Dále se zjistí hodnota v **\$_SESSION['ses_first_login']**. Pokud je nastavena hodnota 1, zjistí se z databáze IP a datum posledního přihlášení a zároveň se provede update nových údajů. Následuje zjištění nových emailů na POP3 serveru viz 4.7.1 a jejich výpis na obrazovku. Nakonec ještě překontrolujeme přidělenou kvótu. Pokud byla překročena varujeme uživatele příslušným hlášením. Jestliže neuposlechne varování smažeme mu nějaké předem nadefinované emaily.

Zprávy se rozlišují barvami podle statusu přečtený/nepřečtený. Emaily jsou uspořádány v tabulce. V její hlavičce jsou odkazy, které umožňují třídění podle toho, na který odkaz klikneme (např. třídění podle velikosti zprávy, předmětu, atd.). Po každém lichém kliknutí na stejný odkaz jsou mailly seříděny sestupně a při každém sudém kliknutí vzestupně.

Ke stránkování potřebujeme znát `$_GET['od']` a `$_GET['kolik']`. Poté se provede výpočet počtu stránek. Jestliže hodnoty těchto proměnných neznáme nastavíme `$_GET['od'] = 0` a `$_GET['kolik']` dostaneme ze souboru *include/config.php* z proměnné `$EmailsPerPage`.

Při každém obnovení (znovunačtení) stránky zjišťujeme nové emaily na poštovním serveru viz 4.7.1. To se provede buď tlačítkem obnovit nebo kliknutím na některou ze složek. Složky jsou vytvořeny pomocí funkce `get_folders()` v souboru *include/functions.php*.

4.7.1 Zjištění nových emailů

Nejprve je potřeba identifikátor, který by byl jedinečný. Použijeme Message-ID. Pokud jsou na POP3 serveru nějaké emaily, provedeme porovnání Message-ID v každém emailu s emaily uloženým v databázi. Jestliže je v databázi dané Message-ID s emaillem dále nepracujeme. Porovnávání provádíme kvůli možné duplicitě emailů. Ta vzniká, když uživatelé zatrhne při přihlašování „Ponechat kopie na serveru“. V případě že Message-ID nebylo v databázi nalezeno, provedeme další zpracování emailu. Zjistíme odesílatele a příjemce a údaje rozebereme podle RFC822 na jednotlivé položky. Dále zjistíme předmět emailu, znakovou sadu, ve které byl email napsán, zda obsahuje nějaké přílohy a jejich kódování, datum, kdy byl email napsán a velikost emailu v bytech. Všechny tyto informace uložíme do databáze do tabulky *messages*. Nakonec se ještě vyhodnotí proměnná `$_SESSION['ses_mail_store']`. Za podmínky že je hodnota této proměnné 0, smažeme email z POP3 serveru. V opačném případě jej tam ponecháme.

4.8 Psaní emailů (compose.php)

Nejprve vytvoříme formulář s dvěma textboxy („Příjemce“, „Předmět“), jedním fileboxem („Příloha“), jednou textareou (pro psaní textu) a třemi tlačítky („Vymazat“,

„Odložit“, „Odeslat“). Jestliže máme nějakou hodnotu v `$_GET['msg_id']`, jedná se o reply a dané položky vyplníme emailem, na který reagujeme.

Data jsou odesílána metodou POST. Nyní je třeba vytvořit hlavičku emailu a složit celé tělo dohromady. To obstarává třída **Mail_mime()**. Text přidáme do emailu pomocí metody **boolean setTxtBody (string \$data)**, přílohy přidáváme metodou **boolean addAttachment (string \$data, string [\$c_type = 'application/octet-stream'], string [\$name = ''], boolean [\$isfile = TRUE], string [\$encoding = 'base64'])** a celé to dáme dohromady metodou **string &get (array [\$param = NULL])**. Poté je ještě třeba přidat námi nadefinované hlavičky metodou **array &headers ([array \$headerEx = NULL])**.

Po vytvoření emailu jej odešleme ke zpracování SMTP serveru. O to se stará třída **Mail()** a její metody **boolean send (mixed \$recipients, array \$headers, string \$body)** a **object &factory (string \$backend [, array \$param = array()])**, kterým předáme příslušné parametry. Jestliže nám SMTP server odpoví +OK, je email úspěšně zpracován a uložíme jej do databáze do tabulky *messages* s hodnotou `id_mailbox = 3` (složka odeslané).

4.9 Prohlížení emailu (view.php)

Zobrazení odesílatele, příjemce, předmětu a data odeslání se provádí jednoduchým dotazem do databáze. Poté je zobrazena nabídka přesunutí a smazání. Přesun je realizován updatem sloupce *id_mailbox* v tabulce *messages* nově zadanou hodnotou. Smazání se provádí nejprve přesunutím, tedy updatem, emailu do složky koš a poté případné fyzické vymazání z databáze.

Dále jsou zobrazeny přílohy, pokud nějaké existují. To je provedeno za pomoci třídy **Mail_mimeDecode()** a její metody **object decode (array [\$args = NULL])**. Zobrazuje se název přílohy. Po kliknutí se provede v případě textu, html souboru nebo souboru s obrázkem jeho zobrazení v novém okně. Jestliže se jedná o jiný typ souboru provede se vyzvání uživatele k jeho uložení do počítače. Toto vše je realizováno souborem *download.php*.

Nakonec je zobrazen případný text emailu.

4.10 Nastavení složek (folder.php)

Zobrazování a skrývání textového pole a vyskakovacích menu je řešeno pomocí javascriptového souboru, který lze nalézt v *javascript/folders.js*.

Vytvoření nové složky je řešeno vytvořením záznamu v tabulce *mailboxes* pro právě přihlášeného uživatele.

Přejmenování se provádí UPDATEm sloupce *nazev* v tabulce *mailboxes*.

Smazání složky je provedeno fyzickým smazáním požadovaného záznamu aktuálně přihlášeného uživatele.

4.11 Natavení stylů (styl.php)

Nejdříve se provede vypsání všech stylů z číselníku *cis_styl* seřazeného podle sloupce *nazev*. Po vybrání stylu se provede update tabulky *user* a hodnoty *id_styl* přihlášeného uživatele.

4.12 Nastavení programu (settings.php)

Z tabulky *user* se vyberou sloupce *delete_trash* a *delete_sent* a podle hodnoty („A“ nebo „N“) se zobrazí zatržítka. Pak se ještě zobrazí dvě textová pole s typem password.

Po odeslání formuláře je nejdříve provedeno porovnání hesel a pokud se shodují uloží se nová hodnota. Taktéž se uloží hodnoty pro sloupce *delete_trash* a *delete_sent*.

4.13 Odhlášení (logout.php)

Při odhlášení jsou zničeny všechny proměnné v session. Pokud bychom nyní stiskli tlačítko zpět již se do aplikace nedostaneme a musíme provést nové přihlášení. Je provedeno přesměrování na *login.php*.

4.14 Přejít na další rok

Když student ukončí studium v 5. ročníku většinou odchází ze školy, ale emaily zůstávají v databázi. Proto je vhodné je smazat. Toto je řešeno pomocí PHP skriptu uloženého v souboru *tools/prechod.php*.

Nejprve se nastaví uživatelé, kteří budou rozvolňovat. Potom jsou vybráni všichni uživatelé, kteří jsou v pátém ročníku a nerozvolňují. Poté jsou vymazány všechny jejich emaily a následně jsou smazáni i tito uživatelé. Nakonec je všem ostatním uživatelům, kteří nerozvolňují, zvednut ročník o jedničku.

5 UŽIVATELSKÁ SEKCE

5.1 Přihlášení uživatele

Uživatel:

Kontext:

Heslo:

☐ Ponechat kopie na serveru

Obr. 2. Úvodní okno

Uživatel má možnost zvolit si ze dvou jazyků a to češtiny a angličtiny. Proveďte to kliknutím na příslušnou vlajku.

Aby se mohl přihlásit, musí vyplnit uživatelské jméno a heslo. Také je potřeba vybrat správný kontext. Pokud zaškrtně položku „Ponechat kopie na serveru“ emaily se nebudou mazat z poštovního serveru. Jinak budou převedeny do databáze a smazány.

5.2 Zobrazení seznamu zpráv

Vítejte Test Testovic Datum posledního přihlášení: 2005-05-08 12:32:53 IP posledního přihlášení: 127.0.0.1																																																																																			
Čist poшту Napsat zprávu Nastavení složek Nastavení stylů Odhlásit																																																																																			
Doručené (4/12) Kuš (0/0) Odeslané (13/14) Spam (0/0) Rozepsané (0/0) abc (0/0) Skolicka (0/0) Nová složka: Vytvořit	<table> <tr> <th>☐</th><th>Odesílatel</th><th>Předmět</th><th>Datum</th><th>Velikost</th><th>Akce</th></tr> <tr> <td>☐</td><td>Test Testovic</td><td>2.ukol</td><td>08.05.2005 00:18</td><td>521</td><td>X P</td></tr> <tr> <td>☐</td><td>P Test Testovic</td><td>2.ukol</td><td>08.05.2005 00:06</td><td>121867</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>Fwd:</td><td>08.05.2005 00:05</td><td>2355</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>RE:Api</td><td>08.05.2005 00:02</td><td>999</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>Api</td><td>07.05.2005 23:59</td><td>972</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>mail</td><td>07.05.2005 20:14</td><td>534</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>Fwd: Fwd: pulci</td><td>05.05.2005 12:58</td><td>506</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>Druhýpříklad</td><td>05.05.2005 12:55</td><td>545</td><td>X P</td></tr> <tr> <td>☐</td><td>P Test Testovic</td><td>Druhýpříklad</td><td>05.05.2005 11:01</td><td>200082</td><td>X P</td></tr> <tr> <td>☐</td><td>Test Testovic</td><td>ahoi</td><td>05.05.2005 10:52</td><td>499</td><td>X P</td></tr> <tr> <td>☐</td><td>P Test Testovic</td><td>Fwd: Fwd: potěšte si očko</td><td>03.05.2005 22:41</td><td>200094</td><td>X P</td></tr> <tr> <td>☐</td><td>=?ISO-8859-1?Q?Tom=E1=9A?= Bubela</td><td>Druhýpříklad</td><td>03.05.2005 11:19</td><td>594</td><td>X P</td></tr> </table>					☐	Odesílatel	Předmět	Datum	Velikost	Akce	☐	Test Testovic	2.ukol	08.05.2005 00:18	521	X P	☐	P Test Testovic	2.ukol	08.05.2005 00:06	121867	X P	☐	Test Testovic	Fwd:	08.05.2005 00:05	2355	X P	☐	Test Testovic	RE:Api	08.05.2005 00:02	999	X P	☐	Test Testovic	Api	07.05.2005 23:59	972	X P	☐	Test Testovic	mail	07.05.2005 20:14	534	X P	☐	Test Testovic	Fwd: Fwd: pulci	05.05.2005 12:58	506	X P	☐	Test Testovic	Druhýpříklad	05.05.2005 12:55	545	X P	☐	P Test Testovic	Druhýpříklad	05.05.2005 11:01	200082	X P	☐	Test Testovic	ahoi	05.05.2005 10:52	499	X P	☐	P Test Testovic	Fwd: Fwd: potěšte si očko	03.05.2005 22:41	200094	X P	☐	=?ISO-8859-1?Q?Tom=E1=9A?= Bubela	Druhýpříklad	03.05.2005 11:19	594	X P
☐	Odesílatel	Předmět	Datum	Velikost	Akce																																																																														
☐	Test Testovic	2.ukol	08.05.2005 00:18	521	X P																																																																														
☐	P Test Testovic	2.ukol	08.05.2005 00:06	121867	X P																																																																														
☐	Test Testovic	Fwd:	08.05.2005 00:05	2355	X P																																																																														
☐	Test Testovic	RE:Api	08.05.2005 00:02	999	X P																																																																														
☐	Test Testovic	Api	07.05.2005 23:59	972	X P																																																																														
☐	Test Testovic	mail	07.05.2005 20:14	534	X P																																																																														
☐	Test Testovic	Fwd: Fwd: pulci	05.05.2005 12:58	506	X P																																																																														
☐	Test Testovic	Druhýpříklad	05.05.2005 12:55	545	X P																																																																														
☐	P Test Testovic	Druhýpříklad	05.05.2005 11:01	200082	X P																																																																														
☐	Test Testovic	ahoi	05.05.2005 10:52	499	X P																																																																														
☐	P Test Testovic	Fwd: Fwd: potěšte si očko	03.05.2005 22:41	200094	X P																																																																														
☐	=?ISO-8859-1?Q?Tom=E1=9A?= Bubela	Druhýpříklad	03.05.2005 11:19	594	X P																																																																														
1																																																																																			

Obr. 3. Seznam doručených zpráv

Po přihlášení se uvidíte obrazovku podobnou jako na Obr. 3.

Horní panel ukazuje přihlášeného uživatele, datum posledního přihlášení a adresu IP, ze které bylo provedeno poslední úspěšné přihlášení.

Levý panel zobrazuje složky. Standardně jsou do složek zahrnuty:

- Doručené
- Koš
- Odeslané
- Spam
- Rozepsané

Uživatel může dále přidávat své vlastní složky pomocí textového pole „Nová složka“ a stiskem tlačítka „Vytvořit“.

Vedle panelu se složkami se nachází hlavní panel s emaily. Po kliknutí na text: odesílatel, předmět, datum nebo velikost se provede seřazení podle zadaného kritéria. Emaily se seřadí vzestupně, po opětovném kliknutí na tentýž text se emaily seřadí sestupně. Kliknutím na čtvereček nalevo od sloupce odesílatel se označí všechny viditelné zprávy. Ve sloupečku „Akce“ jsou dvě ikony, křížek a šipka. Kliknutím na křížek se email přesune do koše a po kliknutí na šipku se vytvoří nový email jako reply. Zprávu přečteme kliknutím na předmět. Listování v seznamu zpráv se provádí pomocí stisku tlačítek „Předchozí“ a „Další“, případně kliknutím na číslo příslušné stránky, které se nacházejí pod seznamem zpráv.

Nad panelem zpráv se nachází standardní panel s menu.

5.3 Čtení zpráv

Vítejte Test Testovic Datum posledního přihlášení: 2005-05-13 18:31:19 IP posledního přihlášení: 127.0.0.1	
Doručené (2/9) Koš (0/1) Odeslané (13/15) Spam (0/0) Rozepsané (0/0) abc (0/0) Školička (0/0)	Číst poštu Napsat zprávu Nastavení složek Nastavení stylů Odhlásit Odesílatel: Test Testovic <bbs@ft-st.localhost.cz> Příjemce: <bbs@localhost> Předmět: Testovací zpráva Datum: 11.05.2005 17:07 Přesunout do: Složky Přesunout Vymazat uo.reg; Einsteinovu srdci byla nejbližší jeho sjednocená teorie pole. Einstein jednou před Helen Dukasovou poznamenal, že tomu, co dělá, možná fyzikové porozumí za sto let. Mýlil se. K oživení zájmu o sjednocenou teorii pole došlo ani ne za padesát let. Sjednocení, jemuž se kdysi fyzikové vysmívali, protože prý beznadějně přesahuje naše možnosti, máme dnes možná na dosah ruky. Je hlavním bodem programu téměř každého setkání teoretických fyziků. Po dvou tisíciletích zkoumání vlastností hmoty, jež začalo okamžikem, kdy si Démokritos a jeho řečtí krajané položili otázku, z čeho je vesmír, zplodila fyzika dvě konkurenční teorie, které jsou zcela neslučitelné. První z nich je kvantová teorie, která jedinečně popisuje svět atomů a subatomárních částic. Druhou je Einsteinova obecná relativita, jež nám dala závratnou teorii černých děr a rozpínajícího se vesmíru. Největším paradoxem je to, že tyto dvě teorie jsou svými pravými opaky. Jsou založeny na odlišných předpokladech, odlišné matematice a odlišných fyzikálních představách. Kvantová teorie spočívá na oddělených balíčcích energie nazývaných „kvanta“ a na reji subatomárních částic. Teorie relativity je založena na hladkých plochách. Nejrozvinutější verze kvantové fyziky, kterou fyzikové zformulovali, je shrnuta do takzvaného „standardního modelu“ a dokáže vysvětlit experimentální data popisující subatomární částice. V jistém smyslu jde o nejúspěšnější teorii přírody, která dokáže popsat vlastnosti tří ze čtyř základních sil (sily elektromagnetické a dále slabé a silné jaderné interakce). Standardní model má přes svůj velký úspěch dva do očí bijící nedostatky. V první řadě je to model krajně ošklivý – jde asi o vůbec nejohrovnější teorii, jež byla kdy ve

Obr. 4. Čtení zprávy

Při čtení zprávy je hlavní panel rozdělen na tři, popř. čtyři části (podle toho jestli čtený email neobsahuje nebo obsahuje přílohu).

V první části můžeme nalézt základní informace o emailu jako jsou odesílatel, příjemce, předmět a datum.

Ve druhé části nám program umožňuje přesunutí nebo smazání emailu. Email přesuneme tak, že vybereme složku kam chceme email umístit a stiskneme „Přesunout“. Vymazání emailu je možno provést stiskem tlačítka „Vymazat“.

Ve třetí části je seznam s přílohami. Kliknutím na přílohu začne stahování. Pokud se jedná o obrázek, text nebo html soubor, zobrazí se v novém okně.

Ve čtvrté části je text emailu.

5.4 Psaní zpráv

Vítejte Test Testovic Datum posledního přihlášení: 2005-05-08 12:32:53 IP posledního přihlášení: 127.0.0.1	
Doručené (4/13) Kuš (0/0) Odeslané (16/17) Spam (0/0) Rozepsané (0/0) abc (0/0) Školička (0/0)	Číst poštu Napsat zprávu Nastavení složek Nastavení stylů Odhlásit Odesílatel: Test Testovic <bbs@ft-st.localhost.cz> Příjemce: bbs@localhost Předmět: Testovací zpráva Příloha: C:\uo.reg Procházet... <pre> DAMAGE=0 117.428574 135.428574 DELAY=2000 2000 SIZE=0.500000 NPCFLAG=021 FACTION=35 [OBJECT] GUID=0150CE3 TYPE=3 FLAGS=04 ENTRY=6491 MODEL=5233 LEVEL=60 XYZ=8923.063477 -169.451157 80.548767 3.466776 MAXHEALTH=100 MAXPOWERS=260 0 0 0 DAMAGE=0 117.428574 135.428574 DELAY=2000 2000 SIZE=0.300000 NPCFLAG=021 FACTION=35 </pre> Vymazat Uložit Odeslat Zpráva nebyla odeslána Špatně zadán email

Obr. 5. Psaní zpráv

Pro odeslání emailu musíme vyplnit položky „Příjemce“ adresou, na kterou má být email doručen, „Předmět“ a případně vyplníme položku „Příloha“ tak, že klikneme na tlačítko „Procházet“. Dále vyplníme textové pole textovou zprávou.

Zprávu odešleme tlačítkem „Odeslat“ jen tehdy, pokud je email příjemce správně vyplněn. Provádí se kontrola platnosti emailu podle RFC822. Zprávu lze též uložit k pozdější editaci pomocí tlačítka „Uložit“. Vymazání celého formuláře provedeme stiskem tlačítka „Vymazat“.

Po odeslání emailu SMTP serveru nás program informuje zda byl či nebyl email skutečně odeslán.

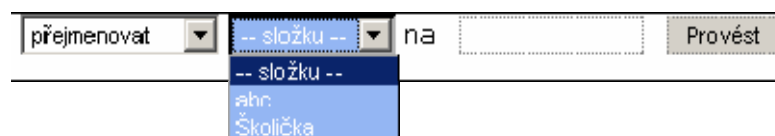
5.5 Nastavení složek

Vítejte Test Testovic Datum posledního přihlášení: 2005-05-08 12:32:53 IP posledního přihlášení: 127.0.0.1	
Doručené (4/13) Kuš (0/0) Odeslané (16/17) Spam (0/0) Rozepsané (0/0) abc (0/0) Školička (0/0)	Číst poštu Napsat zprávu Nastavení složek Nastavení stylů Odhlásit přejmenovat -- složku -- na Provést

Obr. 6. Celkový pohled na nastavení složek

Nastavení složek umožňuje přejmenování, mazání a vytváření nových složek. Vše se ovládá pomocí prvního vysouvacího menu.

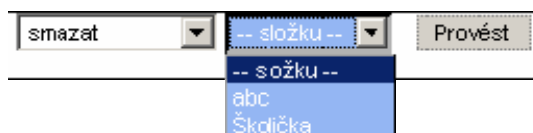
5.5.1 Přejmenování složky



Obr. 7. Přejmenování složky

V prvním vysouvacím menu si zvolíme položku „přejmenovat“. Ve druhém vysouvacím menu vybereme složku, kterou chceme přejmenovat, do textového pole napíšeme nový název a stiskneme tlačítko „Provést“.

5.5.2 Smazání složky



Obr. 8. Smazání složky

V prvním vysouvacím menu si zvolíme položku „smazat“. Ve druhém vysouvacím menu vybereme složku, kterou chceme smazat a stiskneme tlačítko „Provést“.

5.5.3 Vytváření složky



Obr. 9. Vytváření složky

V prvním vysouvacím menu si zvolíme položku „vytvořit složku“. Do textového pole napíšeme nový název složky a stiskneme tlačítko „Provést“.

5.6 Nastavení stylů

Vítejte Test Testovic Datum posledního přihlášení: 2005-05-08 12:32:53 IP posledního přihlášení: 127.0.0.1	
Doručené (4/13) Koš (0/0) Odeslané (16/17) Spam (0/0) Rozepsané (0/0) Jabc (0/0) Školicka (0/0)	Čist poštou Napsat zprávu Nastavení složek Nastavení stylů Odhlásit ☐ Grafický ☐ PDA ☐ Textový ☐ Výrazný Provést změnu

Obr. 10. Nastavení stylů

Nastavení stylů je velice jednoduché. Stačí kliknout na vybraný styl a stisknout tlačítko „Provést změnu“.

5.7 Nastavení programu

Vítejte Michal Kroma Datum posledního přihlášení: 2005-05-22 20:51:07 IP posledního přihlášení: 127.0.0.1	
Doručené (0/1) Koš (0/0) Odeslané (0/1) Spam (0/0) Rozepsané (0/1) Jaho (0/0)	Čist poštou Napsat zprávu Nastavení složek Nastavení stylů Nastavení Odhlásit Smazat emaily z koše: ☒ Smazat emaily ze složky odeslané: ☒ Smazat emaily ze složky rozepsané: ☒ Nové heslo: Ověření nového hesla: Provést

Obr. 11. Nastavení programu

V kategorii „Nastavení programu“ lze provést následující věci:

- **Smazání emailů z koše:** při ukončení práce s programem smaže všechny zprávy ze složky „Koš“.
- **Smazání emailů ze složky „Odeslané“:** při ukončení práce s programem smaže všechny zprávy ze složky „Odeslané“.
- **Smazání emailů ze složky „Rozepsané“:** při ukončení práce s programem smaže všechny zprávy ze složky „Rozepsané“.

- **Nastavení nového hesla:** umožňuje uživateli změnit své heslo. Zadávají se dvě hesla, které jsou stejné (kvůli možné chybě při případném překlepnutí).

5.8 Odhlášení uživatele

Při odhlášení uživatele se vymažou emaily z koše a ze složky „Odeslané“ (pokud je to uživatelem povoleno) a zároveň se zruší všechny informace o přihlášeném uživateli.

ZÁVĚR

Pokusil jsem vytvořit program, se kterým by mohl pracovat i běžný uživatel. Myslím si, že aplikace obsahuje základní funkce pro práci s emaily jako je přijímání a odesílání, možnost přesunu emailů mezi složkami, práci se složkami (vytváření nových složek, přejmenovávání nebo třeba mazání starých nepotřebných složek) a změna stylu emailu (dle uživatelových požadavků).

Největším problémem pro mě bylo asi provedení změny stylu textového v grafický a naopak. Nakonec jsem to vyřešil pomocí zdvojených tagů `<span>` a pomocí `display:none`. Dalším problémem byla pomalost POP3 serveru. Při každém reloadu stránky se totiž stahoval celý seznam zpráv což při 20 emailech trvalo asi 15 sekund. Toto jsem vyřešil přesunem všech emailů do databáze. Tam trvá stažení seznamu zpráv o 20 emailech asi 2 sekundy, což je mnohem rychlejší než přímý dotaz na POP3 server. Nakonec jsem řešil problém s autorizací uživatele. Toto je prozatím, ale funkčně, vyřešeno pomocí FTP protokolu. Je to celkem dobré řešení, pokud je ovšem FTP server online. Nemusí tomu tak vždy být, proto bych rád dodělal přihlašování pomocí LDAP (Lightweight Directory Access Protocol).

V budoucnosti bych chtěl dodělat spoustu dalších věcí, které ovšem překračují rámec zadání bakalářské práce. Jednou z hlavních by bylo fungování aplikace na SSL (Secure Socket Layer) pro zvýšení zabezpečení. Přidat osobní adresář pro každého uživatele, možnost posílat více příloh najednou, schopnost poslat najednou jeden email více adresátům, pokusit se implementovat přihlašování pomocí LDAP, přidat spamový filtr a pravděpodobně i HTML editor řešený pomocí DHTML.

SEZNAM POUŽITÝCH ZDROJŮ

1. BRÁZDA, J.: *PHP 4: učebnice základů jazyka*. GRADA, 2002. ISBN 80-247-0442-0.
2. CASTAGNETTO, J.: *PHP Programujeme profesionálně*. Computer Press, 2001. ISBN 80-7226-310-2.
3. KOSEK, J.: *PHP: Tvorba interaktivních internetových aplikací*. GRADA, 1999. ISBN 80-7169-501-7.
4. MASLAKOWSKI, M.: *Naučte se MySQL za 21 dní*. Computer Press, 2001. ISBN 80-7226-448-6.
5. PROKOP, M.: *CSS: Kaskádové styly pro webdesignéry*. Mobil Média, 2003. ISBN 80-251-0487-7
6. *Odkazy na zdroje o CSS*. [cit. 2005-04-18]. Dostupný z URL: <http://www.jakpsatweb.cz/katalog/css.html>.
7. *Three Columns - Flanking Menus*. [cit. 2005-04-18]. Dostupný z URL: <http://bluerobot.com/web/layouts/layout3.html>.
8. *Blog o tvorbě webu*. [cit. 2005-04-18]. Dostupný z URL: <http://www.maciweb.net/blog/web/ukazky.php>.
9. *Sending mail: mail() - Practical PHP Programming*. [cit. 2005-04-18]. Dostupný z URL: http://www.hudzilla.org/phpbook/read.php/15_5_0.
10. *Using PEAR's mimeDecode module : evolt.org, Backend*. [cit. 2005-04-18]. Dostupný z URL: http://www.evolt.org/article/Using_PEAR_s_mimeDecode_module/18/60103.
11. *V PHP o PHP - Co je to PHP?*. [cit. 2005-04-18]. Dostupný z URL: <http://www.php-o-php.jicinet.cz/cojephphp.php>.
12. *TomasParal.cz | Non ASCII znaky v mailových hlavičkách pomocí PHP*. [cit. 2005-04-18]. Dostupný z URL: <http://www.tomasparal.cz/programator/php/headers/non-ascii-znaky-v-mailovych-hlavickach.html>.

13. *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text*. [cit. 2005-04-18]. Dostupný z URL:
<<http://www.ietf.org/rfc/rfc2047.txt>>.
14. *IANA Charset*. [cit. 2005-04-18]. Dostupný z URL:
<<http://www.ietf.org/rfc/rfc2278.txt>>.
15. *UTF-8, a transformation format of ISO 10646*. [cit. 2005-04-18]. Dostupný z URL: <<http://www.ietf.org/rfc/rfc3629.txt>>.
16. *PHP - 25. díl – úvod do databází*. [cit. 2005-04-30]. Dostupný z URL:
<<http://www.zive.cz/h/Programovani/AR.asp?ARI=117655&CAI=2038>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheets
DBMS	DataBase Management System
DHTML	Dynamic HTML
FTP	Význam druhé zkratky.
HTML	HyperText Markup Language
LDAP	Lightweight Directory Access Protocol.
PHP	PHP: Hypertext Preprocessor
POP3	Význam třetí zkratky.
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SŘBD	Systém Řízení Báze Dat
SSL	Secure Socket Layer

SEZNAM OBRÁZKŮ

Obr. 1. Struktura databáze	22
Obr. 2. Úvodní okno	32
Obr. 3. Seznam doručených zpráv	32
Obr. 4. Čtení zprávy.....	34
Obr. 5. Psaní zpráv.....	35
Obr. 6. Celkový pohled na nastavení složek.....	35
Obr. 7. Přejmenování složky.....	36
Obr. 8. Smazání složky	36
Obr. 9. Vytváření složky.....	36
Obr. 10. Nastavení stylů	37
Obr. 11. Nastavení programu.....	37

SEZNAM PŘÍLOH

P 1 : Disk CD s aplikací a bakalářskou prací