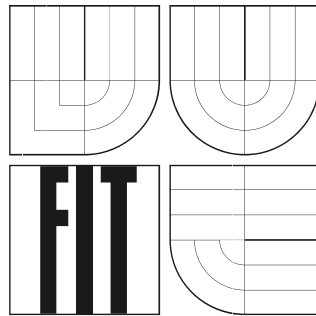


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



# **Webová platforma pro výměnu dat**

Semestrální projekt

# Webová platforma pro výměnu dat

© Martin Herodek, 2007.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

## Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Radka Burgeta, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Herodek

3. 1. 2007

## **Abstrakt**

Tento semestrální projekt zahrnuje analýzu stávajících možností výměny dat, nástrojů pro tvorbu internetových aplikací a návrh aplikace pro snadnou výměnu souborů mezi uživateli přes internet. Důležitým požadavkem je co nejjednodušší práce na straně klienta.

## **Klíčová slova**

PHP, MySQL, JavaScript, server, klient, e-mail, instant messaging, FTP, P2P, P2M, výměnné servery

## **Poděkování**

Děkuji vedoucímu své práce Ing. Radku Burgetovi, Ph.D. za poskytnuté rady a odbornou pomoc

## **Abstract**

This semestral project includes analysis of current possibilities of data exchange, web application development tools and design of a system, which will be able to intermediate as easy as possible data exchange via internet. As easy as possible usage on the client side is one of the most important requirements

## **Keywords**

PHP, MySQL, JavaScript, server, client, e-mail, instant messaging, FTP, P2P, P2M, exchange servers

# Obsah

Obsah .....	5
1 Úvod.....	7
1.1 Dílčí úkoly.....	7
1.2 Obsah jednotlivých kapitol.....	7
2 Teoretický rozbor.....	8
2.1 Sdílení dat.....	8
2.2 Skriptovací jazyky.....	8
2.2.1 PHP .....	8
2.2.2 JavaScript.....	10
2.3 MySQL.....	10
2.4 Srovnání fungování HTML, PHP a MySQL.....	11
3 Analýza .....	12
3.1 Úvod.....	12
3.2 E-mail .....	12
3.3 Instant messaging .....	12
3.4 File Transfer Protocol (FTP).....	13
3.5 Peer To Peer (P2P).....	13
3.6 Peer To Mail (P2M).....	14
3.7 Výměnné servery.....	14
3.8 Další služby .....	14
4 Specifikace požadavků.....	15
4.1 Cílové zaměření aplikace .....	15
4.2 Specifikace požadavků na systém .....	15
4.2.1 Uživatelé .....	15
4.2.2 Administrátoři .....	15
4.2.3 Skupiny .....	16
4.2.4 Zabezpečení .....	16
4.2.5 Komunikace .....	16
4.2.6 Uživatelské rozhraní .....	16
4.3 Specifikace požadavků na server .....	17
4.4 Specifikace požadavků na klienta .....	17
4.5 Omezení aplikace .....	17
5 Návrh řešení .....	18
5.1 Implementační rozhraní.....	18

5.2	Systém ukládání dat na serveru .....	18
5.3	Návrh implementace jednotlivých požadavků .....	19
5.3.1	Uživatelé .....	19
5.3.2	Administrátoři .....	19
5.3.3	Skupiny .....	19
5.3.4	Zabezpečení .....	19
5.3.5	Komunikace .....	20
5.3.6	Uživatelské rozhraní .....	20
6	Závěr .....	21
6.1	Dosažené výsledky .....	21
6.2	Zhodnocení průběhu práce .....	21
	Literatura .....	22

# 1 Úvod

Tento semestrální projekt se zabývá problematikou výměny dat přes internet. Je zde provedena analýza současných možností, jak sdílet určitá data mezi uživateli, kteří nemohou nebo nechtějí řešit jejich transfer v osobním kontaktu a jsou tak odkázáni na síť všech sítí. Na základě této analýzy se bude zakládat návrh optimálního řešení pro datovou výměnu pomocí internetu, jehož vlastní implementace proběhne v rámci diplomové práce.

Při návrhu (a pozdější implementaci) bude kladen důraz na minimální nároky na uživatele a to jak po stránce softwarové tak znalostní. Vzhledem k primárnímu určení aplikace, kterým je nasazení v malých skupinách uživatelů (podniky, studentské skupiny apod.), bude dalším důležitým kritériem minimalizace různých omezení, se kterými se lze setkat u současných použitelných služeb.

## 1.1 Dílčí úkoly

- Seznamte se se stávajícími možnostmi výměny velkých objemů dat mezi uživateli prostřednictvím sítě Internet
- Prostudujte dostupné technologie pro tvorbu webových aplikací
- Navrhněte webovou aplikaci umožňující snadnou výměnu velkých objemů dat mezi uživateli nebo skupinami uživatelů. Zaměřte se na jednoduchost použití i při výměně velkého množství souborů.

## 1.2 Obsah jednotlivých kapitol

Následující kapitola (Teoretický rozbor) představuje uvedení do problematiky sdílení dat a krátce popisuje technologie, které budou při tvorbě aplikace využity. Vlastním technologiím, které lze pro výměnu dat použít, je věnována kapitola třetí (Analýza), která popisuje jednotlivé služby včetně jejich primárního určení, pozitiv a negativ. Z analýzy vychází specifikace požadavků na aplikaci (čtvrtá kapitola), na kterou navazuje pátá kapitola. V ní je uveden návrh řešení systému, stejně jako specifikace doporučeného cílového segmentu uživatelů.

## 2 Teoretický rozbor

### 2.1 Sdílení dat

Internet patří mezi nejpoužívanější zdroje informací. Jeho nejvýznamnější výhodou je dostupnost téměř jakýchkoliv dat z jakéhokoliv zeměpisného místa (předpokladem je samozřejmě aktivní připojení). Uživatel tak může přistupovat vzdáleně k datům, jejichž transport by za jiných okolností nebyl možný. Spolu se zvyšováním osob připojených k internetu, ale hlavně zvyšováním rychlosti připojení se přímo nabízí možnost sdílet (transferovat) data mezi uživateli. Pohodlí a rychlost odeslání souboru pracovnímu partnerovi e-mailem ve srovnání s přenosem na flash disku nemá konkurenci, se vzrůstající vzdáleností rapidně narůstá a od určitých distancí je to již jediné možné řešení.

V současné době lze provádět výměnu elektronických dat řadou způsobů. Jelikož byl každý z nich primárně určen pro specifické využití, ne vždy lze v dané situaci využít kterýkoliv z nich. Volba správného nástroje (služby) většinou závisí na velikosti přenášených dat, požadované rychlosti (a čase), nutnosti současného připojení odesílatele i příjemce a potřebě specifického hardwarového (veřejná IP adresa), softwarového (specifický klient určité služby) či jiného (registrace služby, např. ICQ) prostředku. Popis principu fungování jednotlivých služeb a nástrojů, které lze použít pro datový transfer, spolu s jejich výhodami a nevýhodami jsou podrobně rozepsány v kapitole Analýza.

### 2.2 Skriptovací jazyky

#### 2.2.1 PHP

##### 2.2.1.1 Historie PHP

Jazyk PHP (Personal Home Page) vytvořil v roce 1994 Rasmus Lerdorf. Ten při tvorbě své webové prezentace narazil na problém sledování návštěvnosti, který ho přiměl k vytvoření jednoduchého monitorovacího systému. Lerdorf jej napsal v jazyce Perl, který ale citelně zatěžoval server, proto následoval přepis kódu z Perlu do jazyka C. Tato verze byla poskytnuta i dalším uživatelům, kteří systém přijali s nadšením a postupem času začali přicházet s návrhy možného rozšíření. Rozšířená verze původního systému byla uvolněna pod názvem Personal Home Page Tools, později jako Personal Home Page Construction Kit.

Další rozšíření přineslo podporu příkazů SQL, zobrazování výsledků SQL dotazů a práci s formuláři. Jako PHP/FI 2.0 se tato verze velmi rychle rozšířila a rozmnožila počet spokojených uživatelů. Pod zkratkou FI (Form Interpreter) se skrývá program pro zpřístupnění databází.



PHP 3.0 přineslo v roce 1998 hlavně podporu operačního systému Windows, platformy Macintosh, mnoha databázových systémů, cookies atd., protože až do této verze se jednalo o systém výhradně pro prostředí Linx (Unix). Po opuštění původního významu zkratky se systém dále označuje pouze jako hypertextový preprocesor PHP.

Verzi 4 přichází s novým jádrem Zend, které opět rozšiřuje systém o řadu nových vlastností bez ztráty zpětné kompatibility. Podle testů došlo až k desetinásobnému zrychlení v porovnání s verzí 3.0. Verze 5 zachovává trend vývoje PHP a přidává nově používané technologie jako objektově orientované programování či podporu XML.

### **2.2.1.2 Vlastnosti PHP**

PHP je skriptovacím jazykem prováděným přímo na serveru, kde běží jeho interpret, což je hlavní rozdíl oproti skriptovacím jazykům prováděným na straně klienta (např. JavaScript).

#### **Výhody serverového zpracování**

- Menší objem dat přenášený ke klientovi
- Minimalizace přenosu klient-server při komunikaci s jinými serverovými aplikacemi
- Nižší nároky na klienta, který není zatěžován výpočty
- Vyšší ochrana zdrojového kódu – díky zpracování na serveru k nim nelze ze strany klienta přistupovat
- Vzhledem k úplnému serverovému zpracování odpadá nutnost zahrnout do skriptovacích operací veškeré možné varianty klientského nastavení. Kód se píše na míru známému serverovému nastavení, ne na řadu možných variant klientské konfigurace.

#### **Nevýhody serverového zpracování**

- Vzhledem k provádění veškerých operací na straně serveru je nutné jeho dostatečné softwarové i hardwarové dimenzování
- Díky kompletní interpretaci zdrojového kódu na straně serveru je znemožněna jakákoliv interakce na dění na straně klienta

### **2.2.1.3 Využití PHP**

Nejčastější použití PHP nacházíme v kombinaci s MySQL při řešení webových aplikací, které čerpají z výhod serverového zpracování. Dalším nepopíratelným pozitivem je dynamická tvorba HTML, kde se využívá možnosti generování (vkládání) textu na určené místo. Změna loga na všech stránkách webové prezentace firmy je pak záležitostí několika okamžiků.

## 2.2.2 JavaScript

JavaScript je stejně jako PHP jazykem skriptovacím, na rozdíl od něj je však interpretován na straně klienta. Tato hlavní odlišnost jej přímo předurčuje pro vytváření součástí webových aplikací, které reagují na dění na straně klienta. Od jednoduchých hodin se tak přes různé interaktivní mapy může jednat až o komplexní hry (nebo jiné aplikace).

Nejen JavaScript, ale jakékoliv jazyky interpretované až na straně klienta s sebou přináší v porovnání s jazyky interpretovanými na straně serveru jednu značnou nevýhodu. Při přípravě aplikace stačí znát nastavení interpretu na serveru a podle toho části prováděné přímo na něm implementovat. Při přípravě kódu určeného pro interpretaci na straně klienta ale není jeho konfigurace známa, návrhář proto musí ošetřit co nejvíce různých možných nastavení, aby maximalizoval korektnost fungování aplikace jako celku. Nabízí se zde přímé srovnání s internetovými prohlížeči, jenž pracují na různých zobrazovacích jádrech a podpora standardů je u nich na různé úrovni (mnohdy se liší i mezi verzemi). Nelze opomenout ani fakt, že klient může mít JavaScript zcela zakázán. Pak je nutné tuto skutečnost detekovat a mít alternativu i pro tyto případy.

Většina internetových prezentací nevolí pouze jednu technologii (PHP versus JavaScript), ale používá obě. Každou z nich v oblasti, ve které vyniká. I když lze řadu operací implementovat oběma technologiemi, většinou je použití jedné z nich výhodnější (kvůli principu, na kterém fungují). Např. JavaScriptem lze řešit korektnost vkládaných údajů před jejich odesláním na server, kde proběhne pouze jejich zpracování (např. uložení do databáze). Pomocí PHP by bylo možné tuto operaci řešit také, ale kontrola každé položky by musela proběhnout na serveru. Za předpokladu vložení chybných dat by tak provoz klient-server citelně narostl.

## 2.3 MySQL

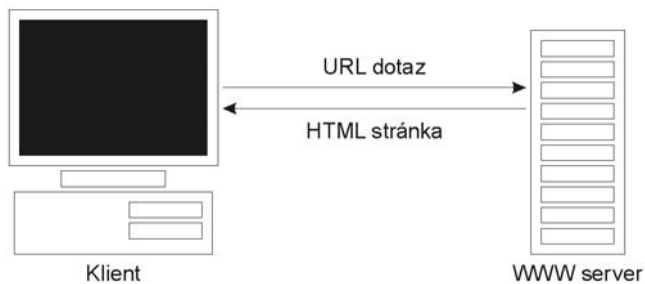
SQL (Structured Query Language nebo Simple Query Language) je dotazovacím databázovým jazykem. Jeho využití spočívá ve tvorbě tzv. dotazů, pomocí kterých lze pracovat s databází. SQL pracuje na řadě platform, jeho nevýhodou je řada různých verzí (tzv. dialektů), u kterých najdeme drobné nuance v tvorbě dotazů. Dotaz fungující v jednom dialektu je tak nutné pro použití v jiném většinou modifikovat.

Velmi často používaným dialektem v internetových aplikacích je MySQL, hlavně kvůli otevřenému zdrojovému kódu a konkurenceschopnosti s drahými protivníky jako např. Microsoft SQL Server nebo Oracle. Častá kombinace MySQL a PHP (doplněná JavaScriptem) je dána hlavně jejich bezproblémovou spoluprací a v neposlední řadě nulovými náklady, protože se ve všech případech jedná o freeware. O oblíbenosti svědčí i řada existujících projektů pro začínající vývojáře v podobě balíku vše v jednom (např. EasyPHP), které po instalaci vyžadují minimum (pokud vůbec) zásahů do konfigurace a uživatel může ihned po instalaci začít vyvíjet.

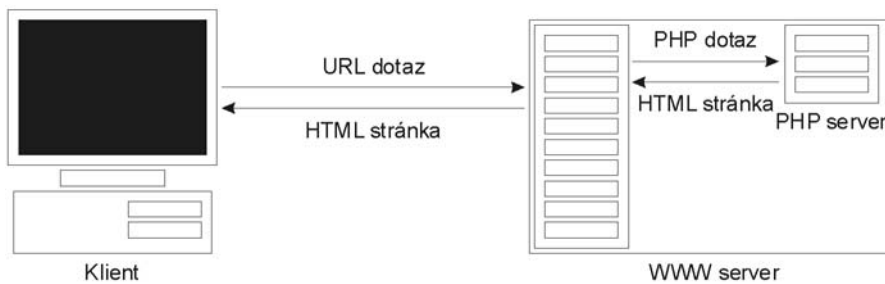
MySQL vyvíjí švédská společnost MySQL AB. Jedná se o systém určený pro správu relačních databází, které uchovávají data v podobě textu, čísel nebo binárních souborů. Jejich vzájemnou provázanost reprezentují tabulky. Práce s relačními databázemi přináší vyšší nároky na návrh a implementaci databáze, což je ale kompenzováno vyšší spolehlivostí a integritou uložených dat.

## 2.4 Srovnání fungování HTML, PHP a MySQL

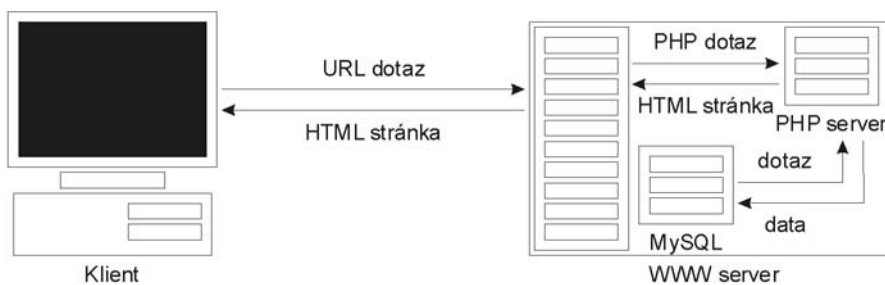
### HTML



### PHP



### MySQL



## 3 Analýza současných systémů

### 3.1 Úvod

Jak již bylo uvedeno výše, poskytuje internet celou řadu nástrojů a služeb, pomocí kterých lze transferovat data. Každá z nich je zaměřena na určitý způsob používání, z čehož plynou její pozitiva a negativa. V hodnocení se zaměříme na využitelnost dané služby pro různé veliké objemy dat, rychlost (relativní) a požadavky speciálního hardwarového, softwarového či jiného vybavení.

### 3.2 E-mail

Snad neexistuje člověk, který aktivně působí na internetu a nepoužívá e-mail. Pro mnohé uživatele je to primární cíl, kvůli kterému začnou internet používat. I když je e-mailová komunikace určena primárně pro text, lze ke zprávám připojovat i datové přílohy.

Zde narážíme na řadu problémů. I za předpokladu, že mají odesílatel i příjemce neomezenou (v porovnání s přílohou) velikost schránky, často je odeslání větší přílohy nemožné kvůli poskytovateli internetového připojení, jehož SMTP server má nastaven určitý limit. Pro zachování alespoň nějaké použitelnosti je tedy nutné, aby všichni účastníci měli e-mail s dostatečnou kapacitou (dnes řada gigabytových freemailů – *www.seznam.cz*, *www.centrum.cz*, *www.gmail.com*) a možnost použít SMTP server s co nejvyšším limitem velikosti odchozích zpráv, které se i v tom nejideálnějším případě pohybují v řádech maximálně desítek megabytů.

Výměna dat pomocí emailové komunikace je vhodná pouze pro malé objemy dat a vždy je nutné brát ohledy i na příjemce, neboť odesílatel ne vždy zná přesné parametry cílové schránky (velikost, aktuální zaplnění, atd.). Nepopíratelným pozitivem je snadnost použití – poštu lze číst přes webové rozhraní (není nutná instalace poštovního klienta) a díky řadě freemailů jsou náklady na případné zřízení nového účtu nulové.

### 3.3 Instant messaging

Pod pojmem Instant messaging se skrývá řada protokolů a jim příslušejících programů pro textovou online komunikaci. Asi nejznámějším protokolem je ICQ, existuje ale i řada dalších.

I když zde velikostní omezení většinou nejsou, narážíme na problém nutnosti připojení obou účastníků transferu. V případě větší skupiny je možné použít příjemce dál jako odesílatele, nicméně pro úspěšnou distribuci celé skupině stále zůstává podmínka, že v případě připojení uživatele bez dat musí být alespoň jeden s daty online. Za předpokladu, že skupinu budou tvořit uživatelé na opačných stranách Země, může časový posun rapidně znepříjemnit vlastní výměnu (výměny). Jelikož se přenos

realizuje v reálném čase, jeho rychlost závisí na nejpomalejším článku celého řetězu. Další možné problémy mohou nastat při komunikaci s různými klienty (např. posílání souborů mezi klienty ICQ a Miranda).

### **3.4 File Transfer Protocol (FTP)**

FTP – File Transfer Protocol je určen primárně právě k přenosu souborů. Pro mnohé se může jevit jako ideální řešení, i zde však lze nalézt řadu negativ. Pro využívání FTP je nutné mít počítač, na kterém poběží FTP server. Ten by měl mít podle plánované zátěže co nejlepší konektivitu a podle požadovaného vytížení i dostatečné skladovací prostory (dostatek disků). I za předpokladu, že tyto požadavky budou splněny, může být služba pro řadu uživatelů nepoužitelná. Důvody jsou dva. Prvním je nutnost alespoň elementární znalosti fungování protokolu a v ideálním případě i funkční klient, i když stahovat ze serveru lze také ve webovém prohlížeči. Druhým důvodem nefunkčnosti služby je blokování této služby ze strany poskytovatelů připojení, kteří se tak brání nadměrnému provozu a přetěžování svých linek.

### **3.5 Peer To Peer (P2P)**

Vývoj Peer To Peer sítí lze rozdělit na 3 generace. Za praotce všech těchto sítí je považován Napster, který se objevil v roce 1999 a zařazuje se mezi centralizované sítě (první generace). Všichni uživatelé se připojovali k centrálnímu serveru, který uchovával informace o připojených uživateliích a sdílených souborech. Server zde zajišťoval funkci zprostředkovatele přímého spojení mezi klienty. Už tato první P2P síť byla trnem v oku mezinárodní asociace nahrávacího průmyslu, protože majoritní část dat šířená přes P2P i dnes porušuje autorská práva. Vzhledem k centralizované struktuře bylo jednoduché tuto síť odstavit z provozu. I přesto se Napster stal se svými 28 miliony instalací jednou z nejúspěšnějších služeb všech dob.

Napster, respektive myšlenka P2P inspirovala řadu jeho pokračovatelů, kteří se poučili z chyb svých předchůdců. Jasným pokračováním musela být decentralizace, díky které nebylo možné jedním přímým zásahem zlikvidovat celou službu. Klienti se propojovali přímo mezi sebou bez nutnosti využít k propojení centrální server, vyhledávání bylo řešeno řetězovým způsobem (dotaz byl přeposílán od klienta ke klientovi). Tento způsob s sebou přinášel nadměrnou zátěž

Třetí generace hybridních sítí spojuje výhody obou předchozích. Mezi klienty jsou vytvářeny superuzly, které fungují obdobným způsobem jako server u centralizovaných, ovšem pouze pro menší skupinu. Sami komunikují mezi sebou, jejich výběr probíhá náhodně, často bez vědomí uživatelů.

Peer To Peer sítě jsou vhodné pro masovou distribuci dat. Jejich nasazení pro výměnu v rámci malé skupiny s sebou přináší hned několik úskalí a to nutnost instalace klienta zvolené sítě, minimální nebo žádné zabezpečení proti stahování třetími stranami, vzhledem k mnohdy jedinému zdroji dat

značně omezené výhody této služby (např. stahování různých částí souboru od různých poskytovatelů současně), často znemožněná funkce stahování bez současného sdílení dat (mnohdy nemalého objemu) a v neposlední řadě také komplikovanost některých sítí a jejich klientů.

## 3.6 Peer To Mail (P2M)

Pod pro mnohé neznámou zkratkou P2M (Peer To Mail) se skrývá v jistém slova smyslu mutace P2P a e-mailu. Využívá skutečnosti, že řada portálů, které nabízí zřízení e-mailové schránky zdarma, poskytuje pro tuto službu relativně velký diskový prostor (ve srovnání s velikostí e-mailové komunikace), který se pohybuje v řádech jednotek gigabytů. Přímou se tak nabízí možnost zřízení řady účtů, které místo svého primárního účelu budou sloužit jako úložiště dat. Data se do schránky nahrají jednou a stačí již jen rozdistribuovat přihlašovací údaje. Pro ještě snadnější použití existuje řada klientů, které jsou koncipovány přímo pro tento účel. Nevýhodou tohoto způsobu sdílení dat je pro mnohé malá kapacita a rovná práva přístupu pro všechny zúčastněné.

## 3.7 Výměnné servery

V oblasti výměny souborů jsou výměnné servery hitem poslední doby (nejznámější asi [www.rapidshare.de](http://www.rapidshare.de)). Poskytovatel dat je nahraje na server a příjemcům pouze rozešle adresu, odkud si mohou data stáhnout. Bezkonkurenční řešení. Na první pohled je ale jasné, že nulové náklady jsou zde utopií. Majitelé těchto serverů stanovují různé způsoby jak lze jejich služeb využívat. Nejčastější je kombinace placeného a neplaceného přístupu. Uživatel může po zaplacení využívat služeb serveru bez omezení (po dobu danou výší poplatku). Neplaticí uživatelé jsou většinou omezeni objemem dat, které mohou stáhnout za určitý čas. Toto omezení se většinou váže na veřejnou IP adresu, proto je pro řadu uživatelů téměř nemožné těchto služeb využívat (např. zákazníci jednoho poskytovatele internetu, který má minimum veřejných IP adres). Dalším oblíbeným omezením je počet uživatelů zároveň stahujících z jedné země. Velmi často se zde setkáváme i s problémem mazání souborů a to jak ze strany poskytovatele služby (soubory nahrané neplaticím uživatelem, které nikdo po určitý časový interval nestáhl), tak ze strany samotných uživatelů. Těm stačí pro zamezení přístupnosti daného souboru často jedno kliknutí, které je primárně určeno minimalizaci sdílení dat chráněných autorskými právy.

## 3.8 Další služby

Výše uvedené možnosti nejsou jediné. Existuje řada služeb (např. [www.spoluzaci.cz](http://www.spoluzaci.cz)), které sdílení souborů nabízejí jako doplňkovou službu, většinou však značně omezenou (nejčastěji velikostí souborů).

## 4 Specifikace požadavků

### 4.1 Cílové zaměření aplikace

Specifikace požadavků, které má vlastní aplikace splňovat, vychází z jejího určení. Primární způsob používání je co nejsnadnější výměna souborů v rámci skupiny uživatelů. Nezbytným aspektem je jednoduchost ovládání a minimalizace hardwarových i softwarových nároků, ideálně i plná funkčnost v rámci prostředků poskytovaných současnými operačními systémy (internetový prohlížeč).

Fyzicky budou data uložena na serveru, na kterém bude služba spuštěna. Tato varianta se jeví jako nejideálnější, protože nevyžaduje současné připojení odesilatele i příjemce, umožňuje paralelní stahování novinek více uživatelům najednou a tím minimalizuje nároky na klienta (jeden z požadavků uvedených výše). Nevýhodou je nutnost kvalitního hardwaru serveru, kterou lze ovšem obejít zakoupením adekvátního webhostingu.

Jako vzorový příklad použití může sloužit skupina studentů, kteří pracují na kolektivním projektu. Potřebují veškeré změny, které provede jedinec, co nejrychleji rozšířit zbytku skupiny. Takto sdílená data mohou být libovolné soubory (zdrojové kódy, seminární práce, zápisky z přednášek, atd.). Jako další možnost využití se nabízí firemní pracovní skupiny, tímto způsobem může komunikovat (sdílet data) i celá firemní jednotka, popř. firma jako celek. Záleží pouze na vhodně zvolené strukturalizaci do skupin.

### 4.2 Specifikace požadavků na systém

#### 4.2.1 Uživatelé

Pro plné využívání aplikace bude nutná registrace, která však bude vyžadovat minimum údajů, a funkční e-mail. Každý uživatel může být členem skupiny (podrobněji uvedeno níže). Přístup k systému bude možný s různými oprávněními, která budou odlišovat uživatele od administrátorů.

#### 4.2.2 Administrátoři

Systém jako celek bude vyžadovat existenci alespoň jednoho administrátora, jejich počet však nebude omezen (vhodnost více než jednoho administrátora je zřejmá). Každá skupina může mít svého administrátora, který bude zodpovědný za aktivity své skupiny. Globální administrátor bude oprávněn nastavovat parametry (maximální velikost souboru apod.) pro jednotlivé skupiny a uživatele.

### **4.2.3 Skupiny**

Skupiny jsou nezbytným prvkem pro fungování celého systému. Každý uživatel může být členem několika skupin. V rámci každé skupiny sdílí data pouze s ostatními členy, v systému volí aktuální skupinu, v rámci které se momentálně pohybuje. Pro změnu skupiny nebude nutné opětovně přihlášené do systému. V rámci skupiny mají její členové rovná práva, příslušnost ke skupině určuje její administrátor, respektive systémový administrátor.

### **4.2.4 Zabezpečení**

Registrace uživatelů je nutná z důvodu přihlašování do systému. Anonymní přístup nebude možný. Další nutnou funkcí je přísné ošetření přístupu uživatelů pouze k datům, ke kterým má daný uživatel oprávnění. Z hlediska uživatelského komfortu bude možné trvalé přihlášení do systému v rámci jednoho počítače (obdobně jako u řady internetových fór).

### **4.2.5 Komunikace**

V rámci zvýšení komfortu používání aplikace je nutné implementovat komunikační systém. Ten bude mít za úkol automaticky (ale i manuálně) upozornit členy skupiny na jakékoliv změny v datech, která skupina sdílí na serveru. Dále bude zodpovědný za komunikaci v rámci skupiny (poznámky k jednotlivým verzím, dodatky, připomínky apod.).

### **4.2.6 Uživatelské rozhraní**

U uživatelského rozhraní je nutné klást důraz na snadnost používání aplikace jako celku. Přímou specifikace požaduje co nejjednodušší manipulaci s velkými počty souborů, respektive jejich velkým objemem. Bude proto nutné implementovat funkce, které budou usnadňovat výběr souborů, se kterými chce uživatel dále pracovat (např. stáhnout), stejně jako jejich co nejpřehlednější zobrazování. Zde se nabízí možnost volby různých pohledů na seznamy souborů na serveru. Připustíme-li variantu, že v rámci jedné skupiny bude na serveru uloženo množství souborů v řádech stovek (čehož není problém dosáhnout i nezáměrně, např. mnoho malých souborů rozsáhlé webové aplikace, na jejímž vývoji se podílí několik programátorů tvořících tým), stane se implementace vyhledávače nezbytností. Stejně jako pro práci s daty přímo na serveru bude nutné ošetřit i hromadné nahrávání většího počtu souborů na server (manuální upload dvaceti souborů po jednom odradí velmi rychle i ostřílené uživatele).



## 4.3 Specifikace požadavků na server

Jelikož nebude aplikace tvořena statickými HTML stránkami, je nezbytně nutné, aby na serveru bylo k dispozici jak PHP tak MySQL. Kombinace těchto dvou produktů postačuje k tvorbě plnohodnotné aplikace, málokterý z poskytovatelů webhostingu je nenabízí (včetně poskytovatelů zdarma), pro náročnost aplikace je jejich dimenzování postačující a v neposlední řadě jsou oba zadarmo. Jedním z omezujících faktorů je často velikost diskového prostoru, který poskytovatelé webhostingu nabízejí svým klientům, proto je nutné před vlastním nasazením aplikace zvážit předpokládaný objem dat, který bude na serveru uložen a podle toho vybrat vhodně jeho umístění. Pokud se předpokládá větší počet uživatelů, kteří budou službu využívat, je vhodné, aby měl server co nejkvalitnější připojení k internetu.

## 4.4 Specifikace požadavků na klienta

V rámci zachování co nejmenších nároků na klienta bude aplikace vyžadovat pouze internetový prohlížeč. Pro zachování stoprocentní funkčnosti bude nutné mít povolen v prohlížeči JavaScript, v případě zákazu sice vlastní funkčnost aplikace nebude ohrožena, zákaz se ale projeví na míře uživatelského komfortu. Nutnost připojení k internetu je zřejmá.

## 4.5 Omezení aplikace

Omezení aplikace vychází hlavně z omezení technologií PHP a MySQL a z hardwarové konfigurace serveru. Kromě omezení uvedených výše v požadavcích na server je nutné zohlednit i maximální velikosti databází, se kterými je schopné MySQL pracovat. Tyto limity jsou ovšem o několik řádů vyšší, než předpokládané maximální dimenzování aplikace.

# 5 Návrh řešení

## 5.1 Implementační rozhraní

Aplikace bude implementována pomocí PHP, MySQL a JavaScriptu. Tato kombinace byla zvolena hlavně proto, že se ve všech třech případech jedná o freeware a kombinace prvních dvou je schopna posloužit plně k požadovanému účelu. Zahrnutí JavaScriptu mezi implementační jazyky bylo zvoleno k minimalizaci nevýhod serverově interpretovaného skriptovacího jazyka (PHP). Bude tak primárně určen ke kontrolám různých vstupů, aby se omezil přenos klient-server na minimum a na server putovala syntakticky správná data (za jejich sémantiku už si zodpovídají sami uživatelé). Další využití se nabízí v podobě zvýšení komfortu při používání aplikace. Nezanedbatelným hlediskem volby je i fakt, že aplikace nebude kromě internetového prohlížeče na straně klienta vyžadovat žádné další programové vybavení.

Na straně klienta bude implementace probíhat tak, aby byla zajištěna korektní funkčnost ve třech nepoužívanějších internetových prohlížečích, kterými jsou Microsoft Internet Explorer, Mozilla Firefox a Opera.

Testovací verze bude umístěna na školním studentském serveru <http://eva.fit.vutbr.cz/>, který poskytuje záruku funkčnosti a běží na něm jak PHP tak MySQL. Samotný vývoj bude probíhat na lokální instalaci serveru s PHP a MySQL (balík EasyPHP) ve shodné konfiguraci se školním serverem, transfer na server bude probíhat v etapách.

## 5.2 Systém ukládání dat na serveru

I když umožňuje MySQL ukládat do databáze i soubory, v aplikaci bude sloužit pouze pro uchovávání údajů o užívatelích, jejich přihlašovacích jménech, heslech a právech. Tato část bude úzce provázána s částí, která bude uchovávat strukturu uložených dat. Vlastní data budou uchovávána v adresářové struktuře, která však bude generovaná z důvodu vyšší bezpečnosti (více v návrhu zabezpečení). Otázka sdílení stejných dat mezi více skupinami bude řešena na uživatelské úrovni dvěma možnými způsoby. Prvním bude fyzická duplikace pro separované využití jednotlivými skupinami, druhým bude jeden fyzický výskyt dat na serveru a vytvoření specializované skupiny (i jednorázové) přímo pro tato data.

## **5.3 Návrh implementace jednotlivých požadavků**

### **5.3.1 Uživatelé**

U uživatelů se budou uchovávat v databázi údaje přihlašovací jméno, heslo, e-mail, práva a zařazení do skupin(y). Uživatel musí být členem alespoň jedné skupiny. Nově registrovaný uživatel bude automaticky zařazen do skupiny Nezařazení. Aby mu byla umožněna práce se soubory, bude nucen se nechat zařadit do některé z existujících skupin (což řeší skupinový administrátor), nebo skupinu založit (a stát se jejím administrátorem).

### **5.3.2 Administrátoři**

#### **5.3.2.1 Systémový administrátor**

Systémový administrátor bude zodpovědný za kontrolu chování skupin. U každé nově vzniklé skupiny bude specifikovat limity datového provozu (maximální velikost souboru apod.) a jako superuživatel bude oprávněn k veškerým úkonům, které bude systém poskytovat.

#### **5.3.2.2 Skupinový administrátor**

Skupinový administrátor bude plnit obdobné funkce jako administrátor systémový, jeho hlavní působnost ale bude věnována skupině. Bude zodpovědný za členskou skladbu skupiny.

### **5.3.3 Skupiny**

Uživatel, který založí skupinu se automaticky stává jejím administrátorem. Tento post lze kdykoliv předat kterémukoliv členu skupiny. Opět pouze skupinový administrátor může skupinu zrušit. Tento úkon spočívá v likvidaci veškerých dat, která se váží na skupinu a smazání příslušných záznamů v příslušnosti uživatelů do skupin. Ti, kteří se smazáním skupiny stanou Nezařazení, musí postupovat stejně jako v případě nového uživatele (požádat o přijetí do skupiny nebo ji sami založit).

V databázi bude tabulka uchovávající informace o skupinách nosičem názvu skupiny a jejím administrátorem. Příslušnost uživatelů do skupin bude řešena zvláštní vazební tabulkou.

### **5.3.4 Zabezpečení**

Aplikace bude zabezpečena proti neoprávněnému přístupu na několika úrovních. První z nich bude povinná registrace všech uživatelů, což zamezí anonymnímu přístupu. Zařazení do již existující skupiny bude podmíněna schválením žádosti příslušného administrátora, v případě vytvoření skupiny nové bude obsah jejích dat nulový. Jelikož budou vlastní data uchovávána v adresářové struktuře, mohl by se k nim útočník dostat i zadáním přímé adresy souboru, pokud by znal systém uložení.

Proto bude adresářová struktura generovaná náhodně, aby znalost názvu souboru, skupiny, popř. uživatelů nevedla k nabourávání systému z pozice členů jiných skupin.

### **5.3.5 Komunikace**

Funkční e-mail je jedním z mála požadavků na uživatele. I když by bylo možné se bez něj zcela obejít a implementovat obdobu e-mailové korespondence přímo do systému, jak můžeme vidět u řady internetových fór na funkci soukromých zpráv, používání této náhražky by znemožňovalo jakoukoliv komunikaci s vnějším světem.

V případě jakýchkoliv změn ve struktuře souborů skupiny (nahrání, smazání) budou automaticky vytvořeny zprávy popisující tyto modifikace, které budou ihned rozeslány všem členům skupiny. Každá skupina bude mít i speciální stránku plnící funkci nástěnky, kam každý ze členů bude moci psát své postřehy, příspěvky apod., které budou také automaticky rozeslány e-mailem. Každý z uživatelů si bude ve svém profilu volit jeden z řady předdefinovaných vzorů zasílaných zpráv (různě podrobné). Toto vylepšení je zde kvůli možnosti přeměrování příchozích zpráv na mobilní telefon (mobilní e-mail), kde se setkáváme s délkovým omezením zpráv. Na druhou stranu uživatel, který si takovéto nastavení vytvoří, bude neustále informován o veškerých změnách v datech skupiny. V dnešní době, kdy i mobilní připojení k internetu lze pořídit za rozumný finanční obnos tak může být uživatel neustále u aktuálních souborů (což je jeden z primárních cílů aplikace).

### **5.3.6 Uživatelské rozhraní**

Jelikož lze předpokládat, že se vyskytnou skupiny, u kterých bude vysoký počet sdílených souborů, je nutné implementovat funkci vyhledávání, a uživatelsky volitelný způsob zobrazení seznamu přístupných souborů. Pro další manipulaci (mazání, stažení) bude implementována funkce filtru, kde uživatel vybere, se kterými daty bude pracovat. Pro stahování bude implementována možnost stažení výběru jako jednoho komprimovaného souboru. Obdobně bude řešeno i nahrávání více souborů na server.

## **6 Závěr**

### **6.1 Dosažené výsledky**

Cílem tohoto semestrálního projektu bylo seznámení se s existujícími službami a nástroji pro výměnu dat. Na základě jejich analýzy, zjištění pozitiv a negativ, které každá služba nese, byly stanoveny požadavky na výslednou aplikaci a posléze byl proveden návrh implementace. Každému kroku uvedeného postupu odpovídá příslušná kapitola, která podrobně vysvětluje jednotlivé dílčí poznatky a závěry.

### **6.2 Zhodnocení průběhu práce**

Je jasné, že ideální služba pro sdílení dat neexistuje. Vždy záleží na tom, jaký účel má plnit. Těžko si lze představit, jak kvůli několika kilobytům zdrojového kódu skupina uživatelů hromadně nainstaluje klienta Peer To Peer sítě (v podstatě libovolné). Stejně komicky působí i druhý extrém, když obraz CD bude distribuován e-mailem. Pokud známe požadavky na aplikaci, můžeme ji už od návrhu tvořit s vědomím způsobu budoucího použití. Výsledek tak bude plnit požadované funkce zcela, nebude obtěžovat uživatele a hardwarově nebo softwarově zbytečně vytěžovat zúčastněné.

Tento projekt volně přechází ve vlastní Diplomovou práci, jíž je součástí. V rámci ní proběhne implementace návrhu systému a výsledkem bude funkční aplikace běžící na školním serveru pro případnou demonstraci jednotlivých funkcí.

# Literatura

[1] HERODEK, Martin. *Knihovnický systém*. [s.l.], 2005. 27 s. Bakalářská práce.

[2] LASEK, Petr. *P2P síť - třetí úhel pohledu* [online]. 23.11.2004 [cit. 2007-01-02]. Dostupný z

WWW: <<http://www.isdn.cz/clanek.php?cid=6143>>.