

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**



**SLUŽBA NETMIND PRO UPOZORŇOVÁNÍ  
NA ZMĚNU WEBU**

**Semestrální práce**

**Brno, 2006**

**David Ondráček**

## Zadání

1. Seznamte se s technologiemi publikování na Internetu, zejména HTML, XML, PHP apod.
2. Navrhněte systém, který bude sledovat obsah zadaných internetových stránek a upozorňovat na jejich změnu. Při návrhu se inspirujte systémem woko.cz.
3. Uvedený systém implementujte. Implementujte vhodné upozorňování uživatele.
4. Diskutujte získané znalosti a možnost dalšího rozšíření projektu.

## **Prohlášení**

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením Ing. Pavla Očenáška. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Brně dne 28.12.2006

.....  
David Ondráček

## **Abstrakt**

Cílem semestrálního projektu je navrhnout a implementovat službu pro upozorňování uživatelů na změnu obsahu webu. To je užitečné zejména v případě stránek, jejichž obsah se mění nepravidelně. Uživatel tak nemusí kontrolovat web manuálně. Stačí, když si v systému zaregistruje jeho sledování. Služba se pak postará o automatickou kontrolu obsahu webu a v momentě, kdy dojde k nějaké změně, uživatel je o této skutečnosti informován e-mailem. Pokud uživatel sleduje takovýchto zdrojů více, může mu služba ušetřit nezanedbatelné množství času.

## **Klíčová slova**

kontrola internetových stránek, změna obsahu webu, netmind, technologie publikování na internetu

## **Abstract**

The main goal of this semestral project is to propose and implement service, which would be able to detect changes on web-sites and remind users about them. This is useful especially in case of web-sites, where content changes irregularly. User doesn't have to check web page manually. The only thing he has to do is register its monitoring in the system. The service will take care of checking the content of web-site automatically. When it detects some changes, user will be notified of them by e-mail. The service can be very time saving for user, especially in case of monitoring more internet sources.

## **Key words**

web-site monitoring, changes of web-site content, netmind, technologies of publishing informations on internet

## Obsah

1	Úvod.....	7
1.1	Struktura práce.....	7
1.2	Cíle práce.....	7
1.3	Současný stav řešené problematiky .....	8
2	Technologie webu .....	8
2.1	HTML (HyperText Markup Language).....	8
2.2	XML (eXtensible Markup Language) .....	9
2.3	PHP (PHP Hypertext Preprocessor) .....	10
2.4	MySQL.....	10
3	Analýza .....	11
3.1	Neformální specifikace požadavků.....	11
3.2	Diagram případů použití .....	11
3.3	Popis případů použití .....	12
4	Návrh systému .....	17
4.1	Návrh databáze – E-R diagram.....	17
4.2	Kontrola změny obsahu webu .....	17
4.2.1	MD5 (Message-Digest Algorithm 5) .....	18
5	Implementace.....	18
5.1	Tabulky (relace).....	18
5.2	Proces registrace uživatele .....	18
5.3	Zaslání hesla .....	19
5.4	Přihlášení, nastavení účtu.....	19
5.5	Nastavení sledování .....	19
5.6	Robot.....	20
6	Nasazení, testování.....	21
7	Funkčnost a omezení vytvořeného programu .....	21
8	Rozšíření.....	22
9	Závěr.....	22
	Literatura.....	23
	Příloha A .....	25
	Příloha B .....	27
	Příloha C .....	29

# 1 Úvod

V dnešní době je internet nepostradatelným a snad i hlavním zdrojem informací. Mnoho lidí se zajímá o informace z několika, ne-li mnoha, vědních oborů. Ať už kvůli práci nebo jen ze zvědavosti. Když k tomu přidáme množství webových stránek týkajících se nejrůznějších koníčků a zálib, stránek se zaměřením na zpravodajství apod., dostaneme poměrně velký počet zdrojů, které je potřeba sledovat. To může být časově velmi náročné. Navíc obsah mnoha webů se mění velmi nepravidelně, čili čas věnovaný kontrole aktualizací nemusí být produktivní. Tento problém je možné vyřešit vhodnou webovou aplikací, která by kontrolovala obsah webu automaticky. Uživatel by si jen zaregistroval adresy stránek, které chce sledovat a jejich změnu by nadále pravidelně kontroloval robot. V případě aktualizace webu by byl uživatel vhodným způsobem (například e-mailem) o této skutečnosti informován. Náplní semestrálního projektu je návrh a implementace takovéto aplikace.

## 1.1 Struktura práce

Na začátku práce je krátce pojednáno o současných technologiích webu, jejichž využití by pro implementaci systému připadalo v úvahu (kapitola 2 – Technologie webu). Následuje analýza problematiky, která obsahuje neformální specifikaci požadavků na systém a diagram a popis případů použití (kapitola 3 – Analýza). Návrh systému je náplní kapitoly následující (kap. 4 – Návrh systému). Jeho součástí je E-R diagram databáze a návrh systému kontroly změny obsahu. Kapitola 5 (Implementace) obsahuje nejdůležitější informace týkající se implementace systému. V další kapitole je uveden odkaz na implementovanou službu pro testování (kapitola 6 – Nasazení, testování). V kapitole 7 jsou shrnuty omezení implementovaného programu. Předposlední kapitola (kapitola 8 – Rozšíření) obsahuje několik návrhů na vylepšení systému v rámci navazující diplomové práce. V závěrečné kapitole jsou zhodnoceny dosažené výsledky. Nakonec je uveden seznam použité literatury.

## 1.2 Cíle práce

Cílem této práce je návrh a implementace základu aplikace popsané v úvodu. Aplikace by měla být schopna uchovávat v databázi informace o uživateli, o registrovaných stránkách a jejich aktuálním obsahu, pravidelně je kontrolovat a v případě změny upozornit e-mailem všechny uživatele, kteří mají danou stránku zaregistrovanou. Implementace dalších rozšíření, zde také diskutovaných, bude náplní navazující diplomové práce. Jedná se např. o více možností upozorňování (SMS, ICQ, apod.), sledování přesměrování stránek či možnost volby intervalu kontroly.

## 1.3 Současný stav řešené problematiky

Podobně zaměřené služby již existují. Jedná se například o aplikaci NetMind (momentálně není v provozu), ChangeDetect, či aplikace českých autorů WOKO.cz, jež mi byla inspirací. Službu bych chtěl v budoucnosti doplnit o rozšíření uvedená v kapitole 8.

## 2 Technologie webu

Ze současných technologií webu, které by bylo možné pro implementaci systému použít, bych se zmínil o HTML, XML, PHP a MySQL. Jejich úplný výklad není hlavní náplní této práce (avšak základní přiblížení je vhodné pro pochopení dalšího textu), proto o nich pojednám jen stručně.

### 2.1 HTML (HyperText Markup Language)

HTML je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému WWW (World Wide Web), jenž umožňuje publikaci stránek na internetu. Je aplikací univerzální značkovacího jazyka SGML (Standard Generalized Markup Language).

V zásadě se HTML dokument skládá z množiny tzv. značek (tagů) a čistého textu. Značky jsou ohraničeny speciálními znaky “<“ a “>“. Jsou obvykle párové (rozlišují se tedy počáteční a koncové značky), ale existují i nepárové. Část dokumentu uzavřená mezi značkami tvoří tzv. prvek (element) dokumentu. Element může obsahovat další vnořené elementy. U každého elementu je možné definovat jeho vlastnosti (atributy).

Příklad elementu s párovými značkami:

```
<a href=http://www.example.com>odkaz</a>
```

Tento element je odkazem na adresu <http://www.example.com>.

Příklad elementu s nepárovou značkou:

```
<br>
```

Tento element označuje zalomení řádku.

Značky lze v základě rozdělit na strukturální, popisné (sémantické) a stylistické. Strukturální značky rozvrhují strukturu dokumentu (například odstavce <p>). Sémantické popisují význam obsahu



elementu (například název dokumentu <title>). Stylistické značky určují vzhled elementu při zobrazení (například značka pro tučné písmo <b>).

K HTML dokumentu lze také připojit tzv. styl, který definuje vzhled jednotlivých elementů. K tomuto účelu se nejčastěji používá CSS (Cascading Style Sheets – kaskádové styly).

Vývoj jazyka HTML byl ukončen verzí 4.01. Stalo se tak z mnoha důvodů. Jedním z nich je existence velkého množství dokumentů, jež nevyhovují specifikaci HTML – obsahují chyby (neuzavřené, překřížené či špatně vnořené elementy apod.). Webové prohlížeče (jejichž vývoj a nekompatibilita byly zároveň příčinou problémů) pro klasické desktopové počítače či notebooky s tímto počítají a snaží se chyby kompenzovat. Takovýto princip však není možné použít u méně výkonných zařízení jako jsou mobilní telefony a kapesní počítače. Dalším důvodem jsou poměrně omezené vyjadřovací schopnosti jazyka. Je tedy třeba použít jazyka s přesně definovanou strukturou a pravidly a také větším potenciálem. Tím je například jazyk XML (viz. dále). V současné době je však jazyk HTML jako prostředek pro publikaci stránek na internetu stále nejrozšířenější.

## 2.2 XML (eXtensible Markup Language)

XML (rozšiřitelný značkovací jazyk) je standard pro obecný značkovací jazyk vyvinutý konsorciem W3C. Vznikl zjednodušením a zároveň zpřísněním pravidel jazyka SGML. Umožňuje snadné vytváření konkrétních značkovacích jazyků (aplikací).

Jedním z problémů jazyka HTML je fakt, že se dnes používá spíše pro popis vzhledu stránky, než pro označení sémantiky jednotlivých elementů. Tímto nedostatkem jazyk XML netrpí. Navíc disponuje řadou dalších vlastností významných pro rozšiřování a vylepšování webových technologií v nejbližší budoucnosti. Mezi ty nejdůležitější patří možnost použít vlastní značky (tagy), což mimo jiné umožňuje vývoj inteligentních vyhledávacích služeb. XML lze s výhodou využít také pro výměnu, sdílení a ukládání informací a pro transformaci dokumentů do různých formátů. Pomocí RDF (Resource Description Framework), lze k dokumentu přidat metadata, což může výrazně podpořit rozvoj sémantického webu. RDF je standard vyvinutý konsorciem W3C. Definuje popis a výměnu metadat. XML formát je také ideální pro komunikaci (přenos dat) mezi klientem (prohlížečem) a serverem.

XML dokument se skládá z elementů podobně jako v HTML. Elementy se tvoří značkami (tagy) a je možné je do sebe vnořovat. Na rozdíl od HTML se nesmí křížit (vnořený element musí být ukončen uvnitř nadřazeného elementu) a musí být vždy uzavřený. V případě elementu s prázdným obsahem není nutné uvádět i koncovou značku, stačí element uzavřít použitím znaku “/” před ukončovacím znakem počáteční značky (např. místo <br></br> lze použít <br />).

XML nalézá díky své otevřenosti a flexibilitě v oblasti technologií webu stále více uplatnění.

## 2.3 PHP (PHP Hypertext Preprocessor)

PHP je skriptovací programovací jazyk, který se používá především pro tvorbu dynamických webových stránek. PHP kód se začleňuje přímo do struktury jazyka HTML. Interpretuje se na straně serveru, prohlížeč dostane jako odpověď již vygenerovaný dokument.

Syntaxe PHP vychází z několika programovacích jazyků – Perl, C, Pascal a Java.

Jazyk PHP je nezávislý na platformě, obsahuje velké množství knihoven funkcí pro zpracování textu, grafiky, umožňuje práci se soubory a mnoha databázovými systémy (MySQL, PostgreSQL, ODBC, Oracle, atd.), podporuje mnoho internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, atd.).

Od verze 5 PHP velmi dobře podporuje objektově orientovaný přístup.

Pro své vlastnosti se PHP velmi často používá v kombinaci s MySQL či PostgreSQL k tvorbě webových aplikací.

## 2.4 MySQL

MySQL je databázový systém vytvořený švédskou firmou MYSQL AB. Jedná se o multiplatformní databázi. Komunikace s ní probíhá pomocí jazyka SQL. Pro svou jednoduchou implementovatelnost (lze jej instalovat na Linux, Microsoft Windows a jiné operační systémy), výkon a především možnost volného šíření, se v současné době používá velmi často. Většinou se používá v kombinaci s PHP a Apache (webový server).

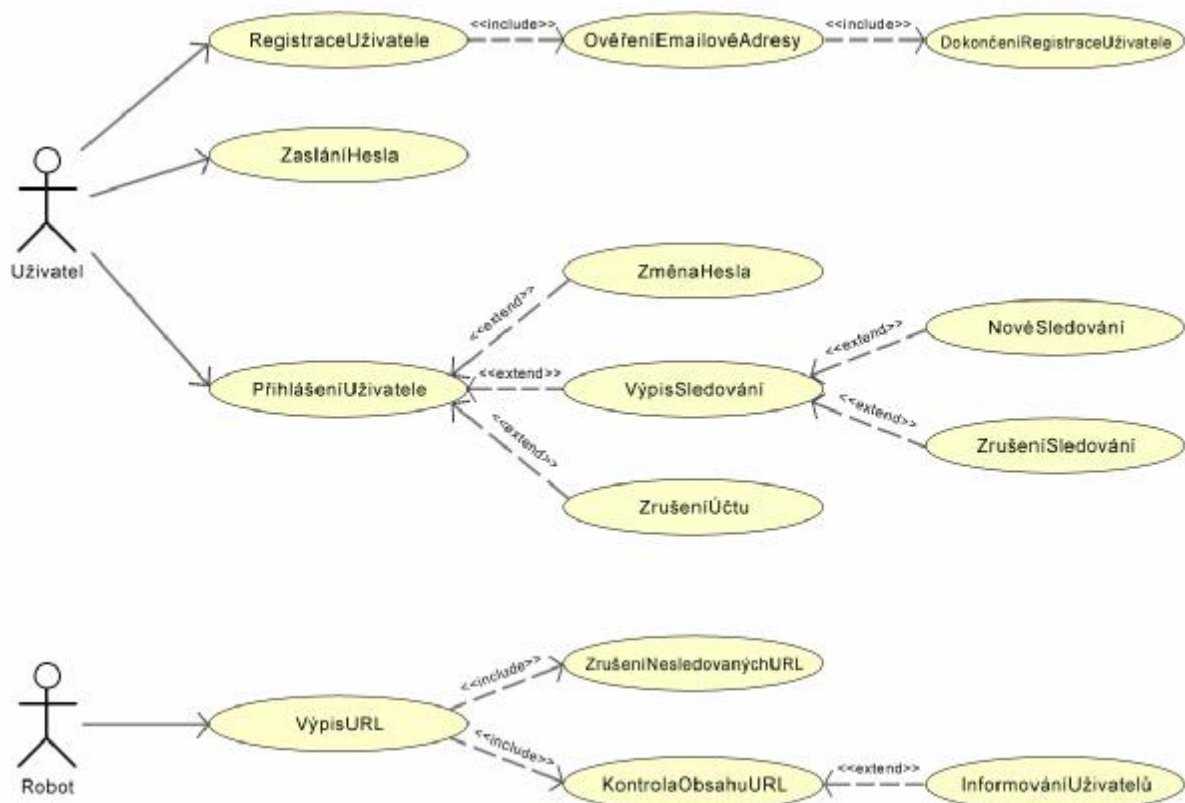
MySQL bylo od počátku optimalizováno spíše pro rychlost, některé funkce tedy nebyly implementovány. Například zálohování bylo velmi zjednodušené, nebyly podporovány pohledy, trigger ani uložené procedury. Podpora těchto vlastností je otázkou několika posledních let.

### 3 Analýza

#### 3.1 Neformální specifikace požadavků

System by měl uživatelům umožňovat jednoduchou registraci a přehledné nastavení a rušení sledování stránek. Dále by měl nabízet možnost změnit heslo pro přístup k účtu, jeho opětovné zaslání v případě zapomenutí a také možnost zrušit celý účet. Samozřejmě bude také potřeba ukládat informace o registrovaných stránkách. Robot by měl kontrolovat pouze změnu samotného obsahu stránek. O změně banneru, obrázku nebo například struktury stránky není nutné uživatele informovat. Stránky by se měly kontrolovat v nějakém rozumném intervalu, aby nedocházelo k zbytečnému přetěžování serverů.

#### 3.2 Diagram případů použití



Obr. 1 – Diagram případů použití

### 3.3 Popis případů použití

Případ použití	RegistraceUživatele
<b>ID</b>	1
<b>Stručný popis</b>	Systém od uživatele žádá registrační údaje (e-mail, heslo).
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. Případ použití se spustí, když uživatel zadá do formuláře registrační údaje (e-mail, heslo) a potvrzovací kód z obrázku.</li> <li>2. Dokud je formát zadané e-mailové adresy neplatný nebo je adresa již v systému zaregistrována nebo uživatel zadal nesprávný potvrzovací kód <ol style="list-style-type: none"> <li>2.1. Systém požaduje, aby uživatel zadal svou e-mailovou adresu, heslo, heslo znovu pro potvrzení a potvrzovací kód.</li> <li>2.2. Systém zkontroluje uživatelem zadané údaje.</li> </ol> </li> <li>3. Systém odešle na uživatelem zadanou adresu e-mail s údaji pro dokončení registrace (obsahuje odkaz na stránku s jedinečně vygenerovaným řetězcem jako parametrem v URL).</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 2.2 hlavního toku v případě, že uživatelem zadané údaje nejsou v pořádku.</li> <li>2. Systém informuje uživatele o chybě (nesprávný formát e-mailové adresy / e-mailová adresa je již v systému zaregistrována / nesprávný potvrzovací kód).</li> </ol>
<b>Následné podmínky</b>	1. Uživateli byl odeslán e-mail s údaji pro dokončení registrace.

Případ použití	OvěřeníEmailovéAdresy
<b>ID</b>	2
<b>Stručný popis</b>	Systém čeká na potvrzení e-mailové adresy.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;include&gt;&gt; ID 1 – RegistraceUživatele.</li> <li>2. Systém čeká, až uživatel potvrdí svou e-mailovou adresu navštívením vygenerovaného odkazu, jež byl v e-mailu odeslán.</li> <li>3. Systém zkontroluje, zda je číslo, předané jako parametr v URL navštíveného odkazu, uloženo v databázi čekajících na potvrzení.</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 3 hlavního toku v případě, že číslo, předané v parametru URL, není uloženo v databázi čekajících na potvrzení.</li> <li>2. Systém informuje uživatele o chybě.</li> </ol>
<b>Následné podmínky</b>	1. E-mailová adresa uživatele byla ověřena.

Případ použití	DokončeníRegistraceUživatele
<b>ID</b>	3
<b>Stručný popis</b>	Systém dokončí registraci uživatele.

<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;include&gt;&gt; ID 2 – OvěřeníEmailovéAdresy.</li> <li>2. Systém vloží údaje uživatele do databáze zaregistrovaných.</li> <li>3. Systém informuje uživatele o dokončené registraci.</li> </ol>
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Uživatel byl zaregistrován.

<b>Případ použití</b>	<b>OdesláníHesla</b>
<b>ID</b>	4
<b>Stručný popis</b>	Systém odešle uživateli heslo pro přihlášení.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. Případ použití se spustí, když uživatel zadá do formuláře pro odeslání hesla svoji e-mailovou adresu a opíše potvrzovací kód z obrázku.</li> <li>2. Systém zkontroluje, zda má uživatelem zadaná e-mailová adresa správný formát, zda je v databázi zaregistrovaných uživatelů a zda je správně opsaný kontrolní kód.</li> <li>3. Systém odešle heslo uživateli e-mailem.</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 2 hlavního toku v případě, že žádná e-mailová adresa není v databázi, má nesprávný formát nebo nesouhlasí potvrzovací kód.</li> <li>2. Systém informuje uživatele o chybě.</li> </ol>
<b>Následné podmínky</b>	1. Uživateli bylo e-mailem odesláno heslo pro přístup do systému.

<b>Případ použití</b>	<b>PřihlášeníUživatele</b>
<b>ID</b>	5
<b>Stručný popis</b>	Uživatel se přihlásí do systému.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. Případ použití se spustí, když uživatel zadá své přihlašovací údaje (e-mail, heslo) do formuláře pro přihlášení.</li> <li>2. Systém zkontroluje, zda jsou uvedené údaje správné (zda je uživatel zaregistrován a zadal správné heslo).</li> <li>3. Systém povolí uživateli přístup k nastavení svého účtu a sledování stránek.</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 2 hlavního toku v případě, že uživatel zadal nesprávné přihlašovací údaje.</li> <li>2. Systém zamítne uživateli přístup k nastavení účtu a sledování stránek.</li> </ol>
<b>Následné podmínky</b>	1. Uživatel byl přihlášen.

<b>Případ použití</b>	<b>ZměnaHesla</b>
<b>ID</b>	6
<b>Stručný popis</b>	Uživatel si změnil přihlašovací heslo.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	Uživatel je přihlášen.
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;extend&gt;&gt; ID 5 – PřihlášeníUživatele</li> <li>2. Případ použití se spustí, když uživatel zadá stávající a nové heslo do formuláře pro změnu hesla.</li> <li>3. Systém zkontroluje, zda je zadané stávající heslo správné.</li> <li>4. Systém aktualizuje heslo uživatele v databázi.</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 3 hlavního toku v případě, že stávající heslo nebylo uživatelem zadáno správně.</li> <li>2. Systém informuje uživatele o chybě.</li> </ol>
<b>Následné podmínky</b>	1. Uživateli bylo změněno přihlašovací heslo.

<b>Případ použití</b>	<b>VýpisSledování</b>
<b>ID</b>	7
<b>Stručný popis</b>	Systém vypíše všechna URL, která má uživatel nastavena pro sledování.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	Uživatel je přihlášen.
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;extend&gt;&gt; ID 5 – PřihlášeníUživatele</li> <li>2. Systém vypíše všechna URL, která má uživatel zaregistrována pro sledování.</li> </ol>
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Systém vypsá uživatelem zaregistrovaná URL.

<b>Případ použití</b>	<b>NovéSledování</b>
<b>ID</b>	8
<b>Stručný popis</b>	Systém přidá uživatelem požadované URL do seznamu sledovaných stránek.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	Uživatel je přihlášen.
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;extend&gt;&gt; ID 7 – VýpisSledování.</li> <li>2. Případ použití se spustí, když uživatel zadá URL do formuláře pro nové sledování.</li> <li>3. Systém zkontroluje, zda je možné zadanou URL sledovat.</li> <li>4. Systém přidá URL do databáze.</li> </ol>
<b>Alternativní toky</b>	<ol style="list-style-type: none"> <li>1. Alternativní tok se spustí po kroku 3 hlavního toku v případě, že uživatelem zadané URL nelze sledovat.</li> <li>2. Systém informuje uživatele o chybě.</li> </ol>
<b>Následné podmínky</b>	1. URL bylo přidáno ke sledovaným.

<b>Případ použití</b>	<b>ZrušeníSledování</b>
<b>ID</b>	9
<b>Stručný popis</b>	Systém zruší uživatelem vybrané sledování.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	Uživatel je přihlášen.
<b>Hlavní tok</b>	1. <<extend>> ID 7 – VýpisSledování. 2. Případ použití se spustí, když uživatel zvolí zrušení položky ve výpisu sledovaných URL. 3. Systém odstraní URL ze seznamu sledovaných.
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. URL bylo odstraněno ze seznamu sledovaných.

<b>Případ použití</b>	<b>ZrušeníÚčtu</b>
<b>ID</b>	10
<b>Stručný popis</b>	Systém zruší uživateli účet.
<b>Primární aktéři</b>	Uživatel
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	Uživatel je přihlášen.
<b>Hlavní tok</b>	1. <<extend>> ID 5 – PřihlášeníUživatele. 2. Systém zruší uživateli všechna sledování a vyjme jeho registrační data z databáze.
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Účet uživatele byl zrušen.

<b>Případ použití</b>	<b>VýpisVšechSledovanýchURL</b>
<b>ID</b>	11
<b>Stručný popis</b>	Systém poskytne robotovi seznam všech sledovaných URL.
<b>Primární aktéři</b>	Robot
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	1. Případ použití se spustí, když si robot vyžádá seznam sledovaných URL. 2. Systém poskytne robotovi výpis všech sledovaných URL.
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Seznam sledovaných URL byl poskytnut robotovi.

<b>Případ použití</b>	<b>ZrušeníNesledovanýchURL</b>
<b>ID</b>	12
<b>Stručný popis</b>	Robot zkontroluje, zda se v systému nevyskytují nesledované adresy.
<b>Primární aktéři</b>	Robot
<b>Sekundární aktéři</b>	

<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;include&gt;&gt; ID 11 - VýpisVšechSledovanýchURL.</li> <li>2. Robot zkontroluje, zda se v seznamu nevyskytují URL, která nejsou sledována žádným uživatelem. <ol style="list-style-type: none"> <li>2.1 Pokud takové URL najde, odstraní jej z databáze.</li> </ol> </li> </ol>
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Robot odstraní nesledovaná URL z databáze.

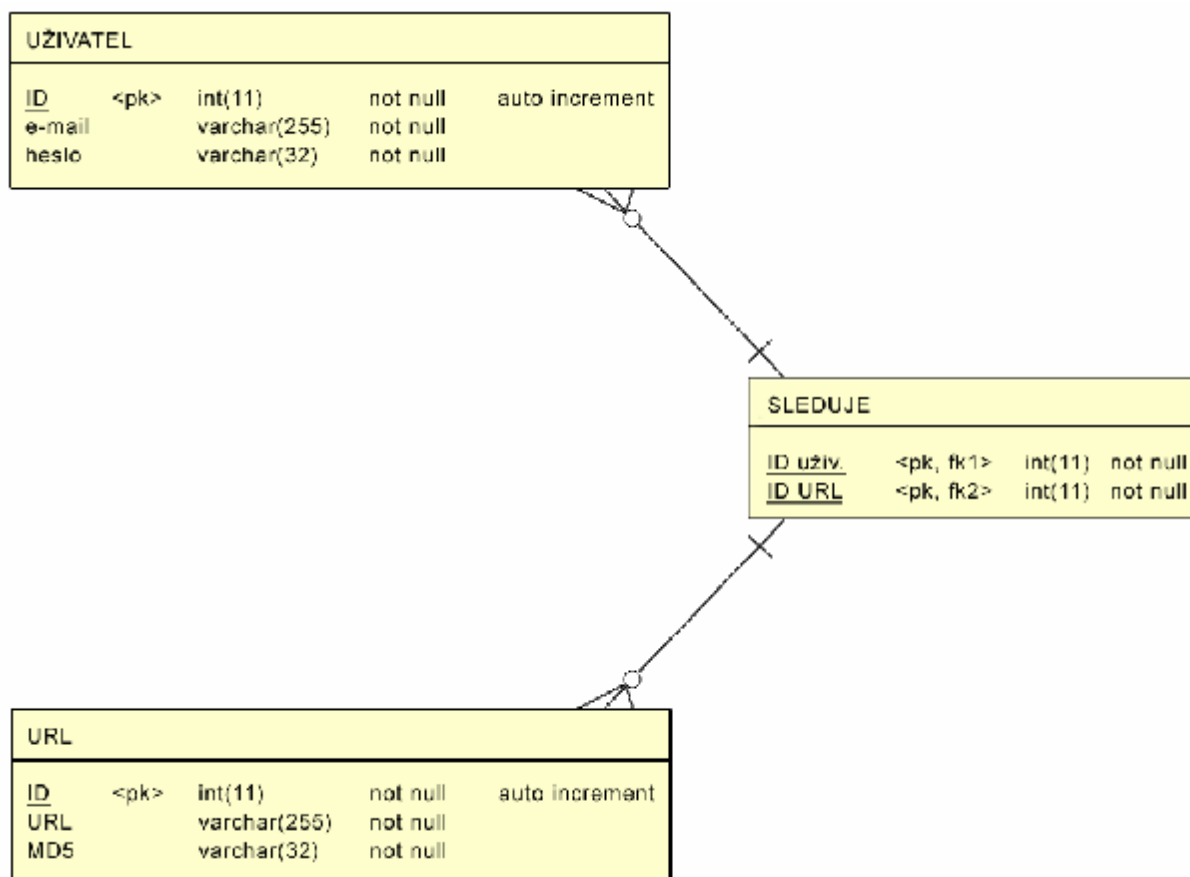
<b>Případ použití</b>	<b>KontrolaObsahuURL</b>
<b>ID</b>	13
<b>Stručný popis</b>	Systém zkontroluje obsah všech URL v seznamu.
<b>Primární aktéři</b>	Robot
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;include&gt;&gt; ID 11 – VýpisVšechSledovanýchURL.</li> <li>2. Dokud není zpracován celý seznam <ol style="list-style-type: none"> <li>2.1. Robot porovná aktuální obsah URL s obsahem při předchozí kontrole.</li> <li>2.2. Robot uloží informace o obsahu webu do databáze.</li> <li>2.3. Robot zkontroluje, zda není na konci seznamu.</li> </ol> </li> </ol>
<b>Alternativní toky</b>	1. Alternativní tok ID 14 - InformováníUživatelů se spustí po kroku 2.1. hlavního toku v případě, že se obsah URL od minulé kontroly změnil.
<b>Následné podmínky</b>	1. Robot zkontroloval obsah všech URL v systému.

<b>Případ použití</b>	<b>InformováníUživatelů</b>
<b>ID</b>	14
<b>Stručný popis</b>	Robot informuje uživatele o změně obsahu webu.
<b>Primární aktéři</b>	Robot
<b>Sekundární aktéři</b>	
<b>Předpoklady</b>	
<b>Hlavní tok</b>	<ol style="list-style-type: none"> <li>1. &lt;&lt;extend&gt;&gt; ID 13 – KontrolaObsahuURL.</li> <li>2. Případ použití se spustí, robot detekuje změnu obsahu webu.</li> <li>3. Robot získá od systému seznam všech uživatelů, kteří mají dané URL zaregistrováno.</li> <li>4. Dokud není zpracován celý seznam <ol style="list-style-type: none"> <li>4.1. Robot odešle e-mail s upozorněním na změnu obsahu webu na adresu uživatele.</li> <li>4.2. Systém zkontroluje, zda prošel celý seznam.</li> </ol> </li> </ol>
<b>Alternativní toky</b>	
<b>Následné podmínky</b>	1. Uživatelé byli informováni o změně obsahu webu.



## 4 Návrh systému

### 4.1 Návrh databáze – E-R diagram



Obr. 2 – E-R diagram

### 4.2 Kontrola změny obsahu webu

Protože většinu uživatelů nebudou zajímat změny designu, struktury, reklamních bannerů či poznámek v kódu, ale pouze změny samotného textu, bude potřeba načtený obsah stránky nejprve odfiltrvat od všech značek (tagů). Pro kontrolu změny lze na zbylý text aplikovat některou z funkcí kontrolního součtu. Její výsledek pak stačí porovnávat s kontrolním součtem v databázi, jenž byl uložen při registraci URL nebo při minulé kontrole. Pokud od poslední kontroly došlo ke změně obsahu, kontrolní součty se budou lišit. Jednou z funkcí kontrolního součtu, která se hojně využívá a dala by se použít i pro tento případ, je MD5.

### 4.2.1 MD5 (Message-Digest Algorithm 5)

MD5 je velmi rozšířená hashovací funkce s otiskem (hashem) o velikosti 128 bitů. Běžně se používá pro kontrolu integrity souborů či ukládání hesel. MD5 hash je typicky 32 znaků dlouhé šestnáctkové číslo, přičemž je velmi nepravděpodobné, že by bylo stejné pro nějaké dva různé inicializační vektory.

Autorem MD5 funkce je Ronald Rivest, profesor MIT (Massachusetts Institute of Technology). Vymyslel ji v roce 1991 jako náhradu za starší hashovací funkci MD4. Stejně jako u MD4 i u MD5 byly později objeveny vady (kolize), proto se od ní v oblasti kryptografie začalo upouštět. Pro kontrolní součty v tomto případě však zcela postačuje.

Příklad:

MD5 ("Tohle je testovací řetězec.") = 41c6539ca3f995df37958af2c36f2434

Pseudokód MD5 funkce lze nalézt v příloze B.

## 5 Implementace

Pro implementaci systému lze, vzhledem k výše uvedené analýze a návrhu, výhodně využít HTML v kombinaci s PHP, databází MySQL a webovým serverem Apache.

### 5.1 Tabulky (relace)

Databáze obsahuje následující tabulky:

accounts (id, email, psw) ... účty uživatelů

urls (id, url, md5sum) ... sledované URL

refs (account\_id, url\_id) ... reference mezi uživateli a URL

reg (email, code, psw) ... využívá se v procesu registrace uživatele (viz. dále)

codes (image\_number, code) ... kontrolní kódy (opatření proti spamovacím robotům – viz. dále)

Skript pro vytvoření tabulek lze nalézt v příloze A.

### 5.2 Proces registrace uživatele

Celý proces registrace uživatele probíhá následovně: uživatel zadá do formuláře registrační údaje (e-mail), heslo a heslo znovu (pro kontrolu). Pro zabezpečení proti rozesílání spamu roboty je také nutné opsat kód z obrázku. Po odeslání formuláře se zkontroluje formát e-mailové adresy a shoda obou

hesel. Zároveň se zkontroluje, zda není e-mailová adresa již v systému. Pokud je vše v pořádku, vygeneruje se náhodný 32 místný kód, který je spolu s registračními údaji uložen do tabulky *reg*. Zároveň se odešle na adresu uživatele e-mail s informacemi pro dokončení registrace. E-mail obsahuje odkaz na skript *ucet.php*, jehož součástí je vygenerovaný kód předaný jako parametr v URL. Pokud uživatel navštíví odeslaný odkaz, kód v URL a kód uložený v databázi se porovnají a pokud jsou shodné, registrace se dokončí - přihlašovací údaje uživatele jsou přesunuty z tabulky *reg* do tabulky *accounts*. Uživatel je informován o úspěšném dokončení registrace a může se přihlásit do systému.

### 5.3 Zaslání hesla

Pokud uživatel zapomene heslo, může si jej nechat znovu poslat e-mailem. Do příslušného formuláře zadá svoji adresu, heslo je podle ní vyhledáno v databázi (tabulka *accounts*) a odesláno e-mailem. Nemůže se tedy stát, že by uživatel takto získal heslo jiného uživatele. I v tomto případě je nutné do formuláře vložit potvrzovací kód z obrázku.

### 5.4 Přihlášení, nastavení účtu

Uživatel zadá své údaje do přihlašovacího formuláře. Po jeho odeslání je zadaný e-mail a heslo porovnáno s údaji v databázi (tabulka *accounts*). Pokud vše souhlasí, uživatel je přihlášen. Po přihlášení může změnit své heslo pro přístup do systému. Také má možnost zrušit celý účet.

### 5.5 Nastavení sledování

Nastavit sledování lze až po přihlášení uživatele. Ten má možnost přidat nové sledování nebo zrušit některé ze stávajících.

V případě nového sledování se zkontroluje, zda je zadaná URL již v databázi (tabulka *urls*). Pokud ano, přidá se jen reference mezi ID uživatele a ID URL do tabulky *refs*. Pokud URL v systému ještě není, pokusí se systém načíst jeho obsah. V případě úspěchu se URL vloží do databáze spolu se zjištěnými údaji o obsahu (viz. dále). Také se vytvoří příslušná reference v tabulce *refs*. Pokud načtení obsahu neproběhne úspěšně, uživatel je informován o chybě.

V případě zrušení sledování se jen odstraní příslušná reference v tabulce *refs*. URL zůstává v tabulce i v případě, že k němu neexistuje žádná reference (žádný uživatel jej nesleduje). O detekování a odstranění takovýchto záznamů se stará robot (viz. dále).

## 5.6 Robot

Implementovaný robot (skript *robot.php*) se spouští automaticky jedenkrát za den. Nejprve projde celou tabulku *urls*, přičemž u každého záznamu zkontroluje, zda nějaký uživatel dané URL sleduje (podle referencí v tabulce *refs*). Pokud neexistuje žádná reference vztahující se k záznamu, je URL z databáze odstraněno. Ze zbylých URL si robot vytvoří seznam pro procházení.

Každá položka v seznamu představuje jedno URL. Robot načte jeho obsah, poté odstraní všechny tagy a ze zbylého obsahu vytvoří kontrolní součet (MD5). Pro odstranění tagů lze využít funkce PHP `ereg_replace()`. Pro kontrolní součet existuje v PHP přímo funkce `md5()`, čili není potřeba ji implementovat.

U každého URL robot porovnává předchozí kontrolní součet s nově zjištěným. Pokud se neshodují, kontrolní součet uložený v databázi se přepíše novým kontrolním součtem. Následně si robot zjistí seznam uživatelů, kteří mají sledování URL zaregistrováno. Podle něj odešle všem uživatelům e-mailem upozornění o změně obsahu.

Automatické spouštění skriptu robota lze v operačním systému Microsoft Windows zajistit pomocí služby „Naplánované úlohy“. Na Unixových systémech lze použít démona Cron.

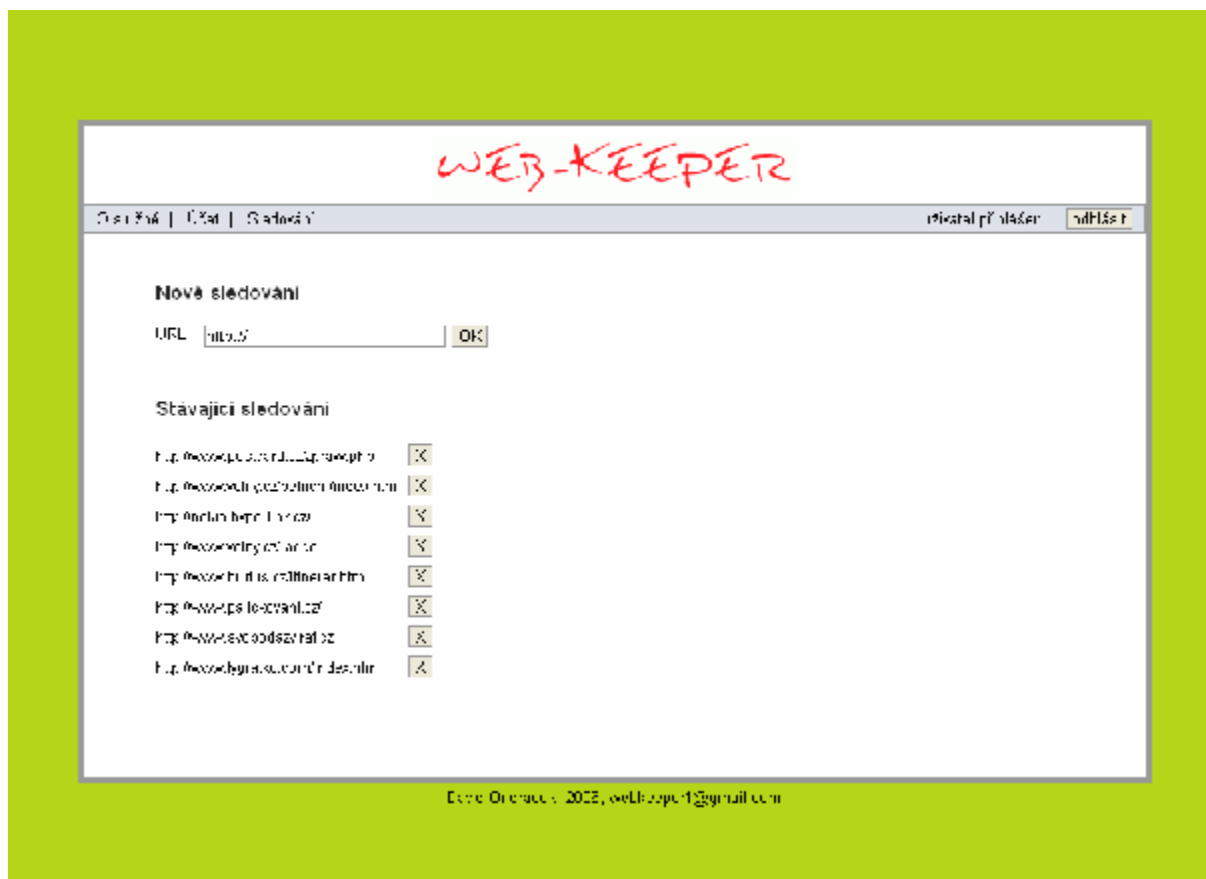
## 5.7 Kontrolní kódy

Obrázky s kontrolními kódy jsou použity jako prevence proti spamovacím robotům. Jejich zavedením je v podstatě možno odlišit, zda formulář s vloženými údaji odesílá člověk či robot. V případě jejich absence by mohlo dojít k rozeslání nevyžádaných e-mailů. Kód na obrázku není přítomen v kódu stránky, lze jej tedy zjistit pouze vizuálně (člověkem) nebo použít určitých technik pro zpracování obrazu. Spamovací roboti však těchto technik zatím nevyužívají nebo nejsou příliš dokonalé. Obrázky jsou navíc většinou vytvořeny tak, aby automatické rozpoznávání bylo co možná nejobtížnější, ale zároveň, aby byly dobře čitelné člověkem. Nevýhodou této techniky je nižší akceptace uživateli a také znemožnění odeslání formuláře pro uživatele, kteří mají v prohlížeči zakázáno zobrazování obrázků.

V systému používám 50 různých obrázků. Při zobrazení stránky se náhodně vybere jeden z nich. Jeho pořadové číslo se odešle po odeslání formuláře. Následně se z databáze podle pořadového čísla zjistí kód (tabulka *codes*) a porovná se s kódem, který do formuláře vložil uživatel.

## 6 Nasazení, testování

Implementovaný systém jsem nazval Web-Keeper a lze jej otestovat na adrese <http://www.web-keeper.zde.cz>. Tato doména je momentálně přeměrována na můj osobní počítač, kde služba běží. Může se tedy stát, že nebude k dispozici vždy. Skript robota se spouští jednou denně.



Obr. 3 – Náhled implementovaného systému

## 7 Funkčnost a omezení vytvořeného programu

Implementovaný systém je pouze základem a jako takový má následující omezení:

1. Systém sleduje pouze zaregistrované URL, nesleduje žádné další odkazy, které se na daném umístění vyskytují.
2. V této verzi také nedokáže pracovat s rámy (tagy frame, iframe).
3. Systém není schopen sledovat přesměrování, jakýmkoliv způsobem zařízené (manuální, meta refresh, javascript, serverovou hlavičkou).
4. Rovněž není možné sledovat zabezpečené stránky přes protokol https.

## 8 Rozšíření

V dalším by bylo vhodné systém rozšířit o další funkce.

1. Sledování přesměrování a obsahu rámců.
2. Sledování odkazů – uživatel by si mohl zvolit, do jaké úrovně chce odkazy sledovat (počínaje zaregistrovanou URL).
3. Způsob informování o změně – uživatel by si mohl vybrat, jakým způsobem chce o změně webu informovat (e-mail, SMS, RSS, ICQ apod.).
4. Uživatel by měl mít možnost, nastavit si interval kontroly zaregistrovaných stránek. Samozřejmě v nějakém zdola ohraničeném intervalu, aby nedocházelo k přetěžování serverů.
5. Některé uživatele by mohly zajímat nejen změny obsahu, ale také změny obrázků, designu, odkazů apod. Systém by měl umět tyto změny také detekovat.
6. Pro uživatele by také mohlo být zajímavé, kdyby systém uměl e-mailem místo upozornění posílat rovnou obsah webu se zvýrazněnými změnami.
7. Užitečnou vlastností systému by také mohla být možnost specifikovat pouze část dokumentu, kterou chce uživatel sledovat. K tomu by se daly vhodně využít regulární výrazy.
8. V některých dokumentech se pravidelně mění pouze informace, které uživatele pravděpodobně nebudou zajímat. Může se jednat například o nějaké datum či časový údaj. Autor webu by měl mít možnost uzavřít tuto část dokumentu do nějakých speciálních tagů, které by označovaly místo, které systém nemá kontrolovat. Také by měl mít možnost robotovi přístup zcela zakázat.

## 9 Závěr

Cílem semestrální projektu bylo mimo jiné seznámení se současnými technologiemi publikování na internetu. Ty byly krátce představeny na začátku práce. Podrobný popis syntaxe jednotlivých programovacích jazyků nebyl její náplní. Pro bližší seznámení s těmito technologiemi si lze prostudovat dokumenty na něž se odkazují v sekci *Literatura*.

Navržený a implementovaný systém je pouze základem pro další pokračování v rámci navazující diplomové práce. V budoucnosti by tedy měl být doplněn minimálně o rozšíření navržená v kapitole 8. Domnívám se však, že již v této fázi je pro detekci změny obsahu webu dobře použitelný.

Službu lze otestovat na adrese uvedené v kapitole 6.

## Literatura

- [1] W3C, *HyperText Markup Language (HTML)* [online]  
<<http://www.w3.org/MarkUp/>> [cit. 2006-12-28]
- [2] W3C, *Extensible Markup Language (XML)* [online]  
<<http://www.w3.org/XML/>> [cit. 2006-12-28]
- [3] MySQL AB, *MySQL 5.0 Reference Manual* [online]  
<<http://dev.mysql.com/doc/refman/5.0/en/>> [cit. 2006-12-28]
- [4] Rivest, R., *The MD5 Message-Digest Algorithm* [online]  
<<http://www.ietf.org/rfc/rfc1321.txt>> [cit. 2006-12-28]
- [5] W3C, *Resource Description Framework* [online]  
<<http://www.w3.org/RDF/>> [cit. 2006-12-28]
- [6] Jiří Kosek, *Kurz HTML* [online]  
<<http://www.kosek.cz/clanky/html/index.html>> [cit. 2006-12-28]
- [7] Jiří Kosek, *XML pro každého* [online]  
<<http://www.kosek.cz/xml/xmlprokazdeho.pdf>> [cit. 2006-12-28]
- [8] Dušan Janovský, *Jak psát web* [online]  
<<http://www.jakpsatweb.cz/index.html>> [cit. 2006-12-28]
- [9] *Interval.cz* [online]  
<<http://interval.cz/webdesign/>> [cit. 2006-12-28]
- [10] Refsnes Data, *PHP Tutorial* [online]  
<<http://www.w3schools.com/php/default.asp>> [cit. 2006-12-28]
- [11] *PHP: Hypertext Preprocessor* [online]  
<<http://www.php.net/>> [cit. 2006-12-28]
- [12] Wikipedia, *HTML* [online]  
<<http://en.wikipedia.org/wiki/HTML>> [cit. 2006-12-28]
- [13] Wikipedia, *XML* [online]  
<<http://en.wikipedia.org/wiki/XML>> [cit. 2006-12-28]

- [14] Wikipedia, *PHP* [online]  
<<http://en.wikipedia.org/wiki/PHP>> [cit. 2006-12-28]
  
- [15] Wikipedia, *MySQL* [online]  
<<http://en.wikipedia.org/wiki/Mysql>> [cit. 2006-12-28]
  
- [16] Wikipedia, *MD5* [online]  
<<http://en.wikipedia.org/wiki/MD5>> [cit. 2006-12-28]



# Příloha A

## SQL skript pro vytvoření tabulek

```
CREATE TABLE accounts (  
  id int(11) NOT NULL auto_increment,  
  email varchar(255) NOT NULL default "",  
  psw varchar(32) NOT NULL default "",  
  PRIMARY KEY (id)  
) TYPE=MyISAM;
```

```
CREATE TABLE refs (  
  account_id int(11) NOT NULL default '0',  
  url_id int(11) NOT NULL default '0',  
  PRIMARY KEY (account_id,url_id)  
) TYPE=MyISAM;
```

```
CREATE TABLE reg (  
  email varchar(255) NOT NULL default "",  
  code varchar(32) NOT NULL default "",  
  psw varchar(32) NOT NULL default "",  
  PRIMARY KEY (email)  
) TYPE=MyISAM;
```

```
CREATE TABLE urls (  
  id int(11) NOT NULL auto_increment,  
  url varchar(255) NOT NULL default "",  
  md5sum varchar(32) NOT NULL default "",  
  PRIMARY KEY (id)  
) TYPE=MyISAM;
```

```
CREATE TABLE codes (  
  image_number smallint(11) NOT NULL default '0',  
  code varchar(5) NOT NULL default "",  
  PRIMARY KEY (image_number)  
) TYPE=MyISAM;
```

```
INSERT INTO codes VALUES (1, 'QXXNP'); INSERT INTO codes VALUES (2, 'JYLSX');  
INSERT INTO codes VALUES (3, 'XZYHA'); INSERT INTO codes VALUES (4, 'CYVFF');  
INSERT INTO codes VALUES (5, 'YAEED'); INSERT INTO codes VALUES (6, 'XJRHS');  
INSERT INTO codes VALUES (7, 'JSPDC'); INSERT INTO codes VALUES (8, 'JNMFY');
```

*INSERT INTO codes VALUES (9, 'IJMOD');*  
*INSERT INTO codes VALUES (10, 'LEESN');*  
*INSERT INTO codes VALUES (11, 'SYNMG');*  
*INSERT INTO codes VALUES (12, 'XQLAS');*  
*INSERT INTO codes VALUES (13, 'VKNRK');*  
*INSERT INTO codes VALUES (14, 'WDXWH');*  
*INSERT INTO codes VALUES (15, 'BPGKH');*  
*INSERT INTO codes VALUES (16, 'FXLKW');*  
*INSERT INTO codes VALUES (17, 'BTXGG');*  
*INSERT INTO codes VALUES (18, 'PBSWB');*  
*INSERT INTO codes VALUES (19, 'XQXVZ');*  
*INSERT INTO codes VALUES (20, 'CMJJE');*  
*INSERT INTO codes VALUES (21, 'CWVQW');*  
*INSERT INTO codes VALUES (22, 'FCFPJ');*  
*INSERT INTO codes VALUES (23, 'MBIMI');*  
*INSERT INTO codes VALUES (24, 'UGBSX');*  
*INSERT INTO codes VALUES (25, 'YAQPL');*  
*INSERT INTO codes VALUES (26, 'DIIUK');*  
*INSERT INTO codes VALUES (27, 'UJRZO');*  
*INSERT INTO codes VALUES (28, 'UWJJJ');*  
*INSERT INTO codes VALUES (29, 'WMIJI');*  
*INSERT INTO codes VALUES (30, 'SSKBB');*  
*INSERT INTO codes VALUES (31, 'KNMBE');*  
*INSERT INTO codes VALUES (32, 'VKFZP');*  
*INSERT INTO codes VALUES (33, 'UNJAP');*  
*INSERT INTO codes VALUES (34, 'VXDKM');*  
*INSERT INTO codes VALUES (35, 'OZCBZ');*  
*INSERT INTO codes VALUES (36, 'SBYZE');*  
*INSERT INTO codes VALUES (37, 'MBHRL');*  
*INSERT INTO codes VALUES (38, 'DDRKO');*  
*INSERT INTO codes VALUES (39, 'SEBWA');*  
*INSERT INTO codes VALUES (40, 'FPEGG');*  
*INSERT INTO codes VALUES (41, 'ENQNR');*  
*INSERT INTO codes VALUES (42, 'HUVQP');*  
*INSERT INTO codes VALUES (43, 'XWLKF');*  
*INSERT INTO codes VALUES (44, 'INDMO');*  
*INSERT INTO codes VALUES (45, 'WDMMC');*  
*INSERT INTO codes VALUES (46, 'UGHEW');*  
*INSERT INTO codes VALUES (47, 'SQEMO');*  
*INSERT INTO codes VALUES (48, 'GFZWT');*  
*INSERT INTO codes VALUES (49, 'TBWMM');*  
*INSERT INTO codes VALUES (50, 'EEHC');*

## Příloha B

### Pseudokód hashovací funkce MD5 [16]

*//Note: All variables are unsigned 32 bits and wrap modulo  $2^{32}$  when calculating*

*//Define r as the following*

*var int[64] r, k*

*r[0..15] := {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22}*

*r[16..31] := {5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20}*

*r[32..47] := {4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23}*

*r[48..63] := {6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21}*

*//Use binary integer part of the sines of integers as constants:*

*for i from 0 to 63*

*k[i] := floor(abs(sin(i + 1)) ×  $2^{32}$ )*

*//Initialize variables:*

*var int h0 := 0x67452301*

*var int h1 := 0xEFCDAB89*

*var int h2 := 0x98BADCFE*

*var int h3 := 0x10325476*

*//Pre-processing:*

*append "1" bit to message*

*append "0" bits until message length in bits  $\equiv 448 \pmod{512}$*

*append bit length of message as 64-bit little-endian integer to message*

*//Process the message in successive 512-bit chunks:*

*for each 512-bit chunk of message*

*break chunk into sixteen 32-bit little-endian words  $w(i)$ ,  $0 \leq i \leq 15$*

*//Initialize hash value for this chunk:*

*var int a := h0*

*var int b := h1*

*var int c := h2*

*var int d := h3*

```

//Main loop:
for i from 0 to 63
  if  $0 \leq i \leq 15$  then
     $f := (b \text{ and } c) \text{ or } ((\text{not } b) \text{ and } d)$ 
     $g := i$ 
  else if  $16 \leq i \leq 31$ 
     $f := (d \text{ and } b) \text{ or } ((\text{not } d) \text{ and } c)$ 
     $g := (5 \times i + 1) \text{ mod } 16$ 
  else if  $32 \leq i \leq 47$ 
     $f := b \text{ xor } c \text{ xor } d$ 
     $g := (3 \times i + 5) \text{ mod } 16$ 
  else if  $48 \leq i \leq 63$ 
     $f := c \text{ xor } (b \text{ or } (\text{not } d))$ 
     $g := (7 \times i) \text{ mod } 16$ 

  temp := d
  d := c
  c := b
   $b := ((a + f + k[i] + w(g)) \text{ leftrotate } r[i]) + b$ 
  a := temp

//Add this chunk's hash to result so far:
h0 := h0 + a
h1 := h1 + b
h2 := h2 + c
h3 := h3 + d

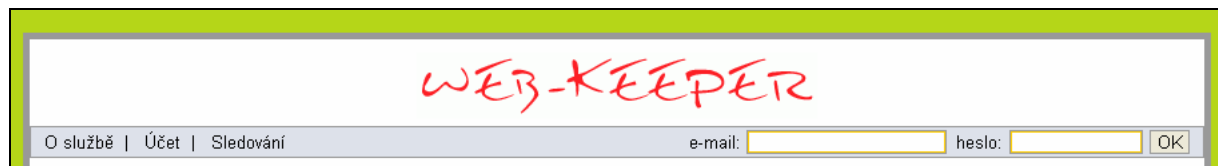
var int digest := h0 append h1 append h2 append h3 //(expressed as little-endian)

```

## Příloha C

### Uživatelská příručka

#### Menu, přihlášení



Menu aplikace je tvořeno třemi položkami – *O službě*, *Účet*, *Sledování*.

Sekce *O službě* obsahuje obecné informace o aplikaci.

Po přihlášení sekce *Účet* nabízí možnost změny hesla, případně zrušení účtu. Jinak je zde možné se zaregistrovat nebo si nechat poslat zapomenuté heslo.

Sekce *Sledování* umožňuje upravovat seznam sledovaných stránek. Je dostupná pouze po přihlášení.

Do systému se přihlašuje zadáním e-mailové adresy a hesla do formuláře vpravo nahoře.


#### Registrace

**Registrace**

e-mail:

heslo:

heslo znovu:



potvrzovací kód:

Jste zaregistrován(a), ale zapomněl(a) jste heslo? Můžete si jej nechat poslat [zde](#).

Pro registraci je nutné vyplnit formulář v sekci *Účet*. Kromě e-mailu a hesla je třeba vyplnit také pole *heslo znovu* (slouží pro eliminaci překlepů) a opsat potvrzovací kód z obrázku (opatření proti spamovacím robotům). Na uvedenou adresu bude odeslán e-mail s informacemi pro dokončení registrace.

## Odeslání zapomenutého hesla

**Zaslání hesla**

Zapomněl(a) jste heslo? Můžete si jej nechat poslat zde.

e-mail:



potvrzovací kód:

V případě zapomenutí hesla je možné si jej nechat poslat e-mailem. Stačí vyplnit formulář v sekci *Účet*. Je nutné zadat správnou e-mailovou adresu a opsat potvrzovací kód z obrázku.

## Změna hesla, zrušení účtu

**Změna hesla**

heslo:

nové heslo:

Změnit heslo pro přístup do systému je možné po přihlášení v sekci *Účet*. Kromě nového hesla je třeba zadat i heslo stávající.

V sekci *Účet* je po přihlášení možné také zrušit účet.

## Nastavení sledování

**Nové sledování**

URL:

**Stávající sledování**

<a href="http://www.postcard.cz/zpravy.php">http://www.postcard.cz/zpravy.php</a>	<input type="checkbox"/>
<a href="http://www.volny.cz/petricm/index.htm">http://www.volny.cz/petricm/index.htm</a>	<input type="checkbox"/>
<a href="http://nokia.hyperlink.cz/">http://nokia.hyperlink.cz/</a>	<input type="checkbox"/>
<a href="http://www.volny.cz/radce">http://www.volny.cz/radce</a>	<input type="checkbox"/>
<a href="http://www.brutus.cz/tinerar.htm">http://www.brutus.cz/tinerar.htm</a>	<input type="checkbox"/>
<a href="http://www.palickovani.cz/">http://www.palickovani.cz/</a>	<input type="checkbox"/>
<a href="http://www.svobodazvirat.cz">http://www.svobodazvirat.cz</a>	<input type="checkbox"/>
<a href="http://www.tygratko.com/index.htm">http://www.tygratko.com/index.htm</a>	<input type="checkbox"/>

V sekci *Sledování* je možné přidat nové sledování stránek. Stačí zadat jejich URL do formuláře. Také je zde seznam již sledovaných stránek. Pro vyjmutí URL ze seznamu slouží křížek u dané položky vpravo.