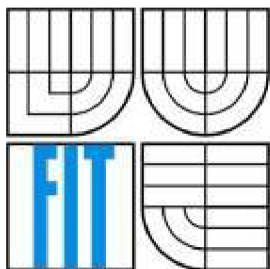




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# WEBOVÁ APLIKACE ZOBRAZUJÍCÍ STATISTICKÉ ÚDAJE O WEBOVÝCH STRÁNKÁCH

WEB APPLICATION FOR STATISTICAL ANALYSIS OF WEB PAGES

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

PETR HOCHMAL

VEDOUCÍ PRÁCE  
SUPERVISOR  
BRNO 2007

ING. JIŘÍ TECHET

[kopie originálu zadání bakalářské práce]

Zadání bakalářské práce/5271/2006/xhochm01

**Vysoké učení technické v Brně - Fakulta informačních technologií**  
Ústav informačních systémů Akademický rok 2006/2007

## Zadání bakalářské práce

Řešitel: **Hochmal Petr**  
Obor: Informační technologie  
Téma: **Webová aplikace zobrazující statistické údaje o webových stránkách**  
Kategorie: Web

Pokyny:

1. Seznamte se s jazykem XHTML, kaskádovými styly CSS, databázovým serverem MySQL, jazykem PHP a jeho SQL dialektem pro přístup k databázovému serveru, a dále potom s formátem RSS.
2. Zamyslete se nad údaji, které chcete u webových stránek sledovat (například velikost dokumentu, počet elementů, počet obrázků, počet odkazů a podobně).
3. Navrhněte aplikaci, která umožní monitorování zadaných webových stránek v čase, zaznamenávání a statistické vyhodnocování sledovaných údajů. Systém musí umožňovat sledování změn na stránkách pomocí RSS kanálů, import a export sledovaných dat ze souboru XML.
4. Implementujte navržený systém s použitím výše uvedených nástrojů.
5. Diskutujte přednosti a nedostatky vašeho řešení a nastiňte další možný vývoj vašeho projektu.

Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací, Softpress 2003, ISBN 8086497607.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdává v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Techet Jiří, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
602 00 Brno, Požetěchova 2  
L.S.



doc. Ing. Jaroslav Zendulka, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA  
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Petr Hochmal**  
Id studenta: 84372  
Bytem: Soběčice 139, 508 01 Hořice  
Narozen: 30. 04. 1985, Jičín  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1  
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Webová aplikace zobrazující statistické údaje o webových stránkách  
Vedoucí/školicel VŠKP: Tchet Jiří, Ing.  
Ústav: Ústav informačních systémů  
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1  
elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění

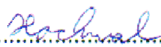
1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

  
.....  
Autor

## **Abstrakt**

Práce se zabývá problematikou sledování změn obsahu webových stránek v čase. Cílem bylo vytvořit webovou aplikaci, umožňující sledování uživatelem definovaného obsahu stránek a jeho záznam v čase, s možností statistického vyhodnocení. Tato práce zprvu stručně a obecně popisuje webové aplikace a technologie, které byly využity k této práci, a později se zabývá analýzou, návrhem a vlastním řešením tvorby aplikace.

## **Klíčová slova**

statistická analýza webových stránek, webová aplikace, XHTML, PHP, MySQL, JavaScript, XML, XPath, CSS

## **Abstract**

This thesis is focused on web page changes monitoring. The main objective of the work was to create a web application that will monitor user defined content of a web page and save it with the possibility of statistical analysis of the saved values. The thesis describes web applications and technologies that were used in this application in general. It also deals with analysis, design and implementation of the web application.

## **Keywords**

statistical analysis of web pages, web application, XHTML, PHP, MySQL, JavaScript, XML, XPath, CSS

## **Citace**

Petr Hochmal: Webová aplikace zobrazující statistické údaje o webových stránkách, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Webová aplikace zobrazující statistické údaje o webových stránkách

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Techeta.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jméno Příjmení  
Datum

## Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Jiřímu Techetovi za jeho cenné připomínky a trpělivost, se kterou práci četl.

© Petr Hochmal, 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..*

# Obsah

Obsah.....	1
1 Úvod.....	3
2 Použité technologie .....	4
2.1 PHP.....	4
2.2 MySQL .....	4
2.3 JavaScript.....	4
2.4 DOM.....	5
2.5 XPath .....	5
3 Webové aplikace.....	6
3.1 Definice.....	6
3.2 Struktura.....	6
3.3 Programování webových aplikací.....	6
4 Analýza a návrh řešení .....	7
4.1 Požadavky na funkčnost aplikace.....	7
4.2 Vyhodnocování obsahu stránek.....	7
4.2.1 Systém vyhodnocování obsahu stránek v čase.....	7
4.2.2 Způsob definice sledování dat.....	8
4.2.3 Nakládání se získanými daty.....	10
4.3 Uživatelské účty .....	11
4.4 Konceptuální model.....	12
4.5 Funkčnost aplikace .....	13
4.6 Uživatelské rozhraní .....	15
4.6.1 Základní rozvržení stránky.....	15
4.6.2 Logická struktura stránek.....	15
4.6.3 Obsah stránek .....	16
5 Implementace aplikace.....	18
5.1 Tvorba databáze.....	18
5.2 Definice hodnoty parametru sledování .....	19
5.3 Systém vyhodnocování obsahu stránek .....	19
5.4 Export a import sledovaných dat .....	21
5.5 Webové uživatelské rozhraní .....	22
5.5.1 Celkový koncept webu.....	22
5.5.2 Uživatelsky přívětivé nastavení aplikace.....	23
5.5.3 Grafové znázornění sledovaných dat.....	23

5.6	RSS kanál.....	24
5.7	Možnosti využití aplikace .....	24
6	Závěr .....	25
	Literatura.....	26
	Seznam příloh .....	27
	Příloha 1 .....	28
	Příloha 2.....	29
	Příloha 3.....	30



# 1 Úvod

World Wide Web, jedna ze služeb sítě internet, bezesporu plní funkci celosvětového informačního média. Velké množství obsahu webu je obsaženo v dynamických webových stránkách a tyto informace se v průběhu času mění. A právě tyto změny mohou být pro někoho předmětem zájmu. Jedná se například o sledování ceny produktu v e-shopu nebo verze námi oblíbeného programu, jež se neustále a rychle vyvíjí. Nebo i jen zaznamenávání a vyhodnocování velikosti i struktury obsahu webové stránky může být v mnoha případech vhodné. Právě sledováním těchto změn se zabývá tato práce.

Výsledkem práce je webová aplikace, která umožňuje uživateli nastavovat stránky, jejichž obsah chce sledovat a definovat, které informace z obsahu stránek chce vyhodnocovat. Data vzniklá tímto vyhodnocením jsou uživateli zprostředkována formou statistik, textového výpisu i zobrazením do grafu. Data lze i exportovat a importovat ve formátu XML. Uživatel je také informován prostřednictvím RSS kanálu o změnách na stránkách.

V následující kapitole bude čtenář seznámen s několika hlavními pojmy, jež jsou ve vytvářené aplikaci používány. Třetí kapitola stručně popíše použité technologie a čtvrtá v rychlosti pojedná o problematice webových aplikací. V kapitole 5 Analýza a návrh řešení se budeme zabývat právě analýzou požadavků na aplikaci a jejím návrhem. V poslední kapitole bude převážně nastíněna implementace nejzajímavějších částí projektu.

## 2 Použité technologie

V následujících řádcích jsou popsány technologie použité k implementaci aplikace. Jedná se zejména o technologie určené pro vytváření a práci s dynamickými webovými stránkami a skladování dat, jež využívají.

### 2.1 PHP

PHP (z původního Personal Home Page, nyní spíše Hypertext Preprocessor) je skriptovací jazyk pracující na straně serveru. Využívá se pro generování dynamického obsahu www stránek. Umožňuje spolupracovat s mnoha databázovými servery a obsahuje mnohé užitečné moduly pro vytváření obsahu webových stránek.

Samotné PHP skripty se zapisují přímo do HTML dokumentu. PHP interpret pak pracuje tak, že HTML příkazy vkládá přímo do výsledného HTML dokumentu a narazí-li na PHP skript, provede ho a výsledek vloží do HTML dokumentu.

### 2.2 MySQL

MySQL je multiplatformní databáze a jako taková je s oblibou používána například ve spojitosti s webovým serverem Apache a skriptovacím jazykem PHP. Komunikace s ní probíhá, jak už název napovídá, prostřednictvím jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

### 2.3 JavaScript

JavaScript je skriptovací jazyk, který se využívá k dodání dynamičnosti webových stránek. Narozdíl od PHP se jedná o tzv. client based skriptovací jazyk, tedy skript prováděný na straně klienta. Výhoda spočívá v rychlosti provádění změn na stránkách, jelikož není nutné přenášet data od a ke klientovi. Na druhou stranu mezi jeho drobné nevýhody patří podpora v prohlížečích, kde proti sobě stojí dobře zdokumentovaný JavaScript a částečně podobný JScript od společnosti Microsoft. Lépe řečeno, problémem je vzájemná kompatibilita těchto dvou jazyků. K provedení jedné činnosti je v některých případech nutné psát rozdílný kód pro různé prohlížeče.

JavaScript umožňuje pracovat přímo s obsahem HTML dokumentu prostřednictvím DOM. Je tak možné přistupovat k jednotlivým prvkům stránky a provádět v nich změny.

## 2.4 DOM

Document Object Model, neboli objektový model dokumentu je API (Application Programming Interface – aplikační programovací rozhraní) pro XML a HTML dokumenty. V podstatě propojuje webové stránky nebo XML dokumenty s programovacími jazyky.

Každý prvek stránky je reprezentován objektem, majícím vlastnosti, metody a události. Celý dokument tvoří jakousi stromovou strukturu, tedy objekty mají předky, následníky a sousedy. Takto lze tedy celým dokumentem procházet.

Například právě implementace DOM je částečně rozdílná mezi JavaScriptem a JScriptem.

## 2.5 XPath

Jedná se o jazyk, nebo spíše výraz, pro adresování částí XML (tedy i HTML) dokumentu. Vyhodnocením XPath výrazu získáme jeden nebo více uzlů splňujících podmínky definované ve výrazu. Můžeme tak tedy přesně adresovat jednotlivé části HTML dokumentu, přistupovat k nim a dále je zpracovávat.

## 3 Webové aplikace

V následujících řádcích se letmo seznámíme s problematikou tvorby webových aplikací.

### 3.1 Definice

Webová aplikace je taková aplikace, jejíž data jsou zpřístupňována prostřednictvím webového rozhraní neboli internetovým prohlížečem zpravidla skrze internet.

Jejich největší výhodou spočívá v přístupnosti, která je závislá pouze na dostupnosti internetového připojení. Jinými slovy lze říci, že k přístupu k využívané aplikaci nejsme omezeni na konkrétní počítač, ale můžeme k ní přistupovat odkudkoliv. Webové aplikace nejsou pro uživatele nijak platformě omezeny. Veškeré podstatné zpracování je prováděno na serveru, kde jsou data ukládána.

### 3.2 Struktura

Webové aplikace jsou obvykle koncipovány jako třívrstvé. V té nejběžnější formě je webový prohlížeč první vrstvou, nástroje pro dynamické generování stránek jsou vrstvou druhou a třetí vrstvou reprezentuje databáze.

Webový prohlížeč (např. Firefox) posílá požadavky druhé vrstvě (např. Apache s PHP), ta je zpracovává mj. prostřednictvím dotazů do databáze (např. MySQL) a generováním dynamického obsahu stránek, ke kterému klient opět přistupuje.

### 3.3 Programování webových aplikací

Mezi jazyky, které se obvykle využívají pro tvorbu webových aplikací, patří PHP, Perl nebo Python, jež komunikují s webovým serverem (např. Apache) pomocí CGI (Common Gateway Interface).

Další podrobnosti lze nalézt např. v [5].

## 4 Analýza a návrh řešení

Tato kapitola je věnována rozboru návrhu tvorby aplikace. Je popsána analýza požadovaných funkcí programu a i dalších prvků, jež budeme tvořit vytvořit.

### 4.1 Požadavky na funkčnost aplikace

Byly definovány následující požadavky:

- Webová aplikace umožní přidávat a ubírat stránky, které chcete sledovat.
- K jednotlivým stránkám musí být umožněno přidání a odebrání sledovaných údajů.
- Pokud je stránka sledována, tak např. každou hodinu provede aplikace vyhodnocení sledovaných údajů na stránce a tato data uloží.
- Zobrazení sledovaných dat formou tabulky nebo grafu.
- Aplikace by měla umožnit sledovat např. velikost stránky, počet tabulek, počet řádků v tabulkách, počet odkazů, počet obrázků, atd. Dále by měla umožnit sledovat počet uživatelem definovaných elementů, případně atributů, které se v dokumentu vyskytují.
- Možnost importu a exportu sledovaných dat ve formátu XML.
- Informování o změnách prostřednictvím RSS.

### 4.2 Vyhodnocování obsahu stránek

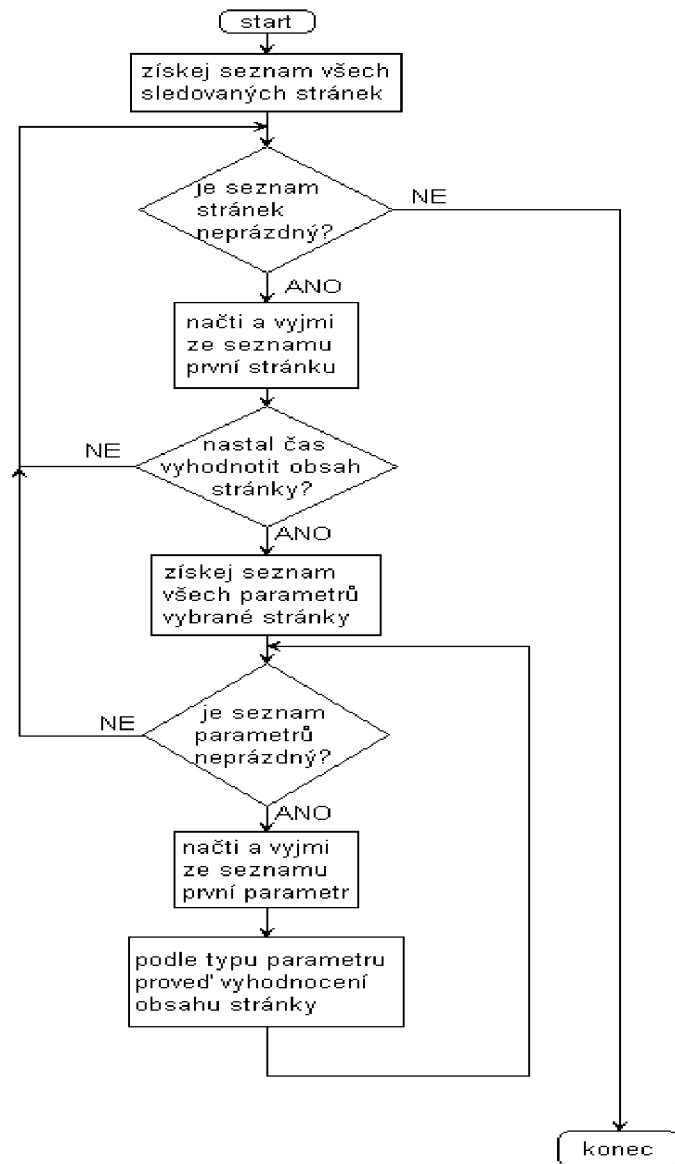
Máme tedy skupinu bodů, které je nutno splnit. Zprvu se bude třeba zaměřit na základní kámen této webové aplikace, a to na samotný problém sledování obsahu stránek. K tomu patří i způsob definování stránky určené ke sledování a toho, co chceme na stránce sledovat.

#### 4.2.1 Systém vyhodnocování obsahu stránek v čase

Jelikož chceme, aby se opakovaně v čase vyhodnocoval obsah stránek, je nutné zajistit pravidelné stahování obsahu těchto stránek. Protože pracujeme na úrovni PHP skriptů, nemůžeme akci vyhodnocování stránek opakovat uvnitř kódu. Jako vyhovující způsob můžeme označit volání PHP skriptu, jenž provádí vyhodnocování obsahu stránek, pomocí externí aplikace. Ta bude zajišťovat periodické spouštění skriptu ve vhodně zvoleném intervalu. Zároveň nám tím odpadne problém s dlouhou prodlevou při vyhodnocování stránek, jež by mohla nastat, a to tím, že nový cyklus vyhodnocování bude probíhat nezávisle na cyklu předchozím.

Algoritmus provádějící vyhodnocování obsahu stránek by měl pracovat na následujícím principu. Získáme seznam všech stránek, které jsou v aplikaci uživateli nastaveny. Budeme přistupovat jednotlivě k těmto stránkám a zjišťovat, zda v tomto cyklu sledování mají být

vyhodnoceny. Pokud ano, získá se seznam parametrů této konkrétní stránky a pomocí každého z těchto parametrů se vyhodnotí obsah stránky. Takto se zpracuje každá stránka, dokud nebude vyhodnocování stránek dokončeno. Přehledněji lze tento postup znázornit pomocí vývojového diagramu, jak vidíme na obrázku 1.



Obrázek 1 - Vyhodnocování obsahu stránek

## 4.2.2 Způsob definice sledování dat

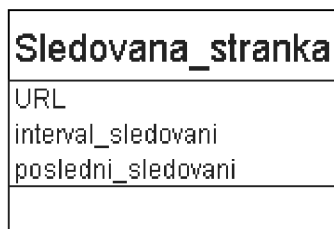
Zavedme si nejprve pro přehlednost následující pojmy:

- Sledovaná stránka – tedy stránka, jejíž obsah se vyhodnocuje
- Parametr sledování – definuje data, jež na stránce chceme sledovat
- Záznam sledování – jedná se o výsledek jednoho provedení vyhodnocení obsahu stránky

Máme tedy seznam pojmů, které budeme dále rozebírat. Sledovaná stránka a záznam sledování nám určují, jaká data budeme na stránce sledovat (a vyhodnocovat). Naopak záznam sledování bude sloužit k ukládání těchto dat.

U sledované stránky nás bude pravděpodobně nejvíce zajímat, o jakou stránku se jedná. Definujeme tedy její *URL*. Dalším a neméně podstatným údajem bude *interval*, ve kterém budeme přistupovat ke stránce a získávat z ní potřebná data. Vzhledem k rychlosti obměny obsahu dynamických webových stránek bude vhodné uvažovat o intervalu v řádu minut až hodin. Samozřejmě jeho hodnota by měla být uživatelsky nastavitelná.

Soubor těchto údajů tvoří entitu *Sledovana\_stranka*:



Obrázek 2 - Entita *Sledovana\_stranka*

Definování sledované stránky však k vyhodnocování obsahu stránek nepostačuje. K další specifikaci poslouží *parametr sledování*. Tím bychom měli definovat, jakým způsobem chceme vyhodnocování provádět, tedy zda zamýšlíme sledovat počet např. obrázků, či velikost celé stránky. Další zajímavou možností by bylo sledování textového obsahu stránky. Tím je myšlen obsah textového uzlu nějakého zvoleného elementu. A další, možná ještě zajímavější možností, by bylo vyhodnocení textového obsahu tohoto uzlu. Tím je myšleno získání číselné hodnoty z textu. Tedy například z následující části HTML dokumentu získat číslo 3136.

```
<span class="rvalue">3 136,- Kč</span>
```

Pro specifikaci parametru sledování budeme potřebovat znát dejme tomu *typ* parametru. Tedy jeden z následujících:

- počet elementů
- velikost dokumentu
- textový obsah elementu
- číselný obsah elementu

Další neméně důležitou vlastností je definice (a hlavně způsob definice) elementu, který v případě třech typů parametru sledování (počet elementů, textový obsah elementu, číselný obsah elementu) chceme vyhodnocovat. To ale v tuto chvíli mějme za nepodstatné a spokojme se s tím, že parametr sledování bude definován jakýmsi výrazem, nazvěme ho **hodnota** parametru, zajišťujícím jednoznačnost výběru jednoho nebo více elementů na stránce.

Nyní se vraťme k případu, kdy typ parametru sledování bude „číselný obsah elementu“ nebo „textový obsah elementu“. Zpravidla nám nebude vyhovovat sledovat celý obsah textového uzlu, a tak je nutné tento text „ořezat“, abychom získali pouze tu část, která nás zajímá. Pro tento účel bude postačovat definovat počet znaků před a za sledovanou částí textu. Nazvěme si například tuto vlastnost parametru sledování jako **zpracování**.

Jak je z výše zmíněného příkladu zřejmé, textový uzel elementu neobsahuje pro nás vhodně formátovanou číselnou hodnotu. A tak v případě, že chceme sledovat číselný obsah, budeme muset tento řetězec náležitě upravit. Pomocí vhodně zvoleného **zpracování** se dostaneme k řetězci „3 136,-“. Z tohoto řetězce bude zapotřebí odstranit mezery, případně i jiné bílé znaky, a vhodně přeformátovat „“ a „-“ na „“ a „0“. Získáme tak desetinné číslo „3136.0“, se kterým se již dají vytvářet statistiky atp.

Jako poslední vlastnost parametru sledování je zvolen **popis**, který by uživateli později napovídal, co konkrétní parametr sledování zajišťuje.

Seskupením těchto poznatků vznikne následující entita *Parametr\_sledovani*:

Parametr_sledovani
typ
hodnota
zpracovani
popis

Obrázek 3 - Entita *Parametr\_sledovani*

### 4.2.3 Nakládání se získanými daty

Poslední potřebnou částí pro úspěšné dokončení vyhodnocení obsahu stránky je uložení vzniklých dat. Tedy vytvoříme **záznam sledování**. Tím je myšleno přidání časového razítka (**čas**) vytvořeným datům (**hodnotě**).



Tím nám vzniká opět entita, tentokrát s názvem *Zaznam sledovani*:

Zaznam sledovani
cas
hodnota

Obrázek 4 - Entita *Zaznam sledovani*

### 4.3 Uživatelské účty

Aplikace, kterou vytváříme, je určena pro víceuživatelský přístup, je tak třeba definovat *uživatele* – uživatelský účet. Ten bude jednoznačně definován *loginem*. Tímto *loginem* se určí i „vlastnictví“ výše zmiňované sledované stránky. V aplikaci budeme rozlišovat více druhů uživatelů – běžný uživatel a administrátor (k jejich kompetencím a možnostem se dostaneme později). Tímto tedy určíme *práva* uživatele. V neposlední řadě je třeba myslet i na *heslo*, které společně s *loginem* bude umožňovat přihlášení do aplikace. Jako další můžeme vytvořit možnost uživatelský účet zablokovat, respektive aktivovat. K tomu poslouží vlastnost *stav*. Na posledním místě jsou doplňkové údaje o uživateli, a to např. *jméno* a *příjmení*.

Seskupením těchto vlastností uživatelského účtu vznikne entita *Uzivatel*:

Uzivatel
login
jmeno
prijmeni
heslo
prava
stav

Obrázek 5 - Entita *Uzivatel*

## 4.4 Konceptuální model

V předchozích odstavcích jsme analyzovali a částečně navrhli entity specifikující objekty v systému sledování stránek. Nyní je potřeba mezi entitami vytvořit vztahy a případně entity lépe specifikovat.

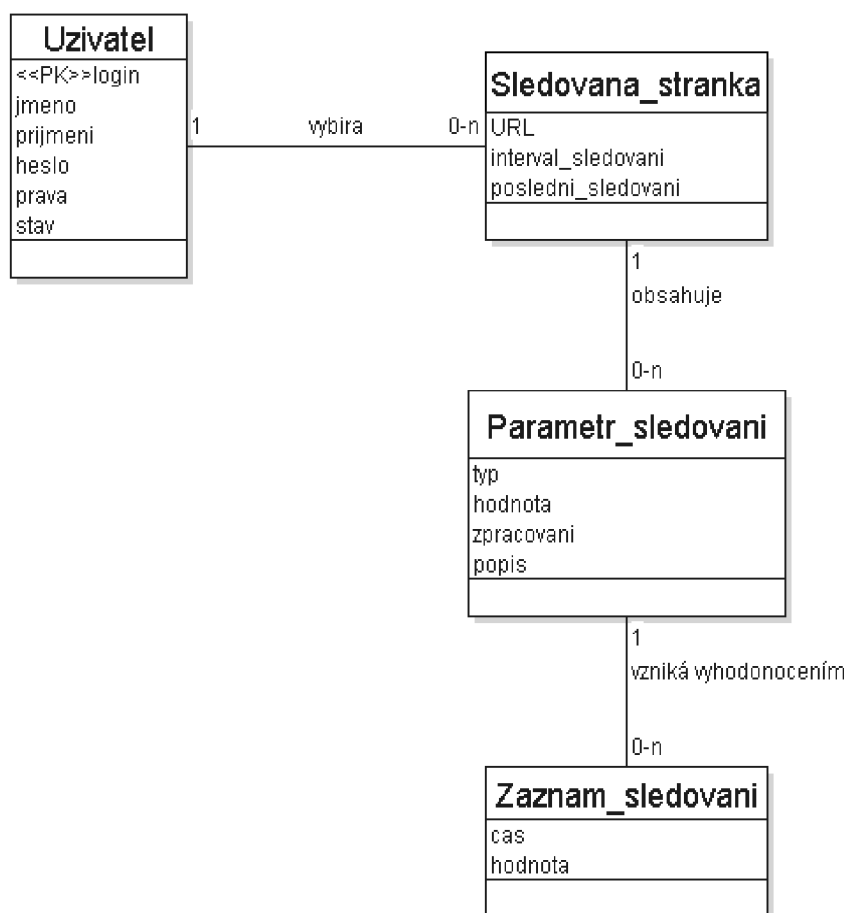
**Entita *Uzivatel*:** Uživatelský účet je jednoznačně identifikován svým *loginem*. Zvolíme tedy za primární klíč *login*. Jak jsme si již řekli výše, vlastnictví sledované stránky je vázáno na uživatelský účet. Vytvoříme tedy vztah mezi entitou *Uzivatel* a *Sledovana\_stranka*. Uživatel vybírá sledovanou stránku, pojmenujme tedy vztah *Vybira*, který bude mít kardinalitu 1:N.

**Entita *Sledovana\_stranka*:** Sledovaná stránka je tvořena svojí URL, intervalem sledování a časem posledního sledování, avšak ani u jednoho z atributů nemůžeme zajistit jeho jedinečnost. Vytvoříme tedy nový atribut, jenž bude nositelem identifikačního čísla – *id*, který bude zároveň tvořit primární klíč (PK) entity. *Sledovana\_stranka* je ve vztahu s *Uzivatelem* a dále s entitou *Parametr\_sledovani*. S tou tvoří vztah nazvaný „*Obsahuje*“ s kardinalitou 1:N.

**Entita *Parametr\_sledovani*:** Tato entita také zatím ztrácí na své jednoznačnosti, dodejme jí tedy také číselný identifikátor *id*, který bude tvořit PK. Kromě výše zmíněného vztahu s entitou *Sledovana\_stranka* vytvoříme i vztah s entitou *Zaznam\_sledovani*. Tento vztah bude mít kardinalitu také 1:N.

**Entita *Zaznam\_sledovani*:** Jedná se o tzv. slabou entitní množinu. Jako diskriminant (D) nám poslouží atribut *cas*. Tato entita tvoří vztah „*vytváří*“ s entitou *Parametr\_sledovani*.

Shrnutím těchto poznatků se dostáváme k tvorbě samotného ER diagramu (obrázek 6). S jeho pomocí budeme v další fázi tvorby aplikace vytvářet tabulky databáze. Tento datový model obsahuje vše, co v tuto chvíli můžeme pokládat za podstatné nebo vhodné.



Obrázek 6 - Konceptuální model db

## 4.5 Funkčnost aplikace

Z pohledu uživatele se jedná o funkčnost, která bude podmíněna právy jeho účtu. Budeme tedy rozlišovat mezi funkcemi běžného uživatelského účtu a funkcemi pro administrátora, který rozšiřuje možnosti běžného účtu.

Ze zadání vyplývá, že uživatel musí být schopen přidávat, ubírat a také i editovat sledované stránky. K těmto stránkám dále může přidávat, ubírat a editovat parametry sledování. Prostřednictvím těchto informací vzniklé záznamy sledování tvoří statistiky, ke kterým má uživatel také přístup. Dále se jedná o export a import záznamů sledování pro možnosti dalšího zpracování uživatelem. A jako poslední jde o informování uživatele o změnách v záznamech sledování pomocí RSS kanálů.

Pokud tyto informace shrneme, získáme následující možnosti běžného uživatele:

- přidání, odebrání, editace sledované stránky
- přidání, odebrání, editace parametru sledování
- sledování statistik
- import + export záznamů sledování
- čtení RSS kanálu o změnách

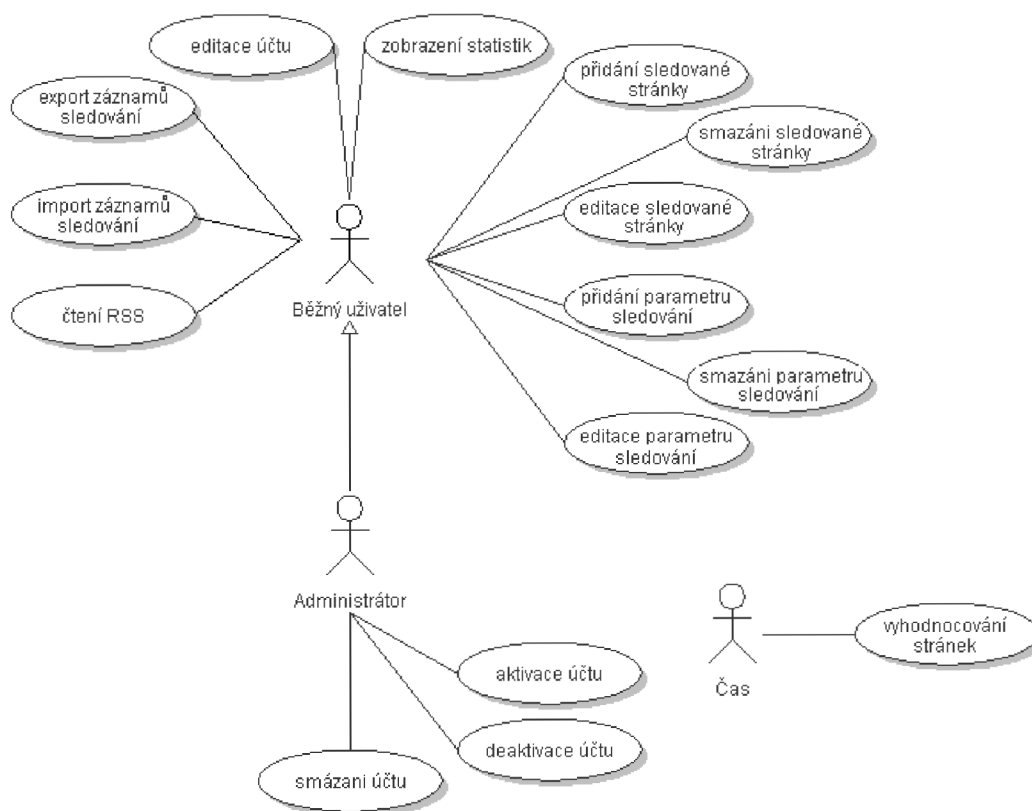
Administrátorský účet rozšiřuje seznam těchto možností o správu uživatelů. Mezi potřebné akce tedy patří povolování účtů a odstraňování uživatelů.

Možnosti administrátora:

- povolování a zakazování účtů
- smazání účtu

Jelikož aplikace musí být schopna vyhodnocovat data bez vlivu uživatele, vstupuje do hry čas. To znamená, že v určitých časových intervalech se vyhodnocuje obsah sledovaných stránek.

Vytvoříme-li z těchto poznatků diagram případů užití (tzv. use case diagram), získáme obrázek 7.



Obrázek 7 - Diagram případů užití

## 4.6 Uživatelské rozhraní

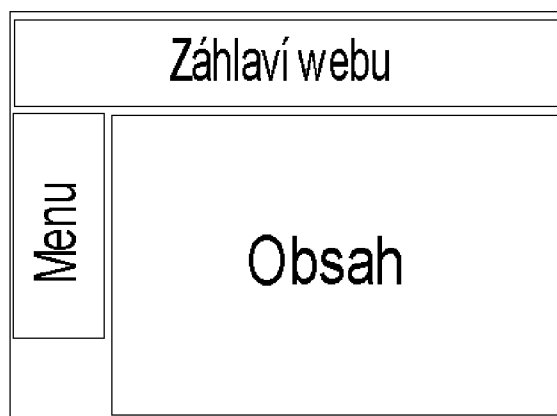
Uživatelé budou k této aplikaci přistupovat přes webové rozhraní, tedy prostřednictvím internetového prohlížeče. Je proto potřebné navrhnout toto uživatelské rozhraní, jež bude zprostředkovávat data uživateli.

### 4.6.1 Základní rozvržení stránky

V návrhu vzhledu stránky by měla být hlavně zohledněna přehlednost, aby se uživatel v obsahu neztrácel a věděl, co který prvek na stránce znamená. Stránku lze rozdělit do tématických celků pomocí rámců (což je v dnešní době pro většinu použití zavržená možnost) nebo pomocí css stylování. Pro použití v této aplikaci se rozhodneme pro druhou možnost. Stránku takto budeme dělit na následující oblasti:

- *záhlaví webu* – mělo by plnit pouze informační účel, tedy nést jméno webu
- *menu* – bude odkazovat na jednotlivé tématické části webu (nastavení, statistiky, import, export, atp.)
- *obsah* – v jeho těle bude zobrazen samotný obsah webu, na který bude odkazováno z menu

Tyto výše jmenované části budou rozloženy dle obr. 7. Tím docílíme přehlednosti pro uživatele, který se tak bude moci na webu snadno zorientovat.



Obrázek 8 - základní struktura webu

### 4.6.2 Logická struktura stránek

Logickou strukturou stránek rozumíme členění stránky na jednotlivé podstránky a jejich vzájemnou návaznost. Vzniká tím tedy jakási stromová struktura stránek, kterou si níže v textu rozebereme.

- hlavní stránka
  - nastavení sledovaných stránek
    - přidání sledované stránky
    - editace sledované stránky
    - nastavení parametrů sledování
      - přidání parametru sledování
      - editace parametru sledování
  - zobrazení statistik – sledované stránky
    - zobrazení statistik – parametry sledování
      - zobrazení statistik
  - export sledovaných dat – sledované stránky
    - export sledovaných dat – parametry sledování
  - import sledovaných dat
  - nastavení účtu
  - správa účtů

### 4.6.3 Obsah stránek

V předchozí podkapitole jsme si znázornili logickou strukturu stránek a nyní se rozhodneme, jaké informace tyto stránky budou zprostředkovávat. Níže uvedený seznam specifikuje přibližný obsah všech částí webu, které budou k dispozici.

- *hlavní stránka* – informace o aplikaci.
- *nastavení sledovaných stránek* – výčet všech uživatelských stránek s možností vyhledávání a přidání nové stránky a zobrazení parametrů sledování každé z nich. Dále každou sledovanou stránku bude možno upravit nebo smazat.
- *přidání/editace sledované stránky* – zadání/změna url a intervalu sledování stránky.
- *nastavení parametrů sledování* – výčet všech parametrů zvolené sledované stránky s možností přidání nového parametru a editace/smazání každého z nich.
- *přidání/editace parametru sledování* – zadání/změna typu, hodnoty, zpracování a popisu parametru.
- *zobrazení statistik - sledované stránky* – výčet sledovaných stránek s možností vyhledání stránky a zobrazení parametrů každé z nich.
- *zobrazení statistik - parametry sledování* – výčet všech parametrů zvolené stránky s možností zobrazení statistik u každého z nich.

- *zobrazení statistik* – zobrazení záznamů sledování vybraného parametru sledování v určitém časovém intervalu (den, týden, měsíc) od konkrétního času a zobrazení statistik (minimum, maximum, průměr) v daném období a grafového znázornění u všech typů parametrů sledování vyjma sledování části textu.
- *export sledovaných dat - sledované stránky* – výčet sledovaných stránek s možností vyhledávání a zobrazení parametrů každé z nich.
- *export sledovaných dat - parametry sledování* – výčet všech parametrů zvolené stránky s možností exportu každého z nich do souboru XML.
- *import sledovaných dat* – možnost importu sledovaných dat ze souboru XML.
- *nastavení účtu* – možnost změny jména, příjmení a hesla uživatele
- *správa účtů* – výpis všech uživatelů s možností aktivace/deaktivace a smazání účtu pro každého z nich.

# 5 Implementace aplikace

S ohledem na analýzu a návrh řešení aplikace se dostáváme do fáze vlastní realizace programu. Zprvu se budeme zabývat tvorbou databáze pro tuto aplikaci. Seznámíme se se způsobem adresace vnitřní struktury HTML dokumentu a možností získání této adresy. Poté vyřešíme realizaci hlavní části aplikace, kterou je vyhodnocování stránek. A v poslední části se budeme věnovat vytváření uživatelského rozhraní aplikace.

Řešení vlastní aplikace bude realizováno s ohledem na možnosti a omezení hostování webu na fakultním serveru *eva*. Obsah této kapitoly bude zaměřený spíše na, pro tuto aplikaci specifické, konstrukce, než na obecné kódy pro generování dynamických webových stránek.

## 5.1 Tvorba databáze

Jak ze zadání vyplývá, zvolenou databází pro tuto aplikaci bude MySQL. Z nabízených možností pro tabulky v této verzi MySQL je nejvhodnější zvolit použití MyISAM tabulky. Bohužel jelikož u tohoto typu tabulky nefungují cizí klíče, je nutné řešit integritu dat v programu.

Při tvorbě tabulek budeme vycházet z ER diagramu (obrázek 6), který znázorňuje konceptuální model databáze. Vytvoříme tedy 4 tabulky, jež budou vyhovovat návrhu databáze. Pro náš účel k názvům tabulek přidáme prefix „bc“ (od angl. bachelor – jedná se o bakalářskou práci), čímž je odlišíme od jiných tabulek v databázi.

Tvoříme následující tabulky:

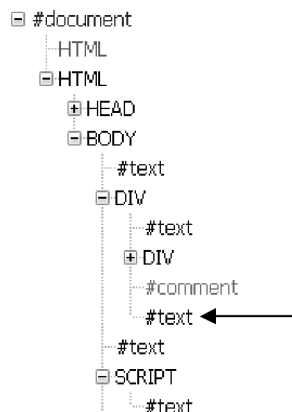
- *bcUzivatel* – tabulka je nezávislá na ostatních, sloupce budou odpovídat návrhu
- *bcSledovanaStranka* – ze vztahu s tabulkou *bcUzivatel* vyplývá přidání sloupce *uzivatel*, jenž bude identifikovat vlastníka tabulky
- *bcParametrSledovani* – přidáme sloupec *id\_stranky*, jenž bude určovat, ke které sledované stránce parametr náleží, jak vyplývá ze vztahu s tabulkou *bcSledovanaStranka*
- *bcZaznamSledovani* – v tabulce přidáme sloupec *id\_parametru*, který bude společně se sloupcem *cas* tvořit primární klíč

Podrobnosti o definici tabulek databáze viz příloha 1 (SQL dotazy pro vytvoření tabulek databáze).



## 5.2 Definice hodnoty parametru sledování

Jak jsme si již v kapitole 5.2.2 řekli, potřebujeme definovat, kterou část HTML dokumentu chceme sledovat. Jelikož obsah HTML (a i XML) dokumentů tvoří stromovou strukturu, kde elementy a text tvoří uzly (viz obrázek 9), lze poměrně snadno definovat cestu ke konkrétnímu uzlu.



Obrázek 9 - Příklad struktury HTML dokumentu

Naštěstí však již existuje mocný nástroj pro adresování prvků v XML dokumentech a tak nemusíme navrhovat vlastní definice cesty k prvku a implementovat kód pro jeho vyhledání v dokumentu. Tímto nástrojem je XPath, jenž je implementován v rámci DOM (Document Object Modeler) v PHP. XPath výraz (vyjádření cesty k uzlu) se podobá např. cestě k souboru na disku. Jako názornou ukázkou si uveďme adresu textového uzlu z obrázku 9 označeného šipkou:

```
/HTML[2]/BODY[1]/DIV[1]/text() [2]
```

Čísla uvnitř hranatých závorek v tomto výrazu vyjadřují pozici onoho uzlu mezi uzly stejného typu a jména v jeho úrovni. Lze takto přesně vybrat jak jeden konkrétní uzel (v případě našich potřeb pro definici hodnoty parametru sledování typu „textový/číselný obsah elementu“), tak i skupinu uzlů (jako definice hodnoty pro typ „počet elementů“). Více informací o XPath a návod na psaní těchto výrazů v češtině lze najít v [2].

## 5.3 Systém vyhodnocování obsahu stránek

Nyní, když již víme jak adresovat jednotlivé prvky v HTML dokumentu, se můžeme začít zabývat implementací vyhodnocování obsahu stránek. Jak v tomto případě postupovat jsme si obecně popsali v podkapitole 4.2.1. V této části se budeme zabývat konkrétní implementací algoritmu v jazyce PHP.

Nejprve, po získání seznamu všech sledovaných stránek, zjistíme, zda aktuálně vybranou stránku je zapotřebí v tomto cyklu vyhodnotit. To provedeme porovnáním aktuálního času s časem vzniklým součtem hodnot `posledni_sledovani` a `interval_sledovani`. Pokud je aktuální čas větší, pokračujeme dalším krokem.

V další fázi, po vyhodnocení, že v tomto čase je třeba obsah sledované stránky sledovat, je třeba zajistit načtení stránky do proměnné. K tomu je k dispozici standardní PHP funkce `fopen()`, nebo knihovna `curl`.

Knihovna `curl` nabízí mnoho možností jak získávat soubory z jiných serverů a podle referencí pracuje rychleji než `fopen`, avšak bohužel si neporadí s přesměrováním na jiný server, jenž některé weby při otvírání HTML dokumentů provádějí. Musíme se tak spokojit s funkcí `fopen()`, pro kterou musí být v nastavení `php` povolena direktiva `allow_url_fopen`. Na fakultním serveru `eva` je toto povoleno a tak nám nic nebrání funkci `fopen()` použít.

Nyní již máme uloženou stránku v řetězcové proměnné, avšak nastává problém s kódováním znaků na stránce. V našich končinách se nejvíce používá kódování `iso-8859-2`, `windows-1250` nebo `utf-8`. Musíme tedy zjistit, v jakém je kódování námi stažená stránka. Tuto informaci je možné získat ze samotného HTML dokumentu, konkrétně z tagu `META`. Např.:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Můžeme ji také získat z hlavičky HTTP odpovědi od serveru (pomocí funkce `get_headers()`), ve které je kódování reprezentováno následujícím řádkem:

```
Content-Type: text/html; charset=windows-1250
```

Je vhodnější vybrat druhou možnost, protože kódování nebývá vždy tagem `META` zmíněno. Převědeme tedy pomocí funkce `iconv()` řetězec s obsahem `html` dokumentu do kódování využívaném v této aplikaci. Nyní můžeme dokument dále zpracovávat.

V následujícím kroku je potřeba převést otevřený HTML dokument na reprezentaci vhodnou pro další zpracování. Jak jsme si již v textu podkapitoly 5.2 řekli, použijeme `DOM`, jenž je součástí jádra `PHP`. Tím získáme možnost pohodlně pracovat s dokumentem prostřednictvím `DOM API`. Zároveň je nám umožněno pomocí třídy `DOMXPath` v tomto dokumentu vyhodnocovat `XPath` výrazy, což bude náležitě využito.

Popíšeme si tedy, jak bude probíhat vyhodnocování jednotlivých typů parametru sledování:

- *typ počet elementů* – vyhodnotíme v dokumentu hodnotu parametru (`XPath` výraz) a zjistíme počet uzlů, které jsme vyhodnocením získali

- typ *velikost dokumentu* – určíme délku řetězce, jež obsahuje načtenou webovou stránku (DOM reprezentace dokumentu byla vytvořena z tohoto řetězce)
- typ *textový obsah elementu* – vyhodnotíme v dokumentu hodnotu parametru a získáme text z textového uzlu (nebo z více uzlů, jejichž text pospojujeme – text v jednom elementu může být rozdělen do více textových uzlů). Takto získaný text dále upravíme podle nastaveného zpracování v parametru sledování, tj. ořízneme text o daný počet znaků na začátku a konci textu. Tím získáme výsledný řetězec, jenž porovnáme s naposledy uloženým řetězcem. Pokud jsou rozdílné, přidáme nový záznam sledování
- typ *číselný obsah elementu* – postupujeme stejně jako u předešlého kroku, s tím rozdílem, že získaný řetězec upravíme tak, aby reprezentoval celé nebo desetinné číslo.

Takto získaná data vložíme do tabulky bcZaznamSledování a aktualizujeme hodnotu posledni\_sledovani v tabulce bcSledovanaStranka u právě vyhodnocené stránky.

## 5.4 Export a import sledovaných dat

Data vzniklá sledováním stránek je nutné umožnit exportovat ve formátu XML a zpětně je i importovat.

Provedení exportu dat je snadný úkol. Nejdříve si však musíme definovat, jakou strukturu bude mít vytvářený XML dokument. Prvně zvolíme jméno kořenového elementu, v našem případě například SOWS (od funkce aplikace – Sledování Obsahu Webových Stránek). Dále vytvoříme element, který bude identifikovat, ke kterému parametru sledování data náležejí, a který bude obsahovat další elementy specifikující samotná data – položky. Budeme mít tedy následující strukturu:

```
<SOWS>
  <statistiky id="číslo parametru">
    <polozka...>
    <polozka...>
  </statistiky>
</SOWS>
```

Nyní se lze rozhodovat mezi uložením informací o záznamu sledování (čas a hodnota) jako atributů elementu polozka nebo jako synovských uzlů tohoto elementu:

```
<polozka cas="timestamp" hodnota="hodnota"/>
```

nebo

```
<polozka>
  <cas>timestamp</cas>
  <hodnota>hodnota</hodnota>
</polozka>
```

Byla zvolena druhá možnost, jelikož atributy by neměly být nositelem dat, ale pouze informací o datech. Konečná struktura exportovaného XML dokumentu bude následující:

```
<SOWS>
  <statistiky id="1">
    <polozka>
      <cas>2007-03-21T20:50:31</cas>
      <hodnota>321</hodnota>
    </polozka>
    <polozka>
      <cas>2007-03-21T21:50:31</cas>
      <hodnota>345</hodnota>
    </polozka>
  </statistiky>
</SOWS>
```

Při importování ze souboru XML je potřeba zaručit, aby struktura importovaného dokumentu byla shodná s námi definovaným vzorem. To lze zajistit například vytvořením DTD (definice typu dokumentu) nebo pomocí XMLSchema. Pro naše použití se zdá být XMLSchema výhodnější. Není třeba pro kontrolu dokumentu uvádět DTD přímo v XML souboru, ale XML dokument načtený do DOM struktury se ověří metodou `schemaValidate()` přímo proti zadanému XMLSchema souboru. Definici XMLSchema je možno vidět v příloze 2.

## 5.5 Webové uživatelské rozhraní

### 5.5.1 Celkový koncept webu

Téměř celá aplikace je koncipována jako jedna HTML stránka s měnícím se obsahem. Tento obsah stránek je zásadně určován parametry předávanými společně s adresou, tedy pomocí metody GET. Celý web je kódován ve znakové sadě windows-1250.

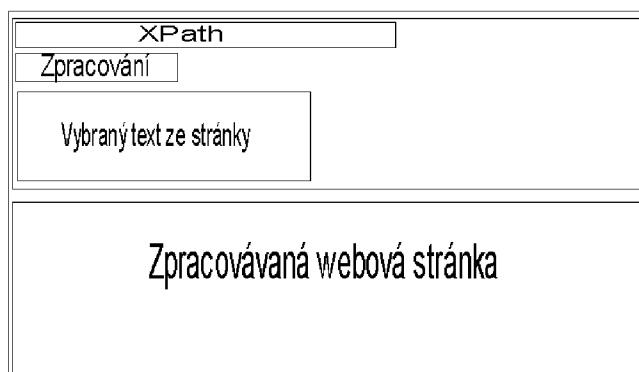
## 5.5.2 Uživatelsky přívětivé nastavení aplikace

V podkapitole 5.3 jsme se seznámili se způsobem určování (XPath) hodnoty parametru sledování. Ačkoliv se jedná o známý jazyk pro adresování uvnitř XML dokumentů, je potřeba nabídnout uživateli možnost snadného získání této adresy z jím požadované stránky.

Vhodným způsobem získání adresy je upravení webové stránky způsobem, kdy by uživatel kurzorem vybral text, který chce sledovat. K tomuto účelu vytvoříme skript, jenž zpracuje zadanou webovou stránku tak, že pro každý element obsahující neprázdný textový uzel nalezneme jeho adresu v rámci dokumentu. Pro další krok využijeme možností javascriptu a to tak, že každému tomuto elementu přidáme akci onclick obsahující volání námi definované javascriptové funkce s parametry, které bude tvořit nalezená adresa elementu a samotný obsah textového uzlu.

„Kliknutím“ na text se tedy provede funkce, jež zobrazí XPath adresu a text na další webové stránce, kde uživatel označením části textu definuje, jak se má text ořezávat. Takto získané hodnoty se poté přenesou do nastavení parametru sledování.

Koncept stránky, zajišťující toto získávání nastavení vypadá následovně:



10 - Koncept stránky pro pomoc s nastavením parametru sledování

## 5.5.3 Grafové znázornění sledovaných dat

Hlavní náplní aplikace má být informování uživatele o změnách jím definovaných hodnot na webových stránkách. Tyto záznamy sledování zobrazíme v textové formě (čas – hodnota), avšak ve třech případech, kdy typem parametru sledování je velikost stránky, počet elementů nebo číselný obsah elementu, můžeme tato data zobrazit promítnutím do grafu.

Pro kreslení grafu využijeme v PHP vestavěnou knihovnu GD, určenou pro základní vytváření obrázků. Z nabízených možností formátů výsledného obrázku (gif, png, jpeg) vybereme png pro jeho bezztrátovou kompresi, která je oproti gif formátu účinnější.

Časový rozsah grafu bude v souladu s definicí obsahu stránky zobrazující statistiky uvedenou v podkapitole 4.6.3.

## 5.6 RSS kanál

Pro informování uživatele o změnách na jím sledovaných stránkách slouží mimo jiné i RSS kanály. Nejdříve ale musíme definovat, jaké konkrétní údaje budeme takto uživateli předávat. Byly zvoleny následující:

- titulek položky: *staráHodnota* -> *nováHodnota* – *popisParametruSledování*
- popis položky: Na stránce *urlSledovanéStránky* došlo ke změně hodnoty sledovaného parametru „*popisParametruSledování*“. Původní hodnota *staráHodnota* (*casStareHodnoty*) – Nová hodnota *nováHodnota* (*casNovéHodnoty*).

K odebrání změn pomocí RSS kanálu použije uživatel své přihlašovací údaje do aplikace a to následujícím způsobem:

<http://login:heslo@webováAdresaAplikace/rss.php>

## 5.7 Možnosti využití aplikace

Vytvořenou aplikaci je možné využít k získání statistických informací o struktuře webové stránky, a to sledováním jak velikosti celého dokumentu, tak počtu elementů na stránce. Tímto můžeme vyhodnocovat rozvíjení se webových stránek (přibývající odkazy, obrázky,...) atp. Další možností využití je kupříkladu sledování ceny výrobku na e-shopu, kde budeme sledovat obsah elementu obsahující cenu převedený na číslo. Tím získáme přehled o růstu nebo klesání cen a statistikách z těchto cen získaných. Poslední jmenovanou možností (ale jistě ne poslední v praktickém využití) je sledování verzí programů, jejichž aktuální verze bývá zobrazována na webových stránkách těchto programů. Uživatel je pak informován o změně pohodlně prostřednictvím RSS kanálů.

## 6 Závěr

Cílem této práce bylo vytvořit webovou aplikaci, jež vyhodnocuje změny webových stránek v čase a informace o těchto změnách zprostředkovává uživateli. Při návrhu aplikace byly zohledněny všechny dané požadavky na funkčnost, které jsou ve výsledku splněny. Při realizaci aplikace jsem využil zejména své znalosti z práce s DOM, včetně XPath apod. Jediným mně známým problémem je převod kódování znakové sady při zpracování stránek, kde u malého počtu webů nebyl z neznámých příčin převod korektní.

Pro další vývoj aplikace je možné se zamyslet nad možností vyhodnocování textu získaného sloučením textových uzlů z více elementů. Tedy například ze zdrojového textu

```
<p>text <b>tučné písmo</b> text</p>
```

by se získal text „text tučné“. Také by bylo možné nalézt další možnosti ve vyhodnocování webových stránek a případně i upřesnit možnosti zpracování textu ze stránky, kdy se v tuto chvíli pouze odstraňuje daný počet znaků ze začátku a konce získaného textu (tímto dochází k problému, pokud se text získaný před zpracování mění i v jiném místě než v tom, které sledujeme).

# Literatura

- [1] *Mozilla developer center: DOM* [online]. 2006 [cit. 2007-05-01].  
Dostupný z WWW: <<http://developer.mozilla.org/cs/docs/DOM>>
- [2] Nič Miroslav: *XPath Tutorial* [online]. 2000 [cit. 2007-05-01].  
Dostupný z WWW:<[http://www.zvon.org/xxl/XPathTutorial/General\\_cze/examples.html](http://www.zvon.org/xxl/XPathTutorial/General_cze/examples.html)>
- [3] ŠKULTÉTY, Rastislav. *JavaScript : Programujeme internetové aplikace*.  
2. aktualiz. vyd. Brno : Computer Press, 2004. 224 s. ISBN 80-251-0144-4.
- [4] ULLMAN, Larry. *PHP : Pokročilé Programování Pro World Wide Web*.  
Překlad: Martin Bartel. 1. vyd. Praha : SoftPress, 2003. 512 s. ISBN 80-86498-36-4.
- [5] Wikipedie: Otevřená encyklopedie: *Webová aplikace* [online]. 2007 [cit. 2007-05-01].  
Dostupný z WWW:  
<[http://cs.wikipedia.org/w/index.php?title=Webov%C3%A1\\_aplikace&oldid=1129617](http://cs.wikipedia.org/w/index.php?title=Webov%C3%A1_aplikace&oldid=1129617)>
- [6] WILLIAMS, Hugh, LANE, David. *PHP a MySQL : Vytváříme webové databázové aplikace*.  
David Krásenský. 1st edition. Praha 4 : Computer Press, 2002. 530 s. , 5. ISBN 80-7226-760-4.



# Seznam příloh

Příloha 1. SQL dotazy pro vytvoření tabulek

Příloha 2. XMLSchema pro XML soubor importu/exportu dat

Příloha 3. Uživatelská příručka

Příloha 4. CD obsahující tuto práci a zdrojové texty webové aplikace

# Příloha 1

SQL dotazy definující databázi.

```
CREATE TABLE `bcUzivatel` (  
  `login` VARCHAR( 20 ) NOT NULL ,  
  `jmeno` VARCHAR( 20 ) ,  
  `prijmeni` VARCHAR( 20 ) ,  
  `heslo` VARCHAR( 20 ) NOT NULL ,  
  `prava` ENUM( 'admin', 'user' ) DEFAULT 'user' NOT NULL ,  
  `enabled` INT DEFAULT '0' NOT NULL ,  
  PRIMARY KEY ( `login` )  
);
```

```
CREATE TABLE `bcSledovanaStranka` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `uzivatel` VARCHAR( 20 ) NOT NULL ,  
  `url` VARCHAR( 50 ) NOT NULL ,  
  `interval_sledovani` INT NOT NULL ,  
  `posledni_sledovani` DATETIME DEFAULT '0' NOT NULL ,  
  PRIMARY KEY ( `id` )  
);
```

```
CREATE TABLE `bcParametrSledovani` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `id_stranky` INT NOT NULL ,  
  `typ` INT NOT NULL ,  
  `hodnota` TEXT NOT NULL ,  
  `zpracovani` VARCHAR( 50 ) ,  
  `popis` TEXT NOT NULL ,  
  PRIMARY KEY ( `id` )  
);
```

```
CREATE TABLE `bcZaznamSledovani` (  
  `id_parametru` INT NOT NULL ,  
  `cas` DATETIME NOT NULL ,  
  `hodnota` TEXT NOT NULL ,  
  PRIMARY KEY ( `id_parametru` , `cas` )  
);
```

# Příloha 2

XML schema definující strukturu dat pro export a import sledovaných dat

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SOWS">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="statistiky" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="polozka" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="cas" type="xs:dateTime"/>
                    <xs:element name="hodnota" type="xs:string"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:nonNegativeInteger"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# Příloha 3

## Sledování obsahu webových stránek

*uživatelská příručka*

Webovou aplikaci je v současné době (květen 2007) možno nalézt v provozu na fakulním studentském webovém serveru eva <<http://www.stud.fit.vutbr.cz/~xhochm01/bc/>>.

### Registrace do aplikace

- Po zobrazení webové aplikace klikneme na odkaz registrace.
- Vyplníme povinné (případně i volitelné) údaje o uživateli.
  - V případě chyby opravíme zadané údaje.
- Po úspěšné registraci se můžeme přihlásit.

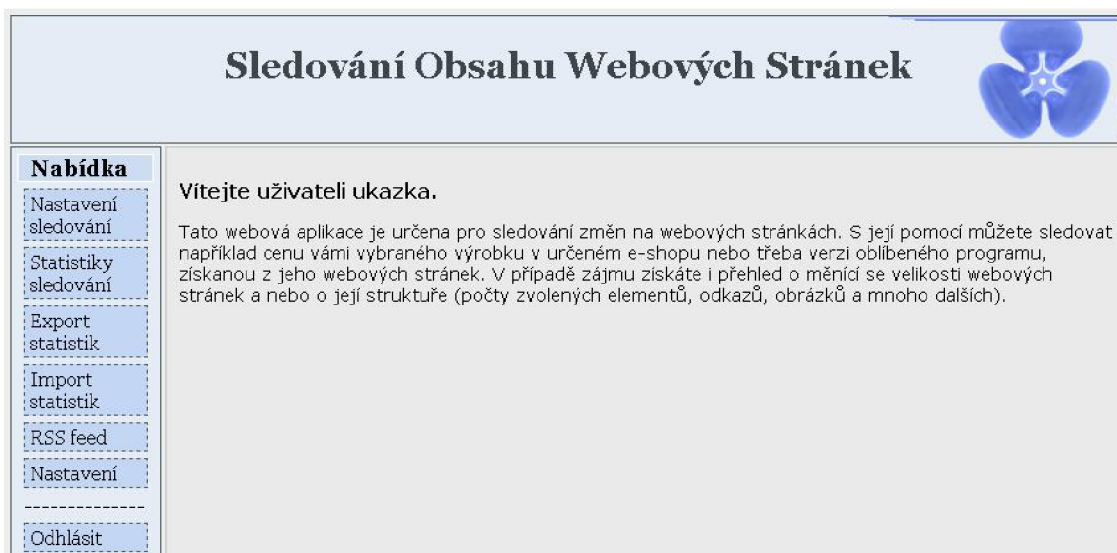
The screenshot shows a web interface for user registration. At the top, the title 'Sledování Obsahu Webových Stránek' is displayed next to a blue logo. On the left, there is a login section with fields for 'Login:' and 'Heslo:', a 'Přihlásit' button, and a 'registrace' link. The main area is titled 'Registrace uživatele:' and contains several input fields: 'login:' (with a sample value 'ukazka' and a note 'alespon 6 písmen, pouze znaky a-z, A-Z, 0-9, - a \_'), 'jméno:', 'příjmení:', 'heslo:' (with a note 'zadejte heslo (alespon 6 znaků)'), and 'znovu heslo:' (with a note 'opět heslo (pro vyloučení překlepu)'). There are 'registrovat' and 'zrušit' buttons at the bottom of the registration form, and a note '\* ) Povinné údaje'.

*1 - registrace do aplikace*

### Přihlášení do aplikace

- Do pole Login napíšeme uživatelské jméno.
- Do pole Heslo zadáme heslo.
- Klikneme na tlačítko přihlásit.

Po přihlášení je uživateli k dispozici nabídka akcí které jsou k dipozici (viz obázek 2).



2 - Hlavní okno aplikace

## Nastavení sledování stránek

- V nabídce aplikace zvolíme položku *Nastavení sledování*.

V této sekci je možné přidat novou sledovanou stránku, procházet uložené sledované stránky, vyhledávat v nich podle jejich url, editovat je, mazat a zobrazovat jejich parametry.



3 - Sekce Nastavení sledování

## Nastavení sledování stránek – přidání sledované stránky

- V nabídce aplikace zvolíme položku *Nastavení sledování*.
- Nyní klikneme na **+Přidat stránku**.
- Zadáme url stránky, jejíž obsah chceme sledovat.
- Vybereme interval v jakém chceme sledování provádět.
- Klikneme na tlačítko *Přidat stránku*.

Nastavení -> Stránky -> přidání:

url

url stránky pro sledování (včetně protokolu: http:// nebo https://)

interval sledování

interval udává jak často se bude sledování stránky provádět

#### 4 - Přidání sledované stránky

Zobrazí se seznam parametrů sledování právě přidané stránky (obrázek 5). Zde je možné procházet parametry sledování této stránky, editovat je, mazat a přidat nový parametr sledování.

Nastavení -> Stránky -> přidání:

url

url stránky pro sledování (včetně protokolu: http:// nebo https://)

interval sledování

interval udává jak často se bude sledování stránky provádět

#### 5 - Sekce Parametry Sledování

- Klikneme na **+Přidat parametr**.
- Vybereme typ sledování: počet elementů, obsah elementu (číslo), obsah elementu (text), velikost stránky.
- Do pole hodnota zapíšeme XPath cestu k datům a nastavíme zpracování dat (oříznutí textu před a za námi vybranou částí). Tyto dvě hodnoty lze vyplnit automaticky pomocí využití funkce získat xpath (popsáno níže).
- Napíšeme popis pro identifikaci sledovaných dat.
- Klikneme na tlačítko *Přidat parametr*.

Nastavení -> Stránky -> Parametry sledování -> přidání:

Stránka: <http://www.czechcomputer.cz/product.jsp?artno=36180>

typ

typ sledování (vyberte jednu z možností)

hodnota

cesta k datům určeným pro sledování reprezentovaná XPath výrazem [získat xpath](#)

zpracovani

definice, jak se má získaný text zpracovat - oříznout (zápis: odstranitZnakÚPřed,odstranitZnakŮza)

popis

popis parametru pro potřeby rozeznání v jiných sekcích aplikace

kontrola získání dat

#### 6 - Přidání parametru sledování

## Funkce získat xpath

- Kliknutím na *získat xpath* v nastavení parametru sledování se zobrazí okno s obsahem stránky, jejíž parametr sledování nastavujeme.
- Najetím kurzorem co nejpřesněji na text, jehož obsah chceme získat a kliknutím na něj se získá xpath adresa k tomuto textu. Tento text se také přeneso do části text (ohraňčený červeně). V této oblasti kliknutím levým tlačítkem a tažením myši vybereme část textu, která nás zajímá. Tím se získá hodnota zpracování.
- Nyní klikneme na tlačítko Nastavit a získané hodnoty se přenesou do okna s nastavením parametru sledování, ze kterého jsme vycházeli.

**Získání XPath ze stránky:**

xpath:

zpracování: 0,3

Klikněte kurzorem na níže zobrazené stránce co nejpřesněji na vámi zvolený text. Ve vybraném textu, jenž se objeví v červeně ohraničené oblasti, vyberte kliknutím a tahem kurzoru část, kterou chcete sledovat.

text: 11 817,- Kč

**Hewlett-Packard LP2065 - LCD 20"**

Výrobce: Hewlett-Packard  
Publikováno: 28.2.2006

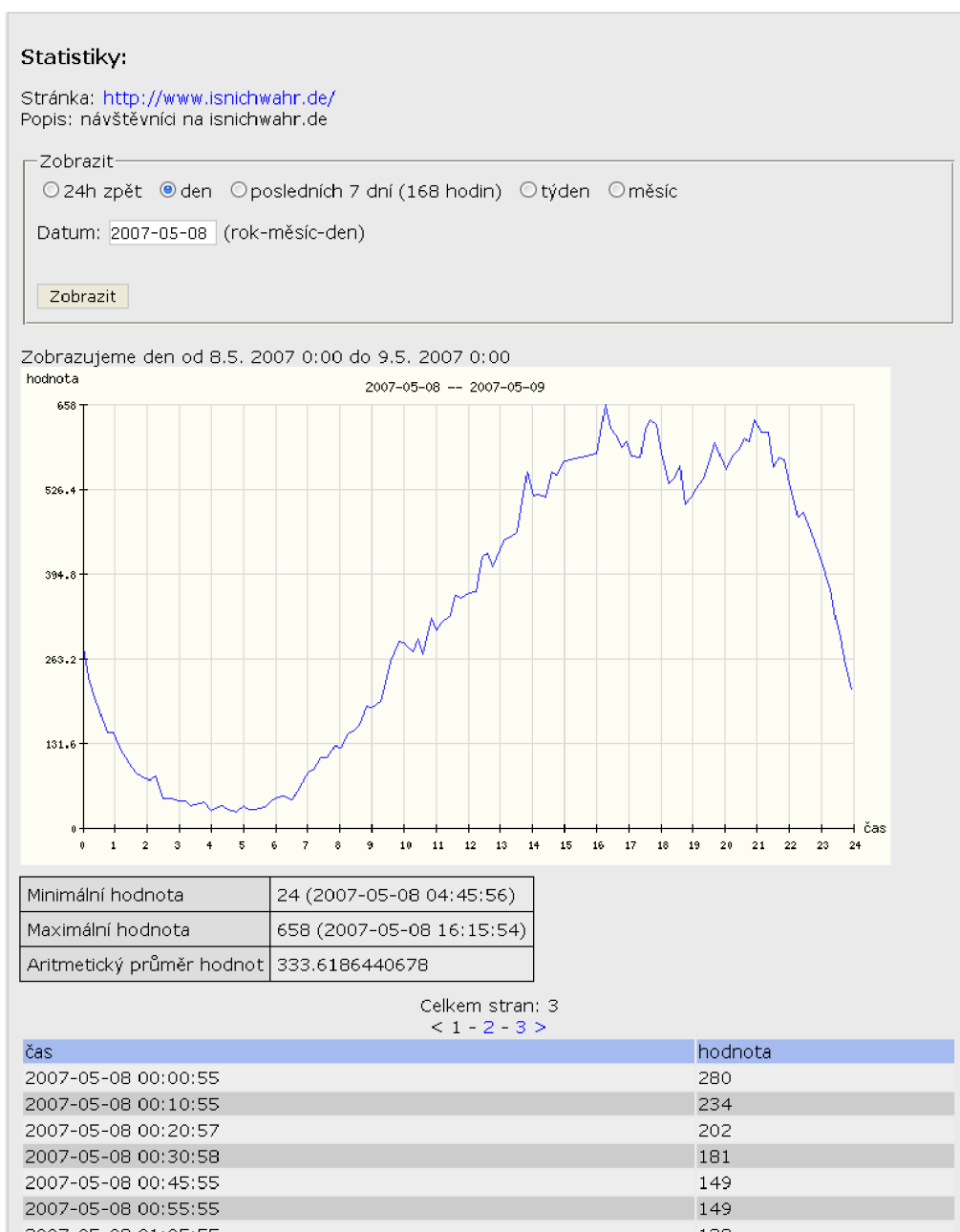
Kód: 36180  
Part number: EF227A4#ABB  
Záruční doba: 3 roky  
Kde reklamovat: Hewlett-Packard  
Skladem: 3  
Dodací lhůta: ihned

Cena bez DPH: 9 930,- Kč  
Cena s DPH: 11 817,- Kč

7 - Získání XPath ze stránky

## Zobrazení statistik sledování

- V nabídce aplikace zvolíme *Statistiky sledování*. Dostaneme se do sekce *Statistiky -> Stránky*.
- Klikneme na odkaz *Parametry* u námi vybrané stránky.
- Dostaneme se do sekce *Statistiky -> Stránky -> Parametry sledování*.
- Klikneme na odkaz *Statistiky* u námi vybraného parametru sledování.
- Zobrazí se sekce *Statistiky*, ve které si můžeme zvolit časový rozsah, který chceme sledovat. Pokud je to možné, zobrazí se data (kromě textového výpisu v tabulce) i ve formě grafu. Více je vidět na obrázku 8.



8 - Sekce Statistiky

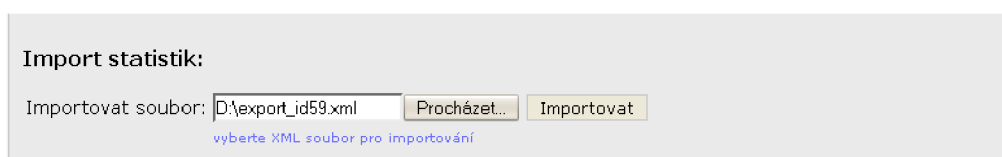


## Export sledovaných dat

- V nabídce aplikace zvolíme *Export statistik*. Dostaneme se do sekce *Export -> Stránky*.
- Klikneme na odkaz *Parametry* u námi vybrané stránky.
- Dostaneme se do sekce *Export -> Stránky -> Parametry sledování*.
- Klikneme na odkaz *Exportovat* u námi vybraného parametru sledování.
- Webový prohlížeč zobrazí dialog pro stažení XML souboru obsahující zaznamenaná data pro vybraný parametr sledování.

## Import sledovaných dat

- V nabídce aplikace zvolíme položku *Import statistik*.
- Kliknutím na tlačítko *Procházet* se zobrazí dialog pro výběr souboru. V něm zvolíme soubor ve formátu XML, jehož obsah chceme importovat.
- Po stisku tlačítka *Importovat* se provede importování dat ze souboru nebo se zobrazí případná chyba ve vybraním souboru.



9 - Sekce Import statistik

## RSS kanál

Pro možnosti snadného přehledu o změnách ve sledovaných datech je určen RSS kanál.

- V nabídce aplikace zvolíme položku *RSS feed*.
- Zobrazí se dialog do kterého zadáme své uživatelské jméno a heslo. Tím získáme přístup ke svému RSS kanálu.
- Další možností je zapsat své uživatelské jméno přímo do adresy k RSS kanálu. Tedy např. <http://login:heslo@www.stud.fit.vutbr.cz/~xhochm01/bc/rss.php>.

## Nastavení uživatele

- V nabídce aplikace zvolíme položku *Nastavení*.

Změna údajů o uživateli:

- Provedeme změnu jména a/nebo příjmení.
- Klikneme na tlačítko *Provést změny*.

Změna hesla:

- Do položky *staré heslo* napíšeme aktuální heslo.
- Do položek *nové heslo* a *znovu nové heslo* napíšeme nové heslo
- Klikneme na tlačítko *Provést změny*.

Odstranění účtu:

- Do položky *staré heslo* napíšeme aktuální heslo.
- Klikneme na tlačítko *Smazat účet*.

Nastavení uživatele:

login: ukazka  
jméno:   
příjmení:   
staré heslo: \*  
zadejte vaše aktuální heslo (v případě změny hesla nebo smazání účtu)  
nové heslo: \*  
zadejte nové heslo (alespoň 6 znaků)  
znovu nové heslo: \*  
opět nové heslo (pro vyloučení překlpu)

\*) Povinné údaje

10 - Sekce Nastavení uživatele

## Správa uživatelů – admin sekce

- V nabídce aplikace zvolíme položku *Uživatelé*. Tím se dostaneme do sekce *Správa uživatelů*.
- Zobrazí se nám seznam všech uživatelů s možností vyhledávání podle loginu, aktivace/deaktivace a smazání vybraného uživatelského účtu.

Správa uživatelů:

Vyhledat login:

Celkem stran: 1  
< 1 >

#	login	jméno	prijmeni	prava	aktivní		
1	hochmal			user	NE	<a href="#">aktivovat</a>	<a href="#">smazat</a>
2	chomat	petr	hochmal	user	ANO	<a href="#">deaktivovat</a>	<a href="#">smazat</a>
3	chomata			user	NE	<a href="#">aktivovat</a>	<a href="#">smazat</a>
4	kanec83	Petr	Janec	user	ANO	<a href="#">deaktivovat</a>	<a href="#">smazat</a>
5	ukazka			user	ANO	<a href="#">deaktivovat</a>	<a href="#">smazat</a>

Celkem stran: 1  
< 1 >