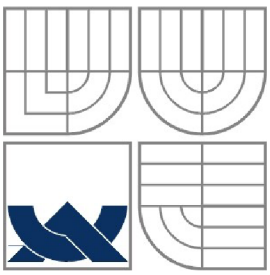
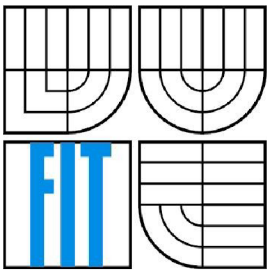


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

STATISTIKY PŘÍSTUPU WEBOVÝCH STRÁNEK

STATISTICS OF WEB-SITES ACCESS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL PETŘÍK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAEL KUNC

BRNO 2007

Zadání bakalářské práce/5749/2006/xpetri10

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Petřík Pavel**
Obor: Informační technologie
Téma: **Statistiky přístupu webových stránek**
Kategorie: Web

Pokyny:

1. Seznamte se s technologiemi pro vývoj dynamických webových aplikací.
2. Navrhněte systém pro ukládání a zobrazování informací o přístupu k webovým stránkám, které se v systému zaregistrují. Pro tyto weby bude aplikace ukládat detailní informace o návštěvnících. Z těchto informací bude možno zobrazit na základě zvolených kritérií příslušná data. Zaměřte se na různé možnosti grafického znázornění. Dále v tomto systému navrhněte funkci, která bude odhadovat návštěvnost v následujícím období.
3. Systém statistik realizujte a otestujte ho. Do testování zahrňte také odhady návštěvnosti, které porovnáte se skutečnou návštěvností.
4. Zhodnoťte dosažené výsledky a diskutujte další možné pokračování tohoto projektu.

Literatura:

- Stejskal, J.: Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Computer Press, 2004.
- Gilmore, W. J.: Velká kniha PHP5 & MySQL. Zoner Press, 2005.
- Ullman, L.: PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. Computer Press, 2004.
- Han, J., Kamber, M.: Data Mining: Concepts and Techniques, str. 489-497. Morgan Kaufmann Publishers, 2006.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

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 odevzdá 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: **Kunc Michael, 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
612 66 Brno, Božetěchova 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: **Pavel Petřík**
Id studenta: 84200
Bytem: Zvole 216, 789 01 Zábřeh
Narozen: 23. 03. 1985, Šumperk
(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: Statistiky přístupu webových stránek
Vedoucí/školitel VŠKP: Kunc Michael, 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 tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....
Nabyvatel

.....
Autor

Abstrakt

Cílem této práce je vytvoření systému statistik. Systém statistik bude umožňovat sledování, zpracování a zaznamenávání návštěvnosti webových stránek. Projekt zahrnuje návrh systému statistik včetně odhadů návštěvnosti a jeho programovou realizaci. Systém je implementován pomocí technologií PHP, MySQL, HTML, CSS a JavaScript.

Klíčová slova

statistika, HTML, CSS, JavaScript, PHP, MySQL, webový server, databáze, WWW

Abstract

Statistics system creating is the purpose of this Project. Statistics system will make possible to monitor, process and record attendance the Websites. The Project includes Statistics system proposal including visit forecasts and it's programme realization. System is implemented by the help of technology PHP, MySQL, HTML, CSS and JavaScript.

Keywords

Statistica, HTML, CSS, JavaScript, PHP, MySQL, Web server, Database system, WWW

Citace

Pavel Petřík: Statistiky přístupu webových stránek, bakalářská práce, Brno, FIT VUT v Brně, 2007

Statistika přístupu webových stránek

Prohlášení

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

.....
Pavel Petřík
Datum

Poděkování

Děkuji především mému vedoucímu bakalářské práce Ing. Michaelu Kuncovi za poskytnuté konzultace, různé nápady na implementaci a řešení této bakalářské práce.

Obsah

1 Úvod

V současné době jsou internetové prezentace velmi rozšířené z pohledu snad všech institucí a téměř každá instituce vlastní své webové stránky. Možnosti využití internetu z hlediska webových prezentací skýtá různé příležitosti. Náplň některých webových stránek má pouze informativní charakter avšak existují i takové společnosti, které je používají například k elektronickému obchodování. Snad každého kdo vlastní webové stránky pro jakékoliv účely zajímá, zda jsou jeho webové stránky navštěvovány. Na tuto otázku dokáže odpovědět systém statistik přístupu webových stránek, jejichž návrh a implementace je předmětem této bakalářské práce.

Systémy statistik se od sebe odlišují a předhánějí se v množství informací, které dokáží o návštěvníkovi webových stránek zjišťovat. Množství zjišťovaných informací může později vypovídat o tom, jakým vývojem se mají stránky dále ubírat. Například optimalizace pro různé prohlížeče, jazykové mutace stránek, pokud víme, že webové stránky navštěvují i zahraniční návštěvníci, ale také jako podklady pro různé marketingové účely. Předběžné odhady návštěvnosti v následujícím období potom mohou být použitelné jak pro správce a provozovatele webu, tak například v oblasti marketingu.

Rámec tohoto projektu zahrnuje návrh systému statistik včetně odhadů návštěvnosti a jeho implementaci. Ve 2. kapitole jsem zdokumentoval požadavky na systém statistik přístupu webových stránek. Kapitola 3 se zabývá použitými technologiemi pro vývoj projektu. Kapitola 4 je věnována analýze a návrhu systému statistik a odhadům návštěvnosti, včetně popisu funkčnosti metody, kterou jsem použil a její demonstraci na příkladu. 5. kapitola popisuje možnosti webového uživatelského rozhraní, jeho funkcionalitu a také se zabývá otázkou bezpečnosti vzhledem k neoprávněným uživatelům.

2 Specifikace požadavků na funkčnost aplikace

2.1 Požadavky na systém statistik

Z hlediska uživatele byl kladen důraz na splnění následujících kritérii:

1. Navrhnout systém pro ukládání informací o přístupu k webovým stránkám včetně zpracování informací o návštěvnících
2. Zaměřit se na to, jak z takto získaných informací o přístupu k webovým stránkám zobrazit na základě zvolených kritérií příslušná data. Při zobrazení se zaměřit na různé možnosti grafického znázornění
3. Navrhnout funkci, která bude na základě dříve zaznamenaných dat o přístupech na webové stránky odhadovat návštěvnost v následujícím období
4. Pomocí dříve zmíněných technologií navrhnout systém, který by umožnil jak registraci uživatele spojenou s případným poskytováním dalších služeb, tak i stránek, které si přeje sledovat, včetně nastavení kritérií pro sledování. Systém by po úspěšné registraci uživatele měl rovněž nabízet možnost přihlášení spojenou se zjištěním, jak na tom jsou zaregistrované stránky s návštěvností za uplynulé období nebo si nechat vypočítat přibližný odhad návštěvnosti v následujícím období

Při návrhu jsem bral v potaz požadavky kladené na funkčnost aplikace a pomocí doporučených technologií hledal způsoby, jak systém statistik přístupu na webové stránky implementovat.

3 Použité technologie

Pro tvorbu systému jsem zvolil multiplatformní skriptovací jazyk PHP v kombinaci s databázovým systémem MySQL. Důvody pro takovou volbu jsou především relativně jednoduchá implementovatelnost, výkon a v neposlední řadě také skutečnost, že v obou případech se jedná o volně šiřitelný software. Dále byly použity jazyky HTML, CSS a JavaScript.

3.1 HTML

HTML (HyperText Markup Language) je značkovací jazyk pro vytváření stránek v systému World Wide Web, který umožňuje publikaci stránek na internetu. Jazyk je podmnožinou dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Jazyk je podporován snad všemi prohlížeči a jeho výhodou je jeho snadné pochopení. Podpora v prohlížečích není bohužel vždy stejná a tento neduh se snaží odstranit standardizační organizace W3C, která stanovuje validitu kódu. Bohužel, i když je kód validní, stává se, že je na různých prohlížečích zobrazen rozdílně. HTML je čistý dokument, který lze vytvářet jakýmkoliv textovým editorem. Je členěn strukturálně a obsah stránky definuje pomocí tzv. tagů. Rozlišujeme tagy párové a nepárové, jsou vždy uzavřeny mezi znaky < a > a mohou nést různé atributy.

Vývoj jazyka HTML byl ukončen verzí 4.01, specifikace ale platí dále a je stále možné vyvíjet webové aplikace podle HTML 4.01. Nástupcem a aktuálně vyvíjeným jazykem pro tvorbu dokumentů na webu je jazyk XHTML.

XHTML (eXtensible HyperText Markup Language) je rovněž značkovací jazyk a jedná se o aplikaci XML, čímž klade větší důraz na přesně definovanou strukturu s přesnými pravidly.

3.2 CSS

CSS je zkratka anglického názvu Cascading Style Sheet neboli kaskádové styly. Jedná se o množinu pravidel pro formátování obsahu souborů jazyka HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C a aktuální specifikací je CSS2. Hlavním cílem při návrhu CSS bylo oddělit vzhled dokumentu od samotného obsahu. Kaskádové styly umožňují například definovat jednotný vzhled elementu pro celý dokument pomocí jediného zápisu pravidla. Pokud je například potřeba změnit design webu, stačí mnohdy změnit pouze jeden soubor kaskádového stylu .css a změna se aplikuje na celý web.

Některé další výhody CSS ve srovnání s HTML:

- Oddělení struktury a stylů několika způsoby zápisu
- Širší množina formátovacích možností
- Snadná tvorba a údržba konzistentního stylu

3.3 JavaScript

JavaScript je multiplatformní objektově orientovaný skriptovací jazyk. V současnosti se zpravidla používá jako interpretovaný skriptovací jazyk pro WWW stránky a je vkládaný přímo do HTML kódu stránky. Jeho syntaxe je podobná jazykům z rodiny C/C++/Java. Pomocí JavaScriptu můžeme například provádět změny v obsahu stránky, aniž bychom odeslali požadavek na server, což může vést k usnadnění a k urychlení práce s aplikací. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, seznamy, textová pole, atd.) nebo tvořeny efekty obrázků a animace.

Některé výhody JavaScriptu:

- Možnost dynamicky reagovat na událost způsobenou klientem (např. Pohyb kurzoru, stisknutí tlačítka myši nebo klávesnice, načtení stránky v prohlížeči, atd.)

Zdrojový kód se do HTML vkládá buď jako interní styl pomocí tagu `script` nebo jako externí soubor s příponou .js.možnosti dynamicky reagovat na událost způsobenou klientem.

3.4 Webový server

Webových serverů existuje celá řada. Jejich hlavním úkolem je naslouchat na některém portu, a když od klienta přijde požadavek na stránku, tak server vytvoří potomka, který následně provede obsluhu požadavku. Nejčastěji je tímto portem port 80 – HTTP. Protokol HTTP (HyperText Transfer Protocol) je srdcem sítě WWW, protože definuje způsob, jimiž se mezi sebou klienti a servery vzájemně dorozumívají. Nejrozšířenějším serverem na webu je Apache. Vděčí za to své modulárnosti, rychlosti a docela široké konfigurovatelnosti.

3.5 PHP

PHP Hypertext Preprocessor je skriptovací jazyk běžící na straně serveru a je speciálně navržen pro tvorbu interaktivních webových stránek. Počátky programovacího jazyka PHP se datují od roku 1994, kdy jeho autor Rasmus Lerdorf zkombinoval několik skriptů a vytvořil balíček *Personal Home Page Tools* (odtud zkratka PHP). O rok později implementoval skriptovací jádro, které obsahovalo nástroj pro analýzu vstupních dat z formulářů HTML. Tím vzniklo PHP 2.0, avšak tato verze pracovala pouze na operačním systému Linux / Unix. Pozdější verze PHP 3.0 již pracovala na 32bitovém

operačním systému Windows a Macintosh. Od této verze je význam značky PHP vykládán spíše jako *PHP Hypertext Preprocessor*.

PHP běží na straně serveru jako interpret, který zpracovává skripty mezi značkami `<?php` a `?>` a tímto se odlišuje např. od JavaScriptu, jehož skripty se stahují přímo s HTML stránkou a jsou vykonány na straně klientského prohlížeče.

Některé důležité znaky skriptovacího jazyka PHP:

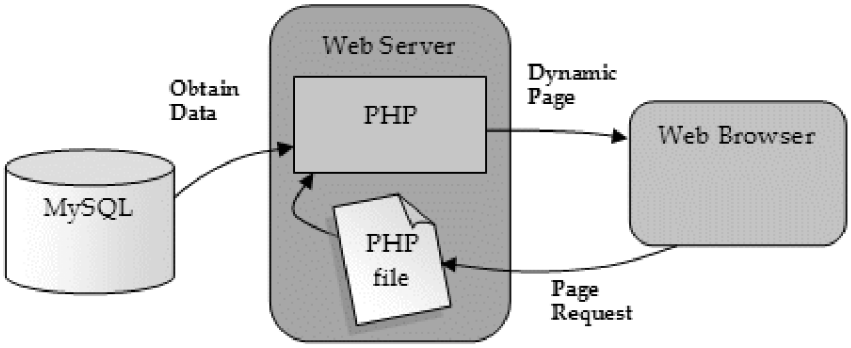
- Jednoduchost, rychlost a stabilita
- Má syntaxi velmi podobnou jazyku C a je tedy většině vývojářů dosti blízký
- Těsná integrace s většinou dostupných databázových systémů
- Nezávislost na platformě
- Podporuje jej velké množství webhostingových služeb
- Můžeme aktualizovat obsah stránek na serveru jen tím, že změníme data v databázi a ne přepisováním každé stránky

3.6 MySQL

MySQL (*My Structured Query Language*) je relační databáze typu DBMS (DataBase Management System). Jedná se o otevřený systém pro správu relačních databází a patří spíše k jednodušším databázovým systémům, avšak v současné verzi podporuje většinu operací, které nabízejí robustnější komerční produkty. Je k dispozici pod licencí Open Source nebo za nízkou cenu pod komerční licencí. Díky své licenci a rychlosti patří v poslední době mezi ty nejoblíbenější databázové systémy. Vychází z deklarativního dotazovacího jazyka SQL (*Structured Query Language*), což je celosvětově používaný standardní dotazovací jazyk pro databáze. Do MySQL lze ukládat různá data (např. texty, obrázky, atd.) s nimiž lze dále jednoduše pracovat (hledat, filtrovat, třídit, měnit apod.). Běží jako samostatný server a nejčastěji se MySQL používá ve spojení se skriptovacím jazykem PHP nebo ASP, který umožňuje přístup k uloženým datům.

3.7 Kombinace PHP s MySQL

Na obrázku 1 je znázorněna spolupráce webového serveru, skriptovacího stroje PHP a databáze MySQL.



Obrázek 1: Kombinace webového serveru PHP a MySQL (převzato z http://www.databasejournal.com)

Běžná žádost o informace z databáze se skládá z několika kroků:

1. Klientský prohlížeč (Web Browser) pošle požadavek (Page Request) pomocí protokolu HTTP na konkrétní stránku s příponou .php
2. Webový server (Web Server) přijme žádost o soubor a pošle jej pro zpracování PHP preprocesoru
3. Preprocesor PHP provádí skript a uvnitř skriptu narazí na příkaz pro připojení k databázi a provedení dotazu. PHP vytvoří spojení s databází MySQL a pošle příkaz v podobě SQL dotazu
4. Databáze MySQL přijme dotaz, zpracuje ho a odešle výsledek
5. PHP případně vygeneruje výsledek dotazu do HTML a dokončí zpracování skriptu. Výsledné HTML odešle webovému serveru
6. Webový server odešle výsledný dokument doplněný o informace z databáze zpátky prohlížeči

3.8 Další technologie pro vývoj webových aplikací

ASP – Technologie ASP (*Active Server Pages*) byla vytvořena a přestavena společností Microsoft. Ve skutečnosti je velmi podobná PHP, rovněž obsahuje dostatek různých funkcí, mezi ty nejdůležitější patří funkce spolupracující s databázovými servery. Při návrhu se již od začátku drží objektového modelu. ASP dnes společně s PHP patří k nejrozšířenějším technologiím, dle mého názoru však nečeká ASP, na rozdíl od PHP, další masivní rozmach. ASP trpí některými nedostatky, které mu zabraňují ve větším rozšíření. Např. to, že tato technologie není na rozdíl od PHP nezávislá na platformě a je ostře vázána na Win/IIS. Dále také nároky na hardware a pořízení serverového OS

není zrovna levnou záležitostí. A v neposlední řadě vyšší režie při spuštění/překladu – stránka se musí po provedených změnách překompilovat.

JSP – Jazyk JSP (*Java server pages*) je skriptovací jazyk pracující na straně serveru a patří mezi uvedené PHP a ASP. JSP je dílem firmy SUN. Pracuje na technologii Java 2 Enterprise Edition. Jeho výhodou je nezávislost na platformě počítače. Naopak nevýhodou je, rovněž jako u ASP, potřeba speciálního serveru. Stránka zpracovávaná technologií JSP je nejprve konvertována do podoby servletu (program generující výsledný HTML kód), ten je dále zpracován pomocí běhového prostředí Javy (*Java Virtual Machine*) a výsledek je předán zpět serveru.

V současné době lze nejlépe hodnotit jazyk PHP, který svým výkonem překonává jak ASP, tak i JSP aplikace. Hlavní výhodou je jeho přenositelnost, modulárnost, menší hardwarové nároky na pořízení a zprovoznění internetového serveru a v neposlední řadě jeho relativní jednoduchost pro menší a středně velké projekty.

4 Analýza a návrh řešení

4.1 Struktura aplikace

Struktura celé aplikace se skládá z několika souborů. Převážnou většinu tvoří skripty v jazyce PHP, dále skript v jazyce SQL, CSS dokument definující rozložení a vzhled aplikace. Mezi ty stěžejní pro chod aplikace patří následující:

- skript v jazyce PHP, který počítá přístupy, zjišťuje údaje o návštěvnících a zapisuje do databáze, je uveden jako zdroj volání obrázku, který v závěrečné fázi generuje do prohlížeče
- skript v jazyce PHP, který vyhodnocuje údaje uložené v databázi a výsledky zobrazí formou statistik uživateli
- skripty v jazyce PHP, které implementují funkci odhadu návštěvnosti v následujícím období z dříve získaných dat
- skript v jazyce SQL pro vytvoření struktury databáze včetně naplnění některých tabulek daty, která budou později použita při porovnávání
- několik dalších souborů (skripty PHP, dokument CSS, obrázky) potřebných především pro správnou funkčnost webového uživatelského rozhraní kooperujícího s databází MySQL, ale zároveň také definujících jeho vzhled a rozvržení jednotlivých prvků HTML

4.2 Přehled často používaných funkcí jazyka PHP a SQL

4.2.1 PHP

Při vytváření webového rozhraní a skriptů jazyka PHP důležitých pro správný chod aplikace jsem často používal níže uvedené funkce.

Funkce pro práci s časem a kalendářem

`date(formát[,časové_razítko])` – Formátuje místní datum a čas. Vrátí jej podle formátovaného řetězce s použitím časového razítka nebo aktuální není-li časové razítko nastaveno

`mktime([hodiny,minuty,sekundy,měsíc,den,rok])` – Vrátí aktuální unixové časové razítko pro datum a čas, nebo aktuální datum a čas

Funkce pro práci s MySQL

`mysql_connect([pc[:port][:cesta/k/soketu], uživatel,heslo])` – Vytvoří spojení se serverem MySQL

`mysql_close([id_odkazu])` – Uzavře spojení

`mysql_query(dotaz,[id_odkazu])` – Odešle dotaz na server MySQL

`mysql_fetch_array(výsledek[,typ])` – Načte výsledný řádek do asociativního nebo číselného pole či do obou druhů polí

`mysql_insert_id([id_odkazu])` – Vrátí generovanou hodnotu ID posledního INSERT příkazu

`mysql_affected_rows([id_odkazu])` – Vrátí počet ovlivněných záznamů po posledním dotazu

`mysql_num_rows(výsledek)` – Vrátí počet záznamů ve výsledku

Funkce pro práci s uživatelskými relacemi

`session_start()` – Zahájí novou uživatelskou relaci

`session_register(název,[proměnná])` – Registruje proměnnou v aktivní uživatelské relaci

`session_destroy()` – Vymaže veškerá data dané uživatelské relace

Funkce pro práci s řetězci

`addslashes(řetězec)` – Vrátí řetězec se zpětnými lomítky před znaky, které by mohly být problematické v databázových dotazech

`echo(řetězec)` – Vypíše jeden nebo více řetězců

`eregi(výraz,řetězec[,pole])` – Prohledá řetězec a vyhledá v něm výraz, nerozlišuje velká a malá písmena

`eregi_replace(výraz1,výraz2,řetězec)` – V daném řetězci vyhledá a nahradí výraz výraz1 výrazem výraz2, nerozlišuje velikost písmen

`explode(separator, řetězec[,limit])` – Rozdělí řetězec na hranicích tvořených řetězcem separator, vrátí pole řetězců, z nichž každý je částí argumentu řetězec

`split(výraz, řetězec[,limit])` – Rozdělí daný řetězec podle zadaného výrazu

`str_replace(starý_výraz,nový_výraz,řetězec[,počet])` – Nahradí všechny výskyty jednoho řetězce dalším řetězcem

`strtolower(řetězec)` – Převede řetězec na malá písmena

`trim(řetězec)` – Odstraní bílé znaky ze začátku a konce řetězce

4.2.2 SQL

Při vytváření dotazů na databázi MySQL jsem často užil níže uvedených funkcí.

Příkazy pro manipulaci s daty

`SELECT` – Vybírá data z databáze, umožňuje výběr podmnožiny, řazení, seskupování dat

`INSERT` – Vkládá do databázové tabulky nová data

`UPDATE` – Edituje data v databázi

`DELETE` – Odstraňuje záznamy z databáze

Komparační funkce

`REGEXP` – Klauzule pro vyhledávání vzoru v řetězci podle příslušného regulárního výrazu

Agregační funkce

`MAX(sloupec)`-Vrátí největší hodnotu ve vybraném sloupci

`SUM(sloupec)`-Vrátí součet hodnot ve sloupci

`COUNT(sloupec)` – Vrátí počet řádků v tabulce

`DISTINCT(sloupec)` – Vrátí počet unikátních řádků v tabulce

Funkce pro práci s kalendářem

Funkce jsem použil hlavně při zobrazování statistik při omezení pro určité řádky tabulky a práci s kalendářem.

`hour(datum)` – Funkce vrátí hodinu dne 0–24

`weekday(datum)` – Funkce vrátí den v týdnu 0 = pondělí ... 6 = neděle

`dayofmonth(datum)` – Funkce vrátí den v měsíci 1–31

`month(datum)` – Funkce vrátí číslo měsíce v roce 1–12

`year(datum)` – Funkce vrátí rok

4.3 Získání a zpracování dat z přístupů

Systém statistik získává informace o přístupech pomocí vloženého objektu do sledovaných stránek. Tímto objektem je většinou obrázek a ve spojení s jazykem HTML a JavaScript dochází k vzdálenému volání skriptu umístěnému na monitorovacím serveru a který je uveden jako zdroj obrázku. Tento skript využívá implementovaných funkcí jazyka PHP pro zjištění informací o návštěvníkovi.

Nyní se stručněji v několika bodech zmíním o tom, co všechno monitorovací skript zjišťuje a vyhodnocuje ve stanoveném pořadí:

- datum a čas přístupu
- unikátní přístup nebo shlédnutí

- IP adresa počítače
- typ prohlížeče včetně možné nadstavby
- operační systém
- rozlišení a barevná hloubka monitoru
- odkud návštěvník přišel na sledovanou stránku
- převod IP na DNS záznam a doména nejvyšší úrovně

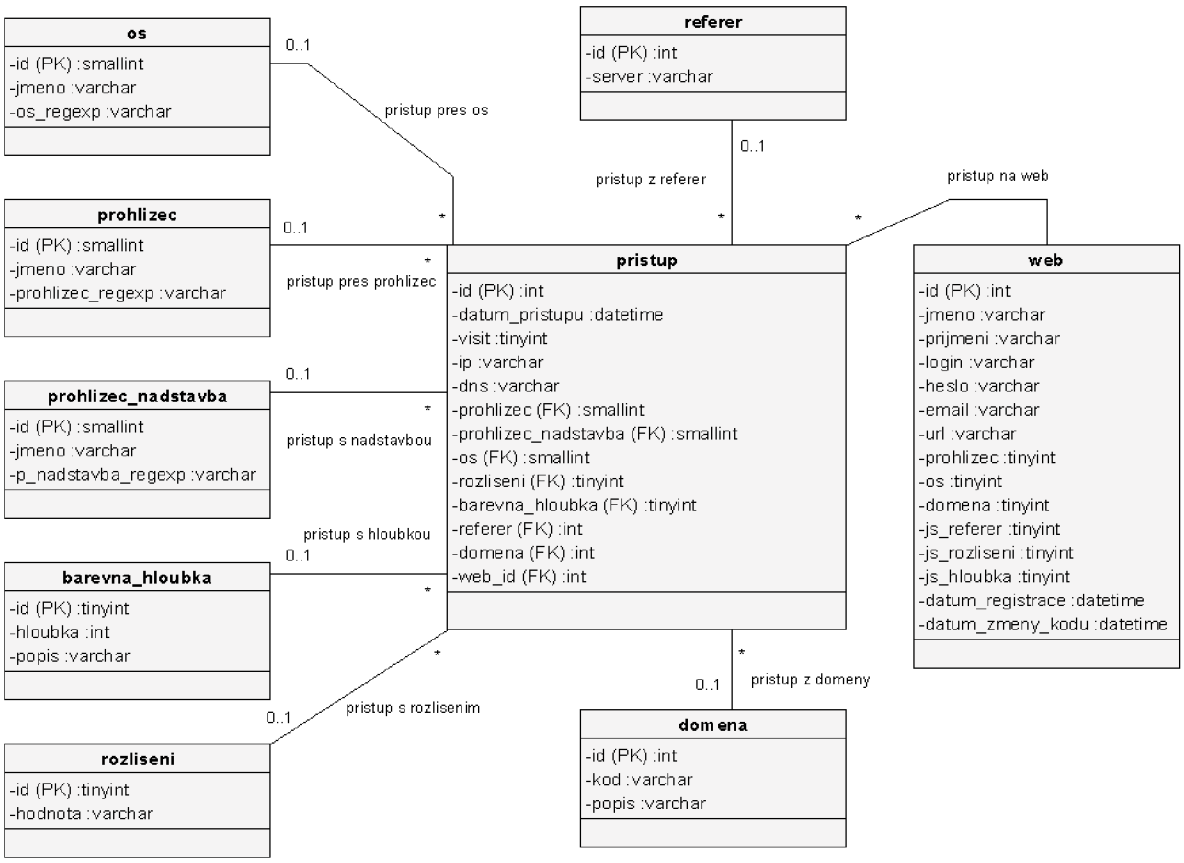
Zjišťování jednotlivých údajů je navíc ovlivněno uživatelem, který má možnost pomocí formuláře zvolit si co chce monitorovat.

4.4 Konceptuální model

Konceptuální model umožňuje identifikovat *entitní typy* a vazby mezi *entitními typy*, které je nutné v systému evidovat. Entitou rozumíme objekt z reálného světa, jehož vlastnosti (*hodnoty atributů*) chceme evidovat. Entitní typ je tedy popsán jménem a množinou *atributů*, entita je konkrétní výskyt (instance) entitního typu s hodnotami atributů. Mezi atributy rozlišujeme tzv. *primární klíč*, což je podmnožina množiny atributů, dle jejichž hodnot je entita odlišitelná od ostatních entit stejného typu.

Vazba modeluje vztah mezi *n* entitními typy a je definována názvem a množinou entitních typů, které jsou v příslušném vztahu.

Konceptuální model je znázorněn pomocí jazyka UML (*Unified Modeling Language*), který je souhrnem především grafických notací k vyjádření analytických a návrhových modelů. UML je také nástroj pro stavbu, vizualizaci, specifikaci a dokumentaci softwarových projektů. Pomocí tohoto jazyka byl vytvořen model struktury databáze reprezentovaný ER diagramem (obrázek 2). Obsahuje celkem 9 tabulek nezbytných pro správný chod aplikace:



Obrázek 2: ER diagram

4.5 Statistiky přístupu

V této kapitole se budu věnovat jednomu ze stěžejních úkolů aplikace, který představuje způsob, jakým lze získat užitečná data pro nejrůznější statistiky a jak tato data dále zpracovat.

Nyní se stručněji v několika bodech zmíním o tom, co všechno statistika přístupu zjišťuje a vyhodnocuje:

- počet unikátních přístupů na sledované stránky (unikátní návštěvník / den)
- celkový počet zobrazení stránek (shlédnutí)
- počet unikátních IP adres
- datum a čas posledního přístupu na sledované stránky
- hodinovou návštěvnost
- denní návštěvnost
- prohlížeče a jejich nadstavby
- operační systémy
- rozlišení pracovní plochy a barevnou hloubku monitorů

- domény nejvyšší úrovně (TLD)
- z jakých stránek návštěvníci na sledované stránky přišli

4.5.1 Struktura tabulky s přístupy na sledované stránky

Ze schématu databáze (ER diagramu) vyplývá, že nejpodstatnější tabulkou databáze je tabulka **pristup**, kam se ukládá každý přístup na konkrétní stránky, které jsou uživatelem v rámci aplikace zaregistrovány.

Tabulka **pristup** obsahuje tyto sloupce:

- *id* – unikátní označení každého přístupu (je primárním klíčem tabulky)
- *datum_pristupu* – datum a čas přístupu
- *visit* – identifikace zda se jedná o visit (unikátní návštěvník / den) nebo ne
- *ip* – IP adresa přiřazená síťovému adaptéru počítače návštěvníka
- *dns* – DNS záznam návštěvníka k příslušné IP adrese (pokud existuje)
- *prohlizec* – identifikace prohlížeče (cizí klíč do tabulky *prohlizec*)
- *prohlizec_nadstavba* – identifikace nadstavby prohlížeče (pokud existuje, obsahuje cizí klíč)
- *os* – identifikace operačního systému (cizí klíč do tabulky *os*)
- *rozliseni* – rozlišení pracovní plochy monitoru (cizí klíč do tabulky *rozliseni*)
- *barevna_hloubka* – barevná hloubka monitoru (cizí klíč do tabulky *barevna_hloubka*)
- *referer* – pokud návštěvník přišel z cizího webu než který je sledovaný, obsahuje cizí klíč do tabulky *referer* odpovídající doménovému jménu serveru
- *domena* – pokud má návštěvník ke své IP adrese registrovaný DNS záznam, obsahuje cizí klíč do tabulky *domena* s identifikací domény nejvyšší úrovně (TLD – Top Level Domain)
- *web_id* – identifikace určující kterému ze sledovaných webů daný přístup patří

4.5.2 Identifikace, datum a čas přístupu

Sloupec **id** tabulky **pristup** je primárním klíčem celé tabulky, obsahuje unikátní označení každého přístupu na všechny registrované stránky v rámci aplikace a zajišťuje ho atribut `AUTO_INCREMENT` jazyka SQL.

Sloupec **datum_pristupu** tabulky **pristup** obsahuje datum a čas přístupu ve formátu *RRRR-MM-DD HH:MM:SS* což v SQL zajišťuje datový typ `datetime`.

4.5.3 Problém při zjištění unikátního přístupu za den

Třetím sloupcem tabulky je informace o tom, zda je konkrétní přístup na stránky unikátní či nikoliv. Je zřejmé, že princip funkčnosti systému zjišťování informací o návštěvnících výrazně ovlivňuje výsledky statistik. Navržený systém rozlišuje mezi dvěma druhy přístupu. Jako první je brán unikátní přístup na stránku, který by větší váhou vypovídal o skutečné návštěvnosti, druhý potom jako opakované shlédnutí stránky.

Po analýze, jakým způsobem by šlo získat důvěryhodná data o unikátních přístupech jsem došel k názoru, že je to veliký problém. Kvůli omezenému počtu veřejných IP adres dochází k použití adres z privátního adresového prostoru a služby DHCP u providerů a nemůžeme tudíž využít IP adres, protože jedna IP adresa může reálně představovat i několik desítek návštěvníků sledovaných stránek.

Nicméně existují způsoby, jak lze lépe identifikovat návštěvníky stránek. Jedním z nich je použití cookies pomocí kterých máme možnost uložit si na disku návštěvníka webu určité informace, které mohou být dostupné při dalším volání stránky. Cookies mají ale také své nevýhody. Prvním z nich je skutečnost, že jejich ukládání lze pomocí nastavení prohlížeče zakázat. V takovém případě by bylo ovlivněno zjištění unikátnosti přístupu. Druhou z nich je možnost cookie kdykoliv z obsahu disku smazat.

Jako možné řešení, jak měřit počet unikátních návštěvníků za den, mi připadne kombinace cookies a IP adres. Lze při tom využít možnosti nastavení data platnosti pomocí funkce `setcookie()` jazyka PHP. Při prvním volání skriptu vloženého do sledovaných stránek uložíme cookie, která bude mít mimo jiné nastavený parametr funkce datum vypršení a bude platná pouze dnes. Cookies se do klientského prohlížeče posílají se zbytkem HTTP hlaviček a musejí předcházet jakémukoliv výstupu na stránku. Volání této funkce musí být tedy umístěno před `<html>` či `<head>` tagy. Jakmile je cookie spolu s dalšími hlavičkami odeslána, může být z pohledu skriptu přístupná přes superglobální pole `$_COOKIE` až v dalším načtení téže stránky. V další fázi zjistíme, zda cookie existuje a pokud ano, o visit se nejedná. V opačném případě musíme zjistit, zda návštěvník s IP adresou, která mu náleží, přistoupil na stránky v poslední předem stanovenou dobu či nikoliv. Tuto dobu jsem orientačně stanovil na 30 minut. Pokud se v databázové tabulce **pristup** se všemi přístupy návštěvníka a danou IP adresou po přičtení oněch 30 minut najde záznam který, když porovnáme s aktuálním datem a časem bude vetší, o unikátní přístup (visit) se nejedná. V opačném případě se jedná o visit.

Ani zde bohužel nedospějeme ke spolehlivému výsledku přesně určujícímu počet unikátních návštěvníků za den, i když je, domnívám se, úspěšnost docela vysoká. Je tu stále ještě možnost návštěvníka s vypnutým ukládáním cookies, který navštíví stránky v průběhu dne několikrát za dobu větší než oněch 30 minut od posledního přístupu a vytvoří tak více unikátních přístupů. Naopak

pokud se na sledovaném webu současně pohybuje několik návštěvníků za jednou veřejnou IP adresou, bude započítán pouze jeden unikátní přístup. Typickým příkladem druhého případu je NAT neboli Network Address Translation, který vznikl jako prostředek k úspoře IPv4 adres. Umožňuje, aby lokální síť používala neveřejné adresy, jež pak při komunikaci s okolním světem převádí na veřejné.

4.5.4 Souhrnné statistiky

Po předchozí analýze můžeme zhotovit některé statistické údaje v rámci jednoho konkrétního sledovaného webu. Mezi ně můžeme zařadit:

- počet unikátních návštěvníků za toto období (*visits)* sečtením řádků tabulky kde sloupce *visit* = 1
- počet shlédnutých stránek – tuto hodnotu dostaneme po sečtení všech řádků tabulky *pristup* (primárních klíčů)
- počet unikátních IP adres – získáme použitím agregační funkce SQL
- datum a čas posledního přístupu na sledované stránky

4.5.5 Hodinová a denní návštěvnost

Pro zjištění hodinové a denní návštěvnosti využijeme SQL funkce pro práci s datem a tabulkou **pristup** konkrétně se sloupcem **datum_pristupu** který obsahuje datum a čas konkrétního přístupu. Hodinovou návštěvnost pro konkrétní období zjistíme tak, že v cyklu s pomocí SQL funkce `HOUR()` s parametrem data a času konkrétního přístupu zjistíme počet přístupů v jednotlivých hodinách (0 – 23). Výsledky dotazů pro jednotlivé hodiny si budeme ukládat do pomocného pole, které později využijeme při grafickém znázornění této statistiky.

Podobně jako u hodinové budeme postupovat při zjištění denní statistiky. Zde se nám nabízí SQL funkce `WEEKDAY()`, které když předáme jako parametr datum (sloupec datového typu `datetime`), vrátí nám číslo dne v týdnu. V cyklu potom přes všechny dny v týdnu (0 – 6, Pondělí – Neděle) projdeme všechny zaznamenané přístupy pro zadané období, ve kterém budeme chtít statistiku vypočítat. Výsledky budeme opět průběžně ukládat do pomocného pole, které spolu s dalšími statistikami graficky znázorníme.

4.5.6 Detekce rozlišení a barevné hloubky monitoru

Jelikož nejsme schopni na straně monitorovacího serveru a skriptu PHP tyto údaje o návštěvníkovi zjistit použijeme možnosti klientského JavaScriptu bez kterého by to bylo prakticky nemožné.

Pokud to uživatel při registraci či změně monitorovacího kódu vyžaduje, vložíme kód (obrázek) mezi HTML značky JavaScriptu a jako zdroj obrázku uvedeme odkaz (zdroj) na náš skript, který počítá přístupy, zjišťuje údaje a zapisuje do databáze MySQL doplněných o několik parametrů předaných metodou GET v URL požadavku. Z parametrů předaných skriptu bude vždy jeden povinný, tím bude identifikace vloženého kódu jednoznačně svázaného se zaregistrovaným webem a několik volitelných, ve kterých využijeme možnosti JavaScriptu. Důležité je že každý obrázek je při načítání stránky webovým prohlížečem získán pomocí samostatného HTTP requestu a tím máme možnost analyzovat jak parametry předané skriptu, tak i hlavičky HTTP protokolu na straně serveru (skriptu).

Aby byl vložený kód co nejefektivnější a mohli bychom zaznamenat i jiné statistiky a přístupy, musíme myslet i na fakt, že ne každý prohlížeč poskytuje podporu klientského JavaScriptu. V tom případě vložíme kód mezi speciální značky, jejichž obsah je těmito prohlížeči vykonán a naopak prohlížeči s podporou JavaScriptu povinně ignorován.

V tabulce **rozliseni** budou obsaženy jednotlivé druhy rozlišení pracovní plochy monitoru použitelné při porovnání a bude mít následující strukturu:

- *id* – unikátní označení druhu rozlišení (primární klíč tabulky)
- *hodnota* – hodnota příslušného rozlišení složená z možného výsledku hodnot šířky a výšky uložených v objektovém modelu dokumentu (`document.screen.width` a `document.screen.height`)

Barevná hloubka monitoru je pomocí Javascriptu zjistitelná přečtením hodnoty objektu z `document.screen.colorDepth`. V tabulce barevna_hloubka budou uloženy typy barevných hloubek a v podobě sloupců má tuto strukturu:

- *id* – unikátní označení typu barevné hloubky (primární klíč tabulky)
- *hloubka* – hodnota dostupná přes volání funkce JavaScriptu a použitá při porovnání
- *popis* – textový popis příslušné barevné hloubky

4.5.7 Detekce prohlížeče a nadstavby

Díky skutečnosti, že při načítání webové stránky doplněné o obrázky dojde k získání každého obrázku pomocí samostatného HTTP požadavku (request) máme možnost informace o prohlížeči a dále i operačním systému na straně monitorovacího skriptu zjišťovat ze superglobální proměnné dostupné z jazyka PHP `$_SERVER['HTTP_USER_AGENT']`. Tato proměnná obsahuje hlavičku **User-Agent**, ve které prohlížeč posílá informace o své verzi.

Typ prohlížeče můžeme například zjistit pomocí porovnání hlavičky User-Agent s několika regulárními výrazy připravenými v příslušné tabulce.

Regulární výrazy jsou sestaveny z analýzy hlaviček User-Agent a spolu se jmény prohlížečů uloženy v databázové tabulce **prohlizec**. Hlavičky User-Agent lze získat např. ze souboru *browscap.ini* který je součástí instalace PHP. Při ukládání jednotlivých záznamů bychom měli dodržet určité pořadí, protože zdaje v hlavičkách, kterými se představují jednotlivé prohlížeče a operační systémy nejsou zdaleka tak jednoznačné. Při postupném prohledávání tabulky příkazem SQL z databáze vybereme identifikátor toho prohlížeče, jehož regulární výraz jako první vyhoví porovnání s hlavičkou User-Agent.

Struktura tabulky **prohlizec** je následující:

- *id* – unikátní označení každého prohlížeče (je primárním klíčem tabulky)
- *jmeno* – název a verze daného prohlížeče
- *prohlizec_regexp* – regulární výraz použitý při detekci daného prohlížeče

Detekce nadstavby prohlížeče je velmi podobná s detekcí prohlížeče. Budeme rovněž porovnávat obsah hlavičky User-Agent, která bývá informací o nadstavbě někdy rozšířena, s předpřipravenými regulárními výrazy.

Struktura tabulky **prohlizec_nadstavba** je následující:

- *id* – unikátní označení nadstavby prohlížeče (je primárním klíčem tabulky)
- *jmeno* – název nadstavby prohlížeče
- *p_nadstavba_regexp* – regulární výraz použitý při detekci dané nadstavby

4.5.8 Detekce operačního systému

Postup při detekci operačního systému je podobný detekci prohlížeče. Informace o operačním systému bývají rovněž součástí hlavičky **User-Agent** a pro účely porovnání vložíme několik regulárních výrazů do příslušné tabulky.

Struktura tabulky **os** je následující:

- *id* – unikátní označení každého operačního systému (je primárním klíčem tabulky)
- *jmeno* – název a verze operačního systému
- *os_regexp* – regulární výraz použitý při detekci daného operačního systému

4.5.9 Detekce serveru, ze kterého návštěvníci přišli

Při detekci stránek, ze kterých návštěvníci na sledované stránky přicházejí, nelze použít superglobální proměnné `$_SERVER['HTTP_REFERER']` jazyka PHP, jelikož bychom z pohledu monitorovacího skriptu dostali URL sledované stránky a tudíž nutným způsobem je použití JavaScriptu. K parametrům z předchozích kapitol (identifikace webu, detekce rozlišení a barevné hloubky) přidáme parametr *page* (proměnnou) který bude obsahovat URL stránky ze které návštěvník přišel na sledovanou stránku. Tuto vlastnost objektového modelu prohlížeče v jazyce JavaScript najdeme v `top.document.referrer`.

V aplikaci budeme rozeznávat celkem tři možné způsoby příchodu návštěvníka na sledované stránky:

- přímé volání
- přechod v rámci sledovaného webu
- příchod z cizího webu (serveru)

Pokud je detekce serveru při registraci stránek uživatelem povolena a proměnná *page*, kterou předáme skriptu jako parametr je prázdná, jedná se o *přímé volání*, což znamená, že návštěvník zadal konkrétní URL sledovaných stránek přímo do adresového řádku webového prohlížeče. Dále je možné rozeznat, zda se nejedná o *přechod v rámci sledovaného webu* a pokud se neuplatní ani jedna z těchto podmínek, uplatní se poslední alternativa, kterou je skutečnost, že návštěvník přišel z *cizího webu (serveru)*.

V případě, že parametr *page* obsahuje řetězec s URL adresou cizího webu, budeme v databázové tabulce **referer** zaznamenávat jen doménové adresy stránek – tedy první a druhou úroveň domény (případně i třetí pokud jí nebude 'www'). Z řetězce odstraníme 'www.' jelikož by bylo v databázi nepotřebné a v případě že server zatím neexistuje, vložíme jej do tabulky jako nový záznam.

Struktura tabulky **referer** je následující:

- *id* – unikátní označení každého serveru (je primárním klíčem tabulky)
- *server* – adresa (doménové jméno cizího serveru) ze kterého návštěvník přišel

4.5.10 Detekce IP adresy a DNS záznamu

Do tabulky **pristup** budeme ukládat jak IP adresu počítače, tak jeho doménovou adresu (pokud existuje k této adrese záznam). Musíme ale počítat s tím, že na cestě mezi klientem (návštěvníkem) a serverem (monitorovacím skriptem) v Internetu se může vyskytovat proxy server, který dělá prostředníka mezi jejich komunikací. IP adresa, kterou obsahuje superglobální proměnná `$_SERVER['REMOTE_ADDR']` tedy může být IP adresou proxy serveru a v tom případě IP adresu cílové stanice návštěvníka obsahuje proměnná `$_SERVER['HTTP_X_FORWARDED_FOR']`. V

monitorovacím skriptu statistiky přístupu se tedy nejprve otestuje, zda byla proměnná `$_SERVER['HTTP_X_FORWARDED_FOR']` nastavena a podle výsledku nastavíme jednu z proměnných v jednom ze sloupců tabulky **pristup** jako správnou IP adresu počítače návštěvníka.

Doménovou adresu (podmínkou je, že cílová stanice má ke své IP adrese DNS záznam) se pokusíme zjistit z proměnné `$_SERVER['REMOTE_HOST']`, pokud bychom obdrželi prázdný řetězec (proměnná je prázdná), tak se pokusíme o zjištění použitím funkce `gethostbyaddr()`, která převádí IP adresu na adresu doménovou. Výsledek této funkce uložíme do nějaké řetězcové proměnné, kterou následně otestujeme, zda obsahuje hodnotu (není prázdná). Pokud neobsahuje hodnotu, IP adresu se nepodařilo převést na doménovou adresu. Podle výsledku jsme k dané IP adrese buď našli adresu doménovou, nebo ne.

4.5.11 Detekce domény nejvyšší úrovně (TLD)

Struktura tabulky **domena** je následující:

- *id* – unikátní označení každé domény (je primárním klíčem tabulky)
- *kod* – kód domén první (nejvyšší – TLD) úrovně použitý při porovnávání
- *popis* – slovní popis domény, rozlišujeme národní (název státu, např.: .uk, .cz) a generické (např.: .com, .org)

Doménu nejvyšší úrovně můžeme zjistit pouze tehdy, má-li návštěvník ke své IP adrese DNS záznam a to takovým způsobem, že doménovou adresu rozdělíme na části oddělené tečkou a ta poslední část je námi hledaná doména. Jednoduchým SQL dotazem poté porovnáváme získanou doménu s doménami uloženými v databázi, při shodě vytáhneme z tabulky identifikátor, který bude spolu s dalšími použit v dotazu při vkládání údajů do tabulky **pristup**.

4.5.12 Zápis údajů do databáze

Při registraci stránek ve webovém rozhraní je uživateli nabídnuto na výběr z několika variant, které by si mohl přát o návštěvnících monitorovat. Tato konfigurace je vytvořena při registraci či pozdější případné změně monitorovacího kódu a pomocí SQL dotazu vložena do tabulky **web**. Konfiguraci lze zjistit v každém přístupu na sledované stránky, je svázána s identifikátorem **web_id** registrovaného webu, který je v podobě parametru předán monitorovacímu skriptu.

Tím lze skriptu říci, jaké statistiky má zpracovávat a které případně vynechat z toho hlediska, že si je uživatel nepřeje monitorovat a nebude si je moci tudíž ani zobrazit. V závěrečné fázi monitorovacího skriptu se zpracované údaje o návštěvě uloží do tabulky **pristup** obsahující přístupy na všechny registrované weby v aplikaci a vygeneruje se obrázek určitých rozměrů (např. obrázek o velikosti 1x1 pixel formátu PNG s transparentním pozadím), jelikož je skript volán ze sledovaných stránek jako zdroj obrázku a jeho výstup je nastaven příslušnou hlavičkou `header('Content-`

`type: image/png'`). Výhodou je, že obrázek o takových rozměrech po vložení nijak nenaruší uspořádání stránky a pokud je navíc možnost zobrazení transparentního obrázku prohlížečem podporována, tak ani její vzhled.

4.6 Odhady návštěvnosti

Jako jeden z požadavků kladených na aplikaci bylo zaměřit se a navrhnout funkci, která by odhadovala návštěvnost v následujícím období.

4.6.1 Analýza dostupných metod

Odhady návštěvnosti v následujícím období souvisí s teorií časových řad užívaných např. ve statistice, a proto bych uvedl některé aspekty, které časové řady vykazují. Analýza časových řad se zabývá jejich dekompozicí s cílem rozložit časovou řadu na „základní složky“:

- *Trend* – odpovídá hlavním tendencím dlouhodobého vývoje statistického ukazatele, který časová řada popisuje
- *Sezónní složka* – odpovídá periodicky se opakujícím odchylkám od trendu, ke kterým pravidelně dochází v rámci nějakého intervalu (např. 24 hodin dne, 7 dní v týdnu)
- *Cyklická složka* – je nejspornější částí časové řady, odpovídá dlouhodobým často nepravidelným cyklům s proměnlivou periodou
- *Náhodná složka* – jde o náhodné pohyby bez systematického charakteru, zahrnuje také chyby měření a chyby ze zaokrouhlování při výpočtech

Při dekompozici časové řady by se nejdříve měl identifikovat trend a potom teprve sezónní složka. K identifikaci trendu se používají především 4 metody:

- Naivní modely
- Proložení matematickou křivkou
- Vyrovnání metodou klouzavých průměrů
- Exponenciální vyrovnání

Naivní modely poskytují jen velmi primitivní pohled na vývoj sledovaného ukazatele. Prokládání časových řad zvolenou matematickou křivkou je souhrnně nazýváno neadaptivními metodami, metoda klouzavých průměrů a metoda exponenciálního vyrovnání metodami adaptivními.

Neadaptivní metody jsou takové metody, které časovou řadu vysvětlí jako celek pomocí několika v čase konstantních parametrů. Takový model se jen velmi pomalu (nebo vůbec) přizpůsobuje změnám v charakteru časové řady.

Adaptivní metody se naopak přizpůsobují změnám v charakteru analyzované veličiny poměrně rychle. Je to způsobeno jejich charakterem.

4.6.2 Metoda použitá v systému a její použití ve výpočtu

Po zhodnocení analýzy jsem dospěl k názoru, že ve svých výpočtech použiji některou z adaptivních metod, která počítá s tím, že se trend v námi pozorované časové řadě mění a není tedy možné rozumně jej vyrovnat na celé této časové ose jednou matematickou křivkou.

Zvolil jsem proto metodu Lineárního exponenciálního vyrovnání (*Linear Exponential Smoothing – LES*), která se mi jeví jako vhodná pro implementaci v aplikaci a zároveň z mého pohledu vykazuje docela rozumné výsledky.

Nyní popíšu metodu LES trochu podrobněji včetně zařazení do aplikace a pokusím se vysvětlit, jak při výpočtech pracuje včetně její demonstrace na vhodně zvoleném příkladu.

Základem pro správný výpočet odhadu návštěvnosti v následujícím období je dostačující vzorek naměřených dat, tedy počet unikátních (věrohodnější) nebo opakovaných návštěv konkrétního sledovaného webu za předešlé období.

V aplikaci jsou v době psaní této práce implementovány v podobě skriptů jazyka PHP následující odhady návštěvnosti:

- Odhad hodinové návštěvnosti v následujícím období
- Odhad denní návštěvnosti v následujícím období

Postupy pro výpočet těchto odhadů již zmíněnou metodou jsou implementovány ve zvláštních souborech (skriptech jazyka PHP *LES_hodinova.php* a *LES_denni.php*) a jsou v případě potřeby volány hlavním skriptem *odhady.php*. Postup výpočtu zmíněnou metodou bylo nutné vhodně dekomponovat v podobě několika funkcí s parametry a jejich postupného volání, jelikož na sebe jednotlivé části výpočtu úzce navazují. Na webové stránce aplikace, v sekci zabývající se odhady návštěvnosti má uživatel z dostupného formuláře možnost zadat jaký odhad návštěvnosti si přeje vypracovat včetně povinných parametrů, kterými jsou intervaly zdrojových dat použitých při výpočtu a cílových dat, která budou po výpočtu metodou odhadnuta.

Proces odhadu návštěvnosti se skládá z těchto kroků:

1. Výběr zdrojových dat z databáze (tabulky *pristup*) za uplynulé předem zvolené období včetně jejich uložení do pomocného pole
2. Aplikace funkce pro výpočet klouzavých průměrů na zdrojová data. Klouzavé průměrování se nejčastěji používá k sezónnímu očištění řady, vyhlazují prudké výkyvy a identifikují trend. Jejich nevýhodou je vycházení z minulosti a tedy jisté zpoždění za aktuálním vývojem. V případě že je klouzavý průměr centrován platí při délce 2m + 1 vztah:

   $$Y(t)=\frac{1}{(2\mathrm{m}+1)}*(Y_{t-m}+Y_{t-(m-1)}+Y_{t+m})$$

3. Určení poměru mezi zdrojovými daty (přístupy) v časovou jednotku (hodina, den) a vypočteným klouzavým průměrem k příslušné časové jednotce. Tím získáme odhad sezónní nepravidelné složky
4. Výpočet sezónního indexu, který vznikne zprůměrováním sezónní nepravidelné složky z bodu č. 3 vždy stejné časové jednotky (např. zprůměrování všech dat pro všechny pondělky), který se dále normuje pro zvýšení přesnosti
5. Aplikace funkce pro výpočet dat očištěných o sezónnost. K výpočtu se použijí zdrojová data, která se vydělí příslušným normovaným sezónním indexem
6. Poté se z dat sezónně očištěných vypočítá předpověď zmíněnou metodou. Vstupem pro výpočet nové hodnoty jsou sezónně očištěná data Y, chyba způsobená metodou e (vypočtena jako rozdíl sezónně očištěných dat a předpovědi) a parametr $alfa$. Parametr `alfa` se přibližně odhaduje ze vzorce $alfa=2/(N+1)$, kde N je počet časových jednotek intervalu – prvek sezónnosti. Pro přesnější odhad metody se však doporučuje parametr *alfa* určit experimentálně ze vzniklé chyby metody podle vzorce pro výpočet střední kvadratické odchylky RMSE (Root Mean Square Error) ve tvaru: $RMSE=\sqrt{ROZPTYL(errors)+(PRUMER(errors))^2}$.

   Rozptyl (též střední kvadratická odchylka) spočítáme pomocí vzorce:

   $o^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-E(x))^2$ kde n je počet prvků pole, x_i jsou hodnoty prvků pole a $E(x)$ je střední hodnota prvků pole. Výpočet nové hodnoty v čase t závisí na hodnotách ze dvou předchozích vstupů v čase $t-1$ a $t-2$:

   $$LES(t)=2\mathrm{Y}(t-1)-Y(t-2)-2(1-alfa)e(t-1)+(1-alfa)^2 e(t-2)$$

   Funkci pro výpočet předpovědi jsem tedy několikrát propočítal pro různý parametr *alfa* v okolí hodnoty $alfa=2/(N+1)$ a pro konečný výpočet předpovědi jsem použil takovou hodnotu parametru *alfa*, jehož střední kvadratická odchylka RMSE chyby způsobené metodou byla nejmenší

7. V poslední fázi výpočtu jsou sezónně očištěné předpovědi obohaceny o sezónnost jejich vynásobením s normovaným sezónním indexem, čímž získáme předpověď pro následující období. Nakonec získané výsledky výpočtu zobrazíme uživateli

4.6.2.1 Demonstrace metody na příkladu

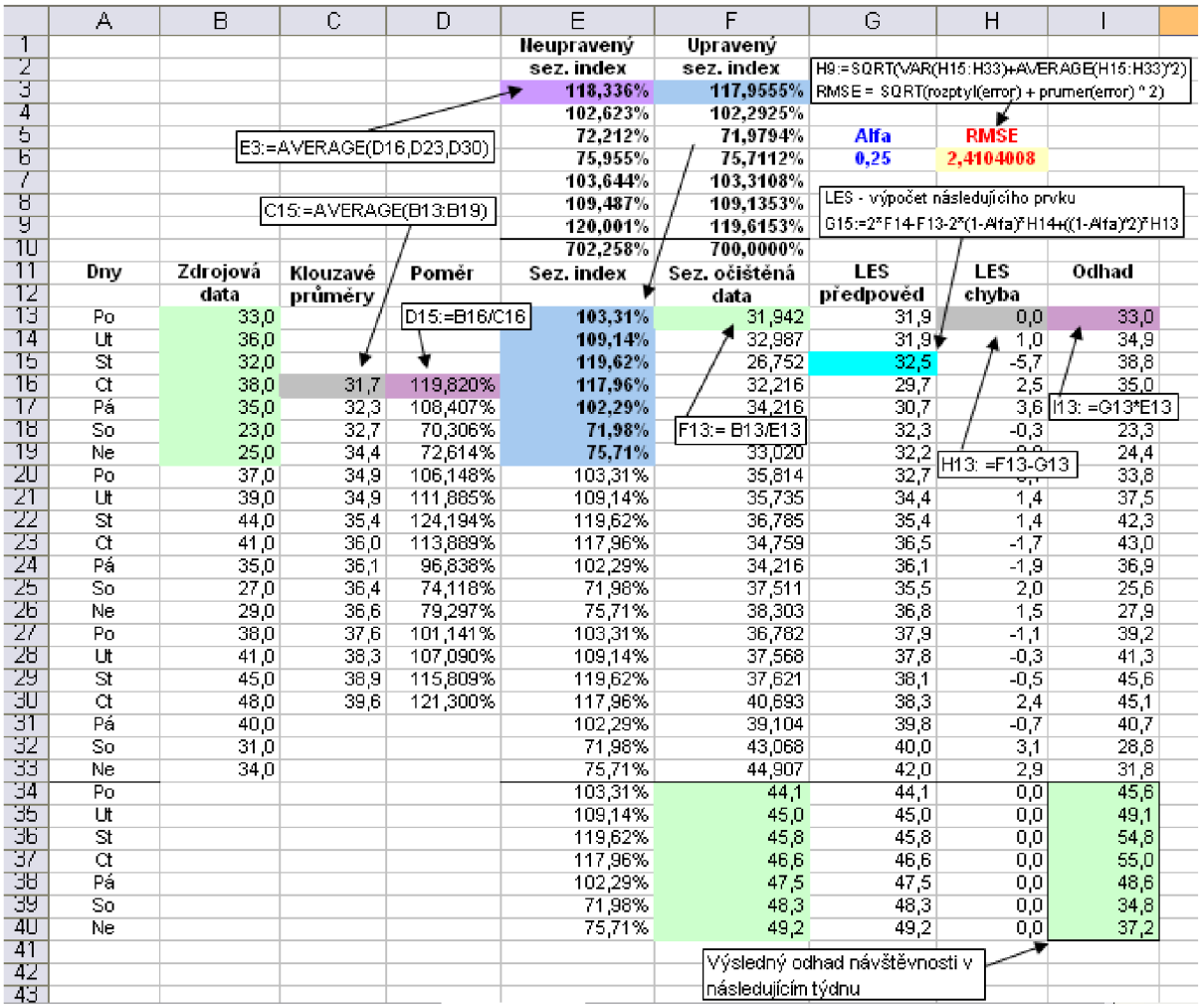
Přihlášený uživatel si přeje vypracovat odhad denní návštěvnosti pro následující období. Zvolí tedy pomocí formuláře příslušnou statistiku, včetně intervalu zdrojových a cílových dat. Dejme tomu, že zvolí vypracování odhadu denní návštěvnosti. Sezónní složce v rámci tohoto intervalu odpovídá 7 dní v týdnu a v tomto intervalu může pravidelně docházet k periodickým odchylkám od trendu (návštěvnost sledovaného webu během týdne je například odlišná od návštěvnosti přes víkend). Je tedy nutné s touto sezónní složkou ve výpočtu počítat.

Jako zdrojová data nutná pro výpočet metody v tomto případu uživatel zadal data o přístupech z předchozích tří týdnů. Čím větší je zvolen interval zdrojových dat, tím je metoda schopna lépe předpovědět trend, který odpovídá dlouhodobějšímu vývoji návštěvnosti. Za cílový interval zvolil následujících 7 dní (počítáno ode dneška, protože dnešní konečná návštěvnost není doposud známa a je také odhadnuta, stejně je tomu tak i u ostatních statistik).

V tabulce 1 jsou informace o zaznamenaných přístupech včetně upraveného výsledku odhadu. Na obrázku 3 jsou z dat o přístupech detailně propočítány jednotlivé fáze výpočtu s pomocí nástroje MS Excel. Z obrázku lze vyčíst jak použité vzorce ve výpočtu, tak i hodnoty po dosažení do vzorců. Na obrázcích 4 a 5 je znázorněno několik spojnicových grafů, jimiž se některé fáze výpočtu ubírají. Postup výpočtu jsem již naznačil v kapitole 4.6.2 a tak se na některé jeho fáze odkáži z legendy grafu.

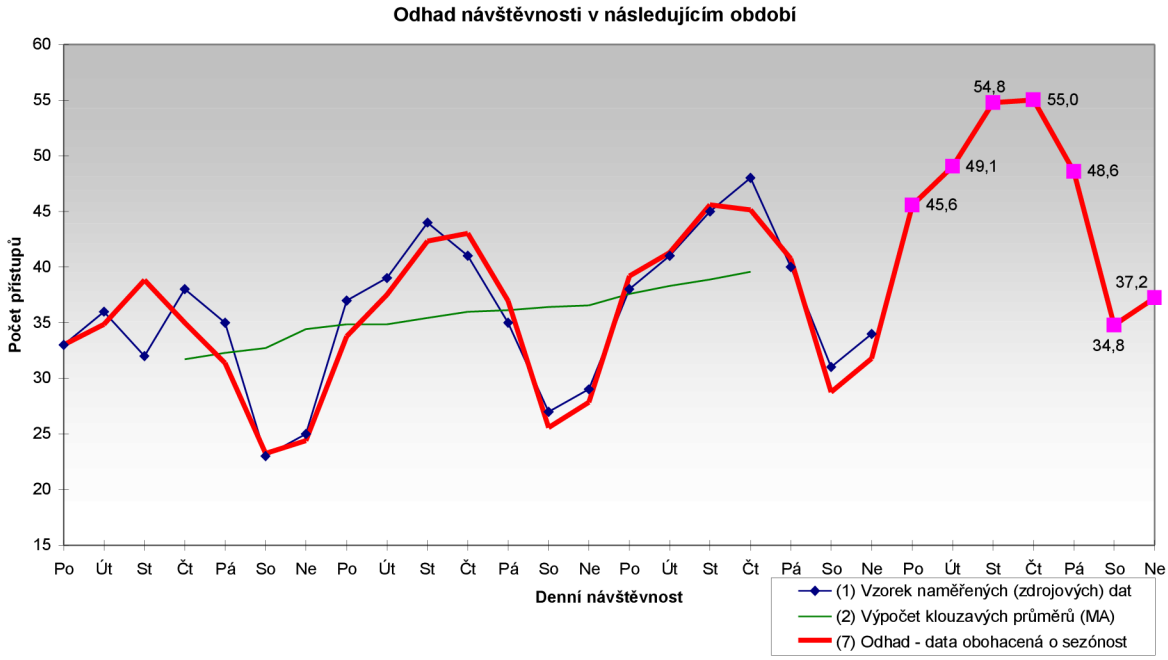
Data o návštěvnosti stránek							
	Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
1. týden	33	36	32	38	35	23	25
2. týden	37	39	44	41	35	27	29
3. týden	38	41	45	48	40	31	34
Předběžný odhad návštěvnosti							
1. týden	46	49	55	55	49	35	37

Tabulka 1: Odhad denní návštěvnosti - zdrojová data o návštěvnosti a odhad v následujícím týdnu

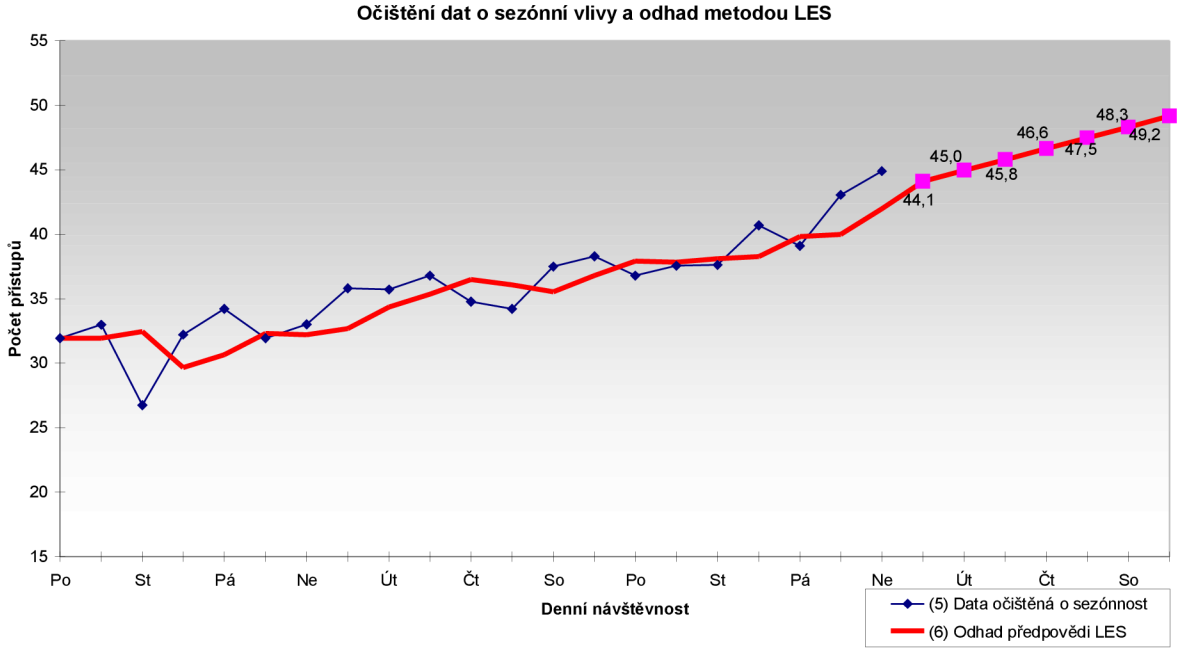


	A	B	C	D	E	F	G	H	I
1					Neupravený	Upravený			
2					sez. index	sez. index			
3					118,336%	117,9555%			
4					102,623%	102,2925%			
5					72,212%	71,9794%	Alfa	RMSE	
6					75,955%	75,7112%	0,25	2,4104008	
7					103,644%	103,3108%			
8					109,487%	109,1353%			
9					120,001%	119,6153%			
10					702,258%	700,0000%			
11	Dny	Zdrojová	Klouzavé	Poměr	Sez. index	Sez. očištěná	LES	LES	Odhad
12		data	průměry			data	předpověď	chyba	
13	Po	33,0			103,31%	31,942	31,9	0,0	33,0
14	Út	36,0			109,14%	32,987	31,9	1,0	34,9
15	St	32,0			119,62%	26,752	32,5	-5,7	38,8
16	Čt	38,0	31,7	119,820%	117,96%	32,216	29,7	2,5	35,0
17	Pá	35,0	32,3	108,407%	102,29%	34,216	30,7	3,6	33,4
18	So	23,0	32,7	70,306%	71,98%	31,950	32,3	-0,3	23,3
19	Ne	25,0	34,4	72,614%	75,71%	33,020	32,2	0,8	24,4
20	Po	37,0	34,9	106,148%	103,31%	35,814	32,7	3,1	33,8
21	Út	39,0	34,9	111,885%	109,14%	35,735	34,4	1,4	37,5
22	St	44,0	35,4	124,194%	119,62%	36,785	35,4	1,4	42,3
23	Čt	41,0	36,0	113,889%	117,96%	34,759	36,5	-1,7	43,0
24	Pá	35,0	36,1	96,838%	102,29%	34,216	36,1	-1,9	36,9
25	So	27,0	36,4	74,118%	71,98%	37,511	35,5	2,0	25,6
26	Ne	29,0	36,6	79,297%	75,71%	38,303	36,8	1,5	27,9
27	Po	38,0	37,6	101,141%	103,31%	36,782	37,9	-1,1	39,2
28	Út	41,0	38,3	107,090%	109,14%	37,568	37,8	-0,3	41,3
29	St	45,0	38,9	115,809%	119,62%	37,621	38,1	-0,5	45,6
30	Čt	48,0	39,6	121,300%	117,96%	40,693	38,3	2,4	45,1
31	Pá	40,0			102,29%	39,104	39,8	-0,7	40,7
32	So	31,0			71,98%	43,068	40,0	3,1	28,8
33	Ne	34,0			75,71%	44,907	42,0	2,9	31,8
34	Po				103,31%	44,1	44,1	0,0	45,6
35	Út				109,14%	45,0	45,0	0,0	49,1
36	St				119,62%	45,8	45,8	0,0	54,8
37	Čt				117,96%	46,6	46,6	0,0	55,0
38	Pá				102,29%	47,5	47,5	0,0	48,6
39	So				71,98%	48,3	48,3	0,0	34,8
40	Ne				75,71%	49,2	49,2	0,0	37,2
41									
42									
43									

Obrázek 3: Jednotlivé fáze výpočtu odhadu návštěvnosti pomocí programu MS Excel



Obrázek 4: Odhad návštěvnosti 1/2



Obrázek 5: Odhad návštěvnosti 2/2

4.6.2.2 Zhodnocení použití metody

Vzhledem k svému charakteru je metoda použitá v aplikaci pro určování přibližného odhadu návštěvnosti v následujícím období relativně použitelná převážně z těchto důvodů:

- Pomocí jednoduchých (prostých) klouzavých průměrů se snaží identifikovat sezónní složku, která se v časové řadě projevuje jako periodické kolísání hodnot
- Poté provádí sezónní očištění, tj. z časové řady se snaží odstranit sezónní složku, jejíž přítomnost zamlžuje charakter trendu a znesnadňuje konstrukci odhadu předpovědi
- Pomocí získaných sezónních indexů je možné konstruovat předpovědi. Sezónně očištěná řada se extrapoluje pomocí lineárního exponenciálního vyrovnání a získané předpovědi se vynásobí příslušnými sezónními indexy

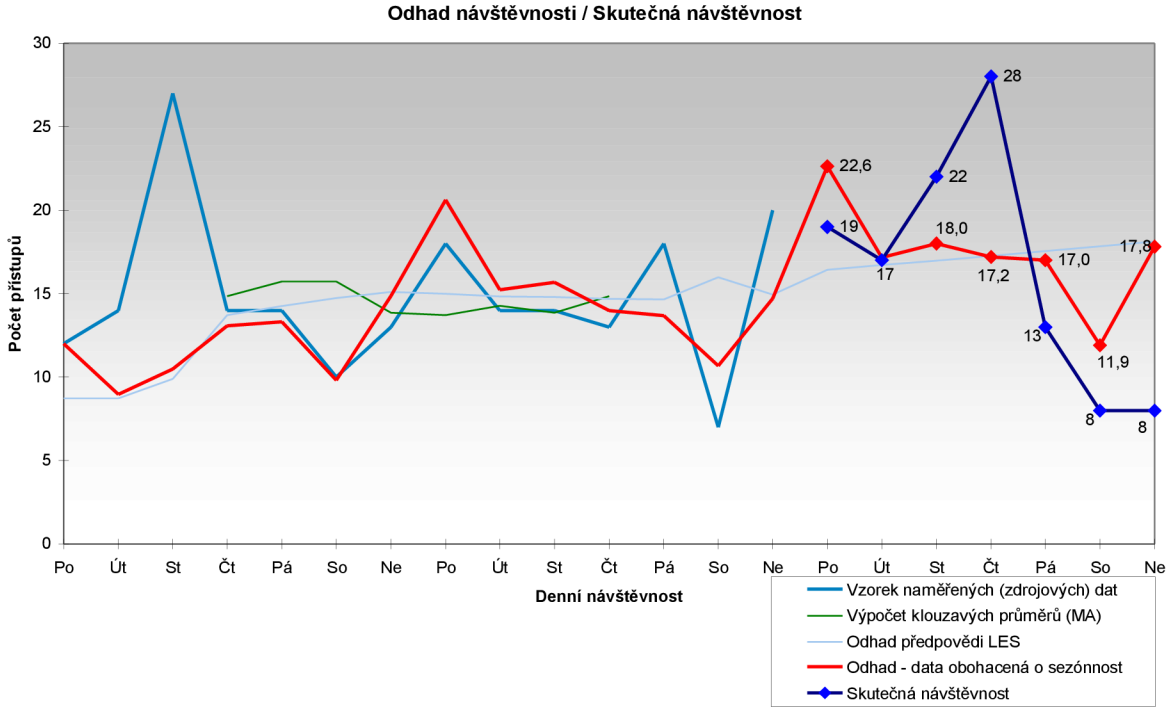
Možnou nevýhodou použití této metody je to, že její lineární růst či pokles daný výpočtem není ničím omezen a pro předpověď odhadu návštěvnosti může do budoucna hůře modelovat realitu.

4.6.2.3 Porovnání odhadu metody se skutečnou návštěvností

Jako příklad č. 1 uvedu odhad denní návštěvnosti následujícího týdne ze zdrojových dat předchozích dvou týdnů. Naměřená data (jak zdrojová, tak skutečný odhad návštěvnosti) vykazují jistou nestabilitu a od toho se také odvíjí porovnání v úspěšnosti metody.

Zdrojová data – 1. týden							
Den / Přístupů	Po / 12	Út / 14	St / 27	Čt / 14	Pá / 14	So / 10	Ne / 13
Zdrojová data – 2. týden							
Den / Přístupů	Po / 18	Út / 14	St / 14	Čt / 13	Pá / 18	So / 7	Ne / 20

Odhad návštěvnosti metodou v následujícím týdnu							
Den / Přístupů	Po / 23	Út / 17	St / 18	Čt / 17	Pá / 17	So / 12	Ne / 18
Porovnání se skutečnou návštěvností							
Den / Přístupů	Po / 19	Út / 17	St / 22	Čt / 28	Pá / 13	So / 8	Ne / 8
Úspěšnost	15/19	1	18/22	17/28	9/13	4/8	0
Celkem	((15/19 + 1 + 18/22 + 17/28 + 9/13 + 4/8) / 7) * 100 = **63%**						



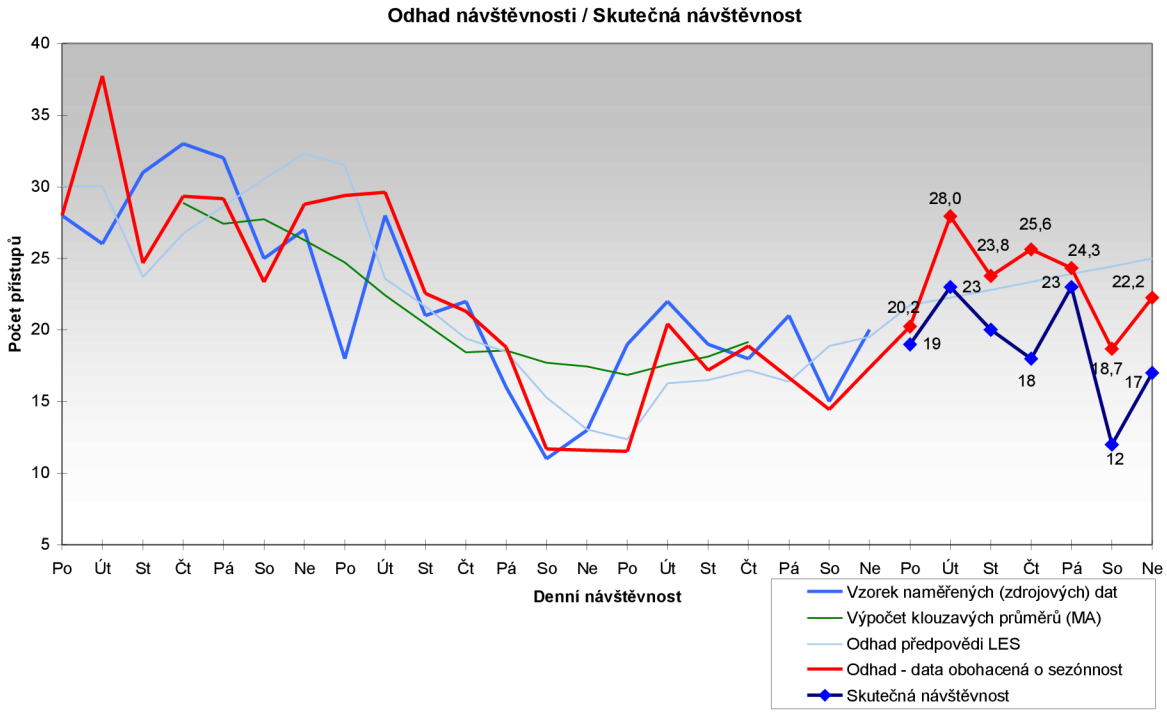
Obrázek 6: Porovnání odhadu se skutečnou návštěvností 1/2

Na tomto příkladu můžeme vidět snahu metody ze zdrojových dat dle očekávání předpovědět trend do budoucna. Bohužel ale není možné předpovídat náhlé výkyvy v návštěvnosti od trendu z dříve naměřených dat a skutečné návštěvnosti webu. Úspěšnost mezi odhadnutými a skutečně naměřenými přístupy je 63 %.

Příklad č. 2 demonstruje použití metody na delším intervalu zdrojových dat (3 týdny), naměřená data jsou navíc stabilnější a nepředstavují tak velké kolísání, jako v předchozím příkladě.

Zdrojová data – 1. týden							
Den / Přístupů	Po / 28	Út / 26	St / 31	Čt / 33	Pá / 32	So / 25	Ne / 27
Zdrojová data – 2. týden							
Den / Přístupů	Po / 18	Út / 28	St / 21	Čt / 22	Pá / 16	So / 11	Ne / 13
Zdrojová data – 3. týden							
Den / Přístupů	Po / 19	Út / 22	St / 19	Čt / 18	Pá / 21	So / 15	Ne / 20

Odhad návštěvnosti metodou v následujícím týdnu							
Den / Přístupů	Po / 20	Út / 28	St / 24	Čt / 26	Pá / 24	So / 19	Ne / 22
Porovnání se skutečnou návštěvností							
Den / Přístupů	Po / 19	Út / 23	St / 20	Čt / 18	Pá / 23	So / 12	Ne / 17
Úspěšnost	18/19	18/23	16/20	10/18	22/23	5/12	12/17
Celkem	((18/19 + 18/23 + 16/20 + 10/18 + 22/23 + 5/12 + 12/17) / 7) * 100 = **74%**						



Obrázek 7: Porovnání odhadu se skutečnou návštěvností 2/2

Díky delšímu intervalu zdrojových dat je metoda schopna lépe odhadnout trend, jehož charakter se dlouhodobě mění z klesajícího v rostoucí. V naměřených datech se navíc nevyskytují takové odchylky v sezónních vlivech, které by větší mírou zamlžovali a znesnadňovali konstrukci odhadu předpovědi. Díky těmto vlivům je úspěšnost metody 74 %.

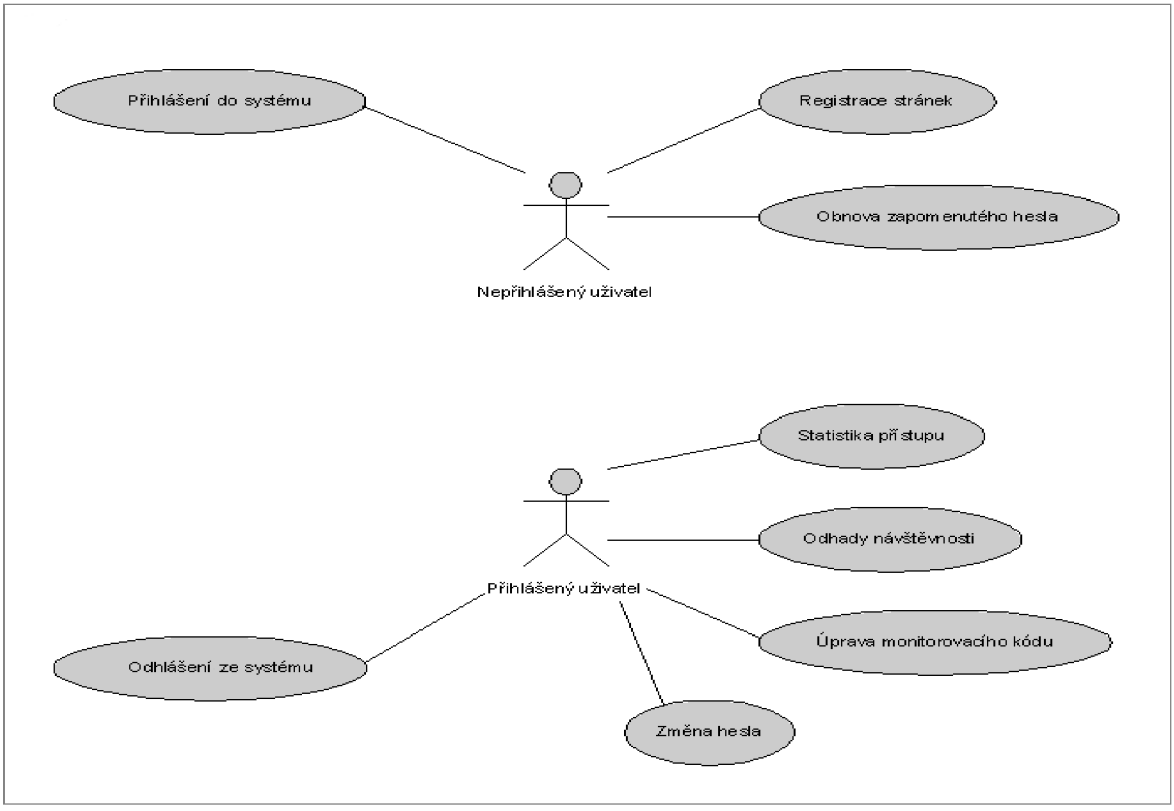
5 Implementace aplikace

Při implementaci systému statistik jsem používal PHP verze 5.2.2, databázový systém MySQL verze 4.0.27 a webové rozhraní pro práci s databází MySQL phpMyAdmin verze 2.5.5-pl1. Pro testování aplikace jsem použil fakultní server EVA FIT VUT.

5.1 Webové uživatelské rozhraní

5.1.1 Diagram případu užití

Diagram případu užití (*Use Case Diagram*) je diagram pro vizualizaci případů užití. Spolu s ER diagramem je navržen pomocí modelovacího jazyka UML. Na obrázku 8 jsou znázorněny role uživatelů, kteří vstupují do systému a aktivity, které mohou provádět.



Obrázek 8: Use Case diagram

Use Case diagram aplikace je tvořen následujícími UML prvky:

- Use Cases – případy užití, které zachycují funkčnost aplikace
- Actors – aktéři, kteří pracují se systémem (vstupují do něj, používají jej)

- Associations – asociace, vazba mezi aktérem a případem užití

5.2 Vzhled a funkcionalita uživatelského rozhraní

Aplikace využívá design šablon záhlaví, obsahu stránky a zápatí, ve kterých jsou využity možnosti kaskádových stylů CSS.

5.2.1 Domovská stránka

Domovská stránka aplikace (soubor index.php) obsahuje základní informace o funkcionalitě. V případě, že je uživatel přihlášen, zobrazí se příslušná uvítací zpráva.

5.2.2 Registrace

Sekce používaná k zaregistrování nového webu, který si uživatel přeje monitorovat, plní následující úkoly:

- Zobrazí formulář pro registraci uživatele včetně nastavení různých parametrů systému získávání a zpracování statistik
- Ošetří odeslaná data pomocí regulárních výrazů a již jednou správně zadaná data ve formuláři zobrazí
- Zajistí, aby byla v databázi uložena pouze unikátní uživatelská jména
- Po úspěšném zaregistrování uživatele v závislosti na zvolených parametrech vygeneruje a zobrazí monitorovací kód, který lze následně vložit do sledovaného webu pro zahájení získávání dat o návštěvnících

5.2.3 Přihlášení a odhlášení

Proces přihlášení spočívá v porovnání zadaného uživatelského jména a hesla se záznamy uloženými v databázi a v případě úspěchu následné registraci vybraných hodnot (především informace o parametrech monitorování) v objektu uživatelské relace. Z toho vyplývá, že pokud se chce uživatel úspěšně přihlásit, měl by mít nastavenou podporu ukládání cookies pro uložení identifikátoru uživatelské relace (sezení). Podle toho zda je uživatel přihlášen nebo odhlášen se v menu zobrazí příslušné odkazy.

Po odhlášení z aplikace budou informace o uživateli uložené na serveru odstraněny a z klientského počítače vymazán soubor cookie.

5.2.4 Zapomenutí a změna hesla

K situaci kdy některý uživatel zapomene heslo pochopitelně může dojít a tak aplikace nabízí možnost vygenerování hesla nového. Vychází se z toho, že uživatel je při registraci povinen zadat uživatelské jméno a svoji e-mailovou adresu. V případě, že heslo zapomene, má možnost pomocí formuláře zadat své uživatelské jméno, jehož existenci ověříme v databázi a případně vybereme e-mail na který odešleme heslo nově vygenerované. E-mail odešleme pomocí PHP funkce `mail()`. Pro správnou funkci se předpokládá správné nastavení SMTP.

Změna hesla je uživateli nabídnuta pouze za předpokladu, že je přihlášený a nepožadujeme po něm heslo staré (současné). Po zadání nového hesla a úspěšné kontrole nového hesla je uživateli heslo v databázi změněno.

5.2.5 Úprava monitorovacího kódu

V případě, že si uživatel přeje změnit chování skriptu, který zjišťuje, zpracovává a ukládá informace o návštěvnících podle zvolených parametrů, je mu tato možnost po přihlášení pomocí formulářových prvků nabídnuta. Po změně se doporučuje aktualizovat monitorovací kód vložený na webu a nové přihlášení uživatele do systému, jelikož to pravděpodobně ovlivní pozdější zobrazení statistik a nemá smysl vypracovávat uživateli statistiky, které si nepřeje monitorovat.

5.2.6 Vypracování statistik

Vypracování statistik je v podstatě zobrazování získaných informací do srozumitelných a potřebných podob. Statistiky, které mohou být z pohledu uživatele zpracovány jsem již uvedl v kapitole *4.5*.

Součástí vypracování statistik by neměla chybět možnost zvolit si období, pro nějž se má statistika vypracovat. Uživatel má možnost vybrat si z kalendáře mezi dnem, týdnem, měsícem nebo formuláře s textovým polem období, pro které si přeje statistiku vypracovat.

Jako možné grafické znázornění systému statistik jsem se zaměřil na vygenerování tabulky v podobě sloupcového grafu a také koláčového grafu za pomoci PHP funkcí pro práci s generováním grafiky a textu do obrázku. Generování tabulky pro určité období probíhá jednoduše pomocí cyklu přes všechny nalezené záznamy v databázi, která splňují požadovaná kritéria. V jednotlivých průchodech cyklem se kombinují značky pro formátování HTML tabulky spolu s SQL dotazy na databázi a vkládáním obrázku s různě nastavenou šířkou pro různé výsledky SQL dotazů.

Tím získáme výsledný vzhled grafu, který obsahuje jak jednotlivé položky, tak počty přístupů s touto položkou doplněnou o obrázek, jehož šířka je dána poměrem k největší hodnotě dané statistiky.

Systém zobrazení statistik obsahuje rovněž filtr, kterým lze případně nastavovat zobrazení jednotlivých statistik. Položky tohoto filtru jsou závislé na nastavených parametrech sledování, které se generátor filtru dozví z proměnných uživatelské relace po přihlášení konkrétního uživatele.

5.2.7 Odhady návštěvnosti

Funkce odhadující návštěvnost v následujícím období jsou závislé na datech jim předaných pomocí formuláře. V době psaní této práce byly v systému k dispozici funkce pro odhad:

- Hodinové návštěvnosti
- Denní návštěvnosti

Za pomoci prvků formuláře lze nastavit který odhad se má vypracovat včetně parametrů předaných metodě v podobě intervalu zdrojových dat, cílových dat a toho, zda si přejeme SQL dotazem na databázi vybrat a následně odhadnout pouze počty unikátních přístupů (visits) nebo všech přístupů na sledované stránky (shlédnutí). Grafické znázornění odhadů návštěvnosti je zobrazitelné uživateli v podobě spojnicového grafu v PHP, který zobrazuje jak data zdrojová, tak i odhady a sloupcového grafu zobrazující pouze odhady v následujícím období.

5.3 Zabezpečení systému

5.3.1 Zabezpečení proti neoprávněnému vstupu pomocí session

Z pohledu uživatele by měla v systému existovat nějaká bezpečnostní procedura, která by zamezila neoprávněnému vstupu na cizí zaregistrovaný web. Systém statistik je navržen tak, že se pro přihlášení od uživatele vyžaduje uživatelské jméno a heslo. Tyto údaje se po odeslání formuláře testují s údaji v databázi, a pokud odpovídají, vytvoří se proměnná typu *session*. Identifikátor relace *PHPSESSID* se uloží v podobě cookie do klientského prohlížeče a registrované proměnné jsou uživateli dostupné do doby, než se odhlásí, ukončí prohlížeč, nebo vyprší životnost cookie.

Na začátku každého skriptu se kontroluje, zda je proměnná typu session zaregistrována a platná. Pokud ano, tak se pokračuje ve zpracování stránky. Pokud ne, vypíše se uživateli chybová zpráva a je mu nabídnuta možnost přihlášení.

6 Závěr

Kombinace použitých technologií vedla ke zdárné implementaci požadavků kladených na systém statistik. Systém se v současné době nachází v první verzi, která slouží především pro potřeby testování. Pro nasazení systému do reálného provozu by bylo třeba delšího testování, čímž by se dala najít některá další vylepšení.

Pokud by se systém opravdu nasadil do reálného provozu, požadavky kladené na návrh systému by snad měli umožnit současné zpracování několika desítek až stovek uživatelů a jejich registrovaných webů. Takový stav, především zátěž na server a databázi se ale velmi těžko odhaduje předem.

Náplň bakalářské práce byla pro mě velmi přínosná. Vyzkoušel jsem si práci na větším projektu s rozložením na delší období a blíže jsem poznal snad všechny partie použitých technologií. Při navrhování jsem pronikl do jazyka PHP a hlavně do jeho spolupráce s databází MySQL a webovým serverem.

Co se týče dalšího vývoje projektu, bylo by dobré zapracovat na zjišťování některých dalších informací o návštěvnících. Jako příklad uvedu zjišťování regionální lokace návštěvníků, kde takto získané informace mohou sloužit k různým marketingovým účelům. Vzhledem k uvedení systému statistik do provozu v těsné návaznosti s odevzdáním této práce byly v sekci zabývající se odhady návštěvnosti záměrně vynechány některé další statistiky (týdenní, měsíční, roční odhad návštěvnosti), jelikož pro svůj výpočet potřebují delší období zaznamenaných dat o přístupech. Z již existujících je to odhad hodinové a denní návštěvnosti, které jsou uživateli znázorněny v podobě tabulky sloupcového grafu a spojnicového grafu generovaného jako obrázek formátu PNG s pomocí grafických funkcí PHP. V dohledné době bych se k zmíněným dalším chtěl vrátit a v systému je doimplementovat.

Literatura

[1] Gilmore, W. J. *Velká kniha PHP5 & MySQL*, Zoner Press, 2005, 712 s., ISBN 80-86815-20-X.

[2] Ullman, L. *PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek*, Computer Press, 2004, 536 s., ISBN 80-251-0063-4.

[3] Kosek, J. *PHP – Tvorba interaktivních internetových aplikací*, Grada Publishing, 1998.

Internetové zdroje

[1] Jelínek, Lukáš, et al. *Online PHP manual* [online]. c2001-2007 , last updated 24 Mar 2007 [cit. 2007-05-07]. Dostupný na URL: <www.php.net/manual/cs> (květen 2007).

[2] Rozsáhlý zdroj informací o tvorbě WWW-stránek, včetně skriptů, programování webových aplikací apod. [online]. 1999-2006 [cit. 2007-05-03]. Dostupný na URL: <www.kosek.cz> (květen 2007).

[3] Janovský, D., *Návod na tvorbu webových stránek* [online]. Dostupný na URL: <www.jakpsatweb.cz> (květen 2007).

[4] Kolektiv autorů serveru interval.cz, *Články o programování v PHP* [online]. Dostupné na URL: <www.interval.cz> (květen 2007).

[5] Kolektiv autorů serveru builder.cz, *Články o programování v PHP* [online]. Dostupné na URL: <www.builder.cz> (květen 2007).

[6] Kolektiv autorů serveru zive.cz, *Články o databázích* [online]. Dostupné na URL: <www.zive.cz> (květen 2007).

[7] Staeding, Andreas. *List of User-Agents* [online]. c2002-2007 , last updated 05/11/2007 [cit. 2007-11-05]. Dostupný na URL: <http://www.user-agents.org/> (květen 2007).

[8] Wikipedie otevřená encyklopedie. Dostupná na URL: <http://www.wikipedia.org/> (květen 2007).

[9] Nau, Robert F. Averaging and smoothing models. *Forecasting* [online]. 2005 [cit. 2007-05-06]. Dostupný na URL: <http://www.duke.edu/~rnau/411home.htm#class4> (květen 2007).

Seznam příloh

Příloha 1. Manuál

Umístění systému

Systém se nachází ve funkčním stavu na fakultním serveru FIT na adrese http://www.stud.fit.vutbr.cz/~xpetri10/BP/php/index.php nebo jej lze instalovat z přiloženého CD.

Přihlášení do systému

Pro účel testování je po zaregistrování v systému monitorována návštěvnost webu www.klubicko.eu
Pro vstup do uživatelské sekce:

- login: xpetri
- heslo: qwert

Instalace a konfigurace z CD

Pro správnou funkčnost systému je třeba mít nainstalovaný webový server, na kterém aplikace poběží (například Apache nebo Microsoft IIS) a databázový systém MySQL.

Samoinstalační balík EasyPHP

Jak nainstalovat a zprovoznit skripty PHP a SQL aplikace na vlastním počítači? Pokud netušíte, jako příklad uvedu balík EasyPHP určený pro systém Windows, který zahrnuje vše potřebné pro správný chod aplikace. V balíku je obsažen webový server Apache, skriptovací jazyk PHP, databáze MySQL a webový konfigurační nástroj PHPMyAdmin. Podobá se jiným balíčkům, jako jsou třeba PHPTriad nebo PHPHome. Balík je možné stáhnout ze stránek www.easyphp.org . Aktuální stabilní verze (1.8) má necelých 8 MB a obsahuje:

- Apache 1.3.33
- PHP 4.3.10
- MySQL 4.1.9
- PHPMyAdmin 2.6.1

Před samotným spuštěním instalace je nutné se ujistit, zda na pozadí (portu) jako služba neběží Apache nebo MySQL (např. z předchozí instalace), což by mohlo vést k problémům. V nouzi se dá upravit soubor *httpd.conf* serveru Apache tak, aby se spouštěl na jiném portu. Další úskalí může být v tom, že používáte nějaký firewall nebo jiný program. Tam je mu potřeba běh serveru povolit.

Po instalaci je možné provést kontrolu funkčnosti zadáním následující URL adresy do prohlížeče:

http://localhost

Již by se měla objevit stránka odněkud z kořenového adresáře lokálního serveru. Adresář, který představuje kořen dokumentů se standardně vytvoří v:

C:\Program Files\EasyPHP1-8\www

Po nahrání zdrojových souborů aplikace do tohoto adresáře by se např. po zadání

http://localhost/statistika/php/index.php

měla již zobrazit domovská stránka aplikace. Více informací o konfiguraci serverů Apache a MySQL se můžete dočíst např. na adrese http://jary.borec.cz/prog/easyphp.php

Dále je v systému MySQL nutné vytvořit databázi a požadované tabulky. Pro tento účel přikládám soubor *pocitadlo.sql* který obsahuje i data potřebná pro správnou práci systému.

Zdrojové soubory je nutné pro uvedení systému do provozu na internetu nahrát do prostoru webového serveru. Dále je nutné nastavit konfiguraci připojení k databázi v souboru *mysql_spojit.php*, včetně některých dalších definovaných proměnných. Více informací o nastavení je poskytnuto v samotném souboru.

Příloha 2. Obsah přiloženého CD

Na přiloženém CD jsou umístěny soubory v následujících adresářích:

- **kořenový adresář –** obsahuje soubor *readme.txt* popisující mapu CD
- **\zdrojove_soubory** – obsahuje zdrojové soubory, kaskádový styl a obrázky potřebné pro aplikaci
- **\sql** – obsahuje SQL skript pro vytvoření a naplnění databáze
- **\dokumentace** – obsahuje text bakalářské práce ve zdrojovém tvaru a formátu *.pdf*