

**Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta**



Bakalářská práce

World wide web aplikace

(World Wide Web Applications)

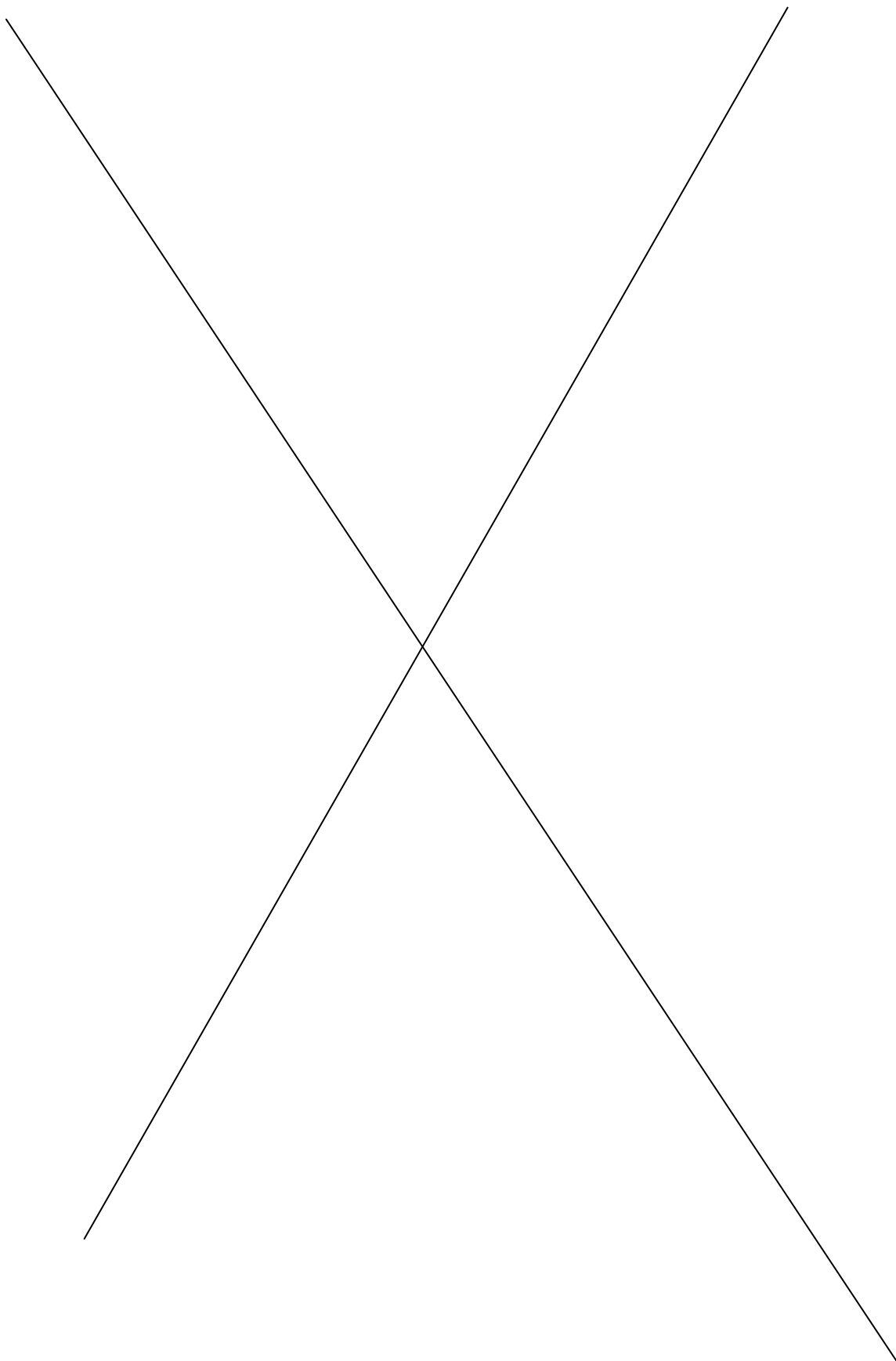
Michal Vondráček

Ing. Pavel Šimek, Katedra Informačních Technologií

Studijní program: Informatika

© 2009 ČZU v Praze

// ZADANI BAKALARSKE PRÁCE (ORIGINAL NEBO KOPIIE)



Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „World wide web aplikace“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne

Poděkování

Rád bych několika slovy poděkoval vedoucímu mé bakalářské práce Ing. Pavlu Šimkovi za jeho odbornou pomoc, konzultace a pomoc při korektuře. Rovněž děkuji Ing. Ivu Náglovi za myšlenkový přínos laika při konstrukci samotné WWW aplikace, Mgr. Věře Vondráčkové a MUDr. Ivě Malé za stylistickou korekturu, Zuzaně Veverkové za pomoc s překlady z a do anglického jazyka a dalším nejmenovaným osobám, které mě ovlivnili před a při vzniku této práce.

World wide web aplikace

World Wide Web Applications

Anotace

Bakalářská práce formuluje základní prostředky potřebné pro vznik WWW aplikací. Charakterizuje obecně WWW aplikace a popisuje funkce a vlastnosti některých známějších aplikací. Zobecňuje postup pro vznik nové WWW aplikace, který je doprovázen vznikem konkrétní WWW aplikace. Věnuje se podrobněji krokům vedoucím k jejímu vzniku, počínaje návrhem až k realizaci a uvedení do provozu.

Zdůrazňuje význam WWW aplikací, jejich praktičnost, cíle, snahy, ale také jejich zranitelnost a i jiné negativní vlastnosti, které s WWW aplikacemi přicházejí.

Annotation

A bachelor job formulates the basic tools needed for creating WWW applications. It generally characterizes WWW applications and describes the functions and features of some common applications. It generalizes the procedure of creating a new WWW application, which is accompanied by the development of the current application. It is applied more specifically to the steps leading to its creation, beginning with a proposal and ending with a realization, and then initiating its operation.

It emphasizes the significance of WWW applications, their practicality, objective, and effort, but also their vulnerability and other negative aspects, all which come with WWW applications.

Klíčová slova

www aplikace, apache, internet, php, http, mysql, 4u.zvyk.com, foryou, uml.

Keywords

www applications, apache, internet, php, http, mysql, 4u.zvyk.com, foryou, uml.

Obsah

World wide web aplikace.....	1
1. Úvod.....	4
2. Cíl práce a metodika	7
3. Prostředky webových aplikací	9
3.1. Pracovní a vývojová prostředí	10
4. Programovací jazyky	12
4.1. Programovací jazyky potřebné pro aplikaci 4U (viz 6.)	14
5. Současné aplikace.....	16
6. Návrh a realizace vlastní webové aplikace	19
6.1. Návrh	20
6.2. Iluminace	21
6.3. UML	22
6.4. Volba realizace.....	22
6.5. Návrh jádra	23
6.6. Návrh GUI (<i>Graphical User Interface</i>).....	24
6.7. Vzhled	25
6.8. Přenositelnost a dokumentace.....	26
6.9. Realizace jádra jazykem PHP	26
6.1. Část databázové struktury.....	31
7. Závěr	32
8. Seznam literatury.....	34
9. Přílohy.....	35
9.1. Skript index.php (jádro).....	35
9.2. Skript class_MySQL.php	38
10. Seznam obrázků.....	40
11. Seznam zdrojových kódů	40

1. Úvod

„Svět se mění a my s ním“ je známá fráze z televizní reklamy. Jejím odrazem lze sledovat rok od roku jak se vzdálenosti zmenšují, čas se zkracuje a svět je menším. To přináší své pro i proti. Jedním z mnoha aspektů je fakt, že internet je dostupný téměř po celé Zemi.

Internet jako síť sítí je silným médiem pro komunikaci počítačů po celé planetě a to v řádech milisekund. Historie říká, že vize o vzniku této sítě byla zachycena v roce 1946 v literatuře. V roce 1957 se objevuje „vzdálený přístup k počítači“ a time-sharing, který umožňuje práci více uživatelů současně. S rokem 1958 přichází DARPA s požadavkem na vznik sítě, která by zkvalitnila výměnu informací. Tato síť dostala jméno Arpanet (USA, 1966). Obdobně vznikla síť RAND(USA), NPL(GB), Cyclades(FR). Tyto sítě položily základ moderního internetu. V únoru 28., 1990 Arpanet odstranil ze sítě hardware, který byl dříve nutný a internet se stal nezávislým a funkčním.

V roce 1989 přichází Sir Timothy „Tim“ John Berners-Lee s návrhem vývoje World Wide Web, který publikoval v Information Management: A Proposal. O rok později spolu s belgičanem Robertem Cailliauem publikují koncept hypertextu. Tímto položil základy pro internetovou službu WWW a odstartoval vývoj tohoto komunikačního media se vším, co s tím v budoucnu souvisí. Přichází tedy doba World wide web a na internetu se objevují stacionární stránky. Publikování dat s informativním charakterem se stává brzy nedostatečné a objevují se požadavky na vznik dynamických prvků. Internet se začíná dostávat k více lidem a tím počet uživatelů na síti expanzně roste. Síť se rozšiřuje do nynější podoby.

S rostoucí sítí přibýly organizace, které usměrňují vývoj internetu jako takového. Jejich smyslem je formální formulace norem a standardů, které se stávají následně jakostní známkou produktů, které vznikají. Vznik těchto norem umožňuje kompatibilitu

vznikajících produktů od různých společností. Mezi hlavní organizace patří ISOC¹, IAB², IETF³, IESG⁴, IRTF⁵, IANA⁶ a ICANN⁷.

Jsou kladeny nároky na dynamizaci informací v internetu a odezvou na tyto požadavky vznikají world wide web aplikace. Tyto aplikace jsou programy se vzdáleným přístupem, které fungují na straně serveru a jsou ovládány dotazy od klientů. Na straně klienta se zobrazují v prohlížečích, které je graficky formátují a umožňují jednoduše odesílat dotazy na server. Aplikace mají různé funkce. Mohou sloužit jako vyhledávače v katalozích. Mohou také zprostředkovávat grafické rozhraní pro jiné internetové služby. Nejznámější takto zprostředkovávanou službou jsou WWW aplikace pro elektronickou poštu známé jako webmails⁸.

Aplikace komunikují s klientem prostřednictvím hypertextu a jsou nezávislé na platformě klientské stanice. Aplikace fungují nezávisle na rozdílnostech operačních systémů a typech zobrazovacích zařízení, která jsou sladěna díky normám a standardům. V dnešní době je WWW aplikací celá řada. Jedná se převážně o programy, které komunikují s databázemi nebo tvoří interface k jiným službám či nastavení hardwaru.

Aplikace umožňují vyhledávat, zobrazovat, přijímat a uchovávat potřebné informace v katalozích, které fungují jako databanky, nebo pracovat přímo se soubory či datovými toky. V aplikacích je pracováno i s citlivými daty, které se mohou stát předmětem zneužití a proto je zde důraz na dodržování určitého stupně zabezpečení vzhledem k datům, s kterými aplikace pracuje.

¹ www.isoc.org - Internet Society

² www.iab.org - Internet Architecture Board

³ www.ietf.org - Internet Engineering Task Force

⁴ www.ietf.org/iesg.html - Internet Engineering Steering Group

⁵ www.irtf.org - Internet Research Task Force

⁶ www.iana.org - Internet Assigned Numbers Authority

⁷ www.icann.org - Internet Corporation for Assigned Names and Numbers

⁸ Webmail vzniklo zkrácením názvu World wide **web** electronics **mail** access.

Aplikace mohou působit pozitivně i negativně v závislosti na datech, které zprostředkovávají. Mohou vyvolávat různé pocity a způsobovat i závislost. Aplikace mohou obsahovat bezpečnostní díry, kterých může být zneužito a tak místo prospěchu přinést spíše obtíže. Mohou být mocným nástrojem analytiků, prostředkem pro simulace nebo nástrojem pro šíření zpráv nejrůznějších obsahů, které se mohou dostat do nepravých rukou.

Aplikace pro simulace mohou být určeny pro vědecké účely i pro zábavu. WWW aplikace umožňují uživatelům internetu komunikovat na velké vzdálenosti v reálném čase. Vznikly hry⁹, simulující různé situace, spojující masy lidí nezávisle na věku, které mohou být potencionálním prostředkem k zrození hráčské závislosti. Například WWW aplikace, které jsou určeny k masové komunikaci - online chaty, nebo WWW portály¹⁰ umožňující sdílení fotografií a videí s jinými uživateli na hranici blízké narušení soukromí či umožňující šikanu mohou mít negativní dopad. WWW aplikace mají obecně vysoké dispozice, a proto jsou některé z nich potencionálním prostředkem k porušení zákona. Takovéto aplikace je snahou chránit od nežádoucích obsahů.

⁹ Tyto hry jsou známé pod termínem browsergame – hra v okně prohlížeče. Mezi nejznámější patří OGame nebo Travian.

¹⁰ www.facebook.com, www.youtube.com.

2. Cíl práce a metodika

Hlavním cílem práce je vyvinutí obecně platného principu, podle kterého bude navržena vlastní aplikace. Je vytvořen projekt, který aplikaci provede fází návrhu, konstrukce až k jejímu dokončení. Pomocí UML¹¹ je aplikace obecně specifikována a následně provedena vizualizace a odladění. Vybranou formou je realizována, vytvořena a následně zpřístupněna světu.

Dalšími cíly bakalářské práce je informovat o WWW aplikacích obecně, jaké přinášejí možnosti, jaké aplikace jsou dnes známé po technologické stránce. Je uvedeno několik různorodých WWW aplikací, s nimiž je dnes možno se setkat a pracovat. Pozornost je věnována zachycení jejich pozitiv i negativ, jejich síle a rychlosti. Dále je zachycena podstata fungování WWW aplikací. U vybraných veřejných aplikací jsou stručně popsány jejich hlavní funkce.

Práce se zaměřuje také na téma ergonomie aplikací, jejich socializaci a vlivů aplikací na jejich uživatele. Zmiňuje se o možné nebezpečnosti, závislosti či jiných úskalích. Probrány jsou aspekty komercializace, náboženství a etiky. Zhodnocuje aspekty všedního života práce s webovými aplikacemi. Upozorňuje na chyby.

Práce je dělena na dvě separované roviny. První z těchto rovin je obecný metodický postup pro tvorbu WWW aplikace. Uvádí příslušné aspekty, které v zásadě ovlivňují vznik nové aplikace. Podtextově je kladen důraz na ekonomické přednosti a kroky vedoucí k vybudování cílené aplikace. Druhá polovina této práce se věnuje fyzickému vzniku konkrétní aplikace. Od prvních myšlenek směřujících k cíli účelu aplikace až po fyzickou realizaci a praktické využití aplikace v reálném životě.

Práce je zaměřena na tvorbu logických algoritmických řešení problémů všedního života a odladění chyb zanesených lidským faktorem do logického modelu, který nebude probrán do podrobností z důvodu omezení rozsahu této práce. Fyzická

¹¹ UML - Unified Modeling Language

realizace konečného řešení aplikace v závislosti na aplikaci programovacích a skriptovacích jazyků je zachycena pouze pro jádro samotné aplikace, které je podrobně zdokumentováno a popsáno v rešeršní části této práce. Součástí rešeršní části je i podrobnější popsání související tematiky, která vyplývá z realizované WWW aplikace.

3. Prostředky webových aplikací

Jakákoliv WWW aplikace potřebuje ke své funkční existenci zázemí. Tím je rozuměn nějaký hardwarový stroj, který pro potřeby vývinu aplikace nemusí být připojen k internetu. Tento fyzický stroj musí obsahovat nějaký operační systém, který umožňuje běh softwarové aplikace zprostředkovávající hypertextovou komunikaci – obvykle známé jako http server software. Prakticky je možné si tento software naprogramovat, ale v praxi se používají již vyvinuté softwary, které mají mnoho funkcí a možných nastaveb. Nejznámější je dnes Apache server. Http server softwarů je celá řada. Dalšími jsou například: CERN server, NCSA server, Spinner, CL-http, MacHTTP, Perl server, Phttpd server, Windows http 1.4.

Pro skutečnou aplikaci potřebujeme, aby daný http server software měl podporu nějakého z programovacího nebo skriptovacího jazyka. To umožní oprostit se od statické formy sdělování informací a vytvořit dynamicky měnící se obsah dokumentu, který je následně interpretován http serverem.

Vhodným nositelem dat jsou kromě fyzicky uložených souborů na pevných discích i databanky. Ve výpočetní technice jsou databankami softwary pro SŘBD¹². Tyto systémy umožňují shromažďovat velké množství dat a obsahují výkonné algoritmy pro logické nakládání s těmito daty.

Apache

„The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. This project is part of the Apache Software Foundation. In addition, hundreds of users have contributed ideas, code, and

¹² SŘBD – Systém řízení báze dat (DBMS - Data Base Management System)

documentation to the project. This file is intended to briefly describe the history of the Apache HTTP Server and recognize the many contributors.“ [9]

PHP

„PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.“ [10]

„What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.“ [10]

3.1. Pracovní a vývojová prostředí

Každý počítačový jazyk má své charakteristické znaky, názvosloví a prvky, kterými se od sebe liší. Nedá se přesně určit, který nástroj je globálně nejvhodnějším pro vývin WWW aplikace. Volba nástroje závisí na osobní preferenci jednotlivých programátorů, kodérů. Tuto volbu může ovlivnit i samotný programovací jazyk. Existují různá jednostranně orientovaná vývojová prostředí pro jednotlivé jazyky, které spojují funkční, přehledné s užitečným a praktickým.

Globálně vhodné je pro jakýkoliv nástroj který člověk používá, aby mu usnadňoval práci. Vhodné zobrazení pomocí barevného zvýraznění klíčových slov, formátování textu a celková přehlednost práci ulehčuje.

Obecně se dá programovat v jakémkoliv editoru textu, ze kterého je možnost dokument vhodně uložit. Editorů existuje celá řada. Od téměř nevhodných jako je MS Word, přes vhodnější Notepad nebo MS Edit směrem k ještě vhodnějším jako je například VIM nebo Kate. Dále existují přímo určená prostředí, kterými jsou například Adobe Dreamweaver nebo Rapid PHP.

Jednostranně orientovaná vývojová prostředí obsahují doplňky, které je liší od běžných editorů a jsou přímo směřovány na určitý počítačový jazyk. Obsahují například kontextová menu s přednastavenými příkazy. Při formulaci funkcí může být zobrazována nápovědná charakteristika dané funkce. Zvýrazněná poloha kurzoru a numerický ukazatel polohy kurzoru může pomoci při odladování kódu. Pomáhají dynamicky se měnící párové značky v kódech v závislosti na poloze kurzoru.

Rapid PHP 2008 (fast PHP IDE and HTML, CSS, JavaScript editor)

„Rapid PHP editor is a powerful, quick and sophisticated PHP editor with features of a fully-loaded PHP IDE¹³ and speed of the Notepad. Convenient features enable you to instantly create and edit not only PHP, but also HTML, XHTML, CSS and JavaScript code, while integrated tools allow you to easily debug, validate, reuse, navigate and format your code. Whether you are a professional developer or pure enthusiast, with Rapid PHP editor you can get your job done faster, save time and increase productivity.“ [12]

Zend Framework

„Extending the art & spirit of PHP, Zend Framework is based on simplicity, object-oriented best practices, corporate friendly licensing, and a rigorously tested agile codebase. Zend Framework is focused on building more secure, reliable, and modern Web 2.0 applications & web services, and consuming widely available APIs from leading vendors like Google, Amazon, Yahoo!, Flickr, as well as API providers and cataloguers like StrikelIron and ProgrammableWeb.“ [11]

¹³ IDE – Integrované vývojové prostředí

4. Programovací jazyky

Programovacím jazykem se rozumí určitá forma komunikace mezi člověkem a strojem. Člověk zaznamenává své představy a myšlenky prostřednictvím programovacího jazyku, kterému počítač rozumí a je schopen je vykonávat. Programátor při konstrukci svých algoritmů je vázán na vlastnosti jazyka, ve kterém tyto algoritmy reprezentuje stroji. Smyslem těchto jazyků je usnadnění práce při vývoji algoritmů a lepší orientace v nich.

Programovací jazyky určené pro WWW aplikace jsou především interpretované programovací jazyky (JAVA, PERL, PHP, ASP, PHYTON, RUBY), zároveň patří mezi vyšší programovací jazyky vzhledem k abstrakčnímu rozhraní. Jazyky se také liší dle způsobu interpretace:

- Nepřevádí se přímo do strojového kódu, ale zdrojový kód se vždy spouští přes interpret, který ho zpracovává (BASH).
- Převádí se do tzv. bytecode (mezikódu) a ne do strojového kódu (PYTHON).
- Převádí se do strojového kódu po spuštění programu (JAVA).

Není jednoznačně možné určit ideální jazyk pro programování WWW aplikací. Každý z jazyků má své vlastnosti, které ho liší od ostatních. Jednotlivé jazyky mají různou stabilitu, výkonnost a rychlost při provádění algoritmů. Je prokazatelné, že pro různé WWW aplikace by mohl být vhodný jiný programovací jazyk. Také záleží, jaký je k dispozici hardware, a jestli je daný programovací jazyk umí správně využívat. Rychlosti zpracovávání algoritmů mohou u jednotlivých jazyků být různě časově náročné vzhledem k množství požadavků na zpracování během kratšího časového intervalu.

Neméně významným parametrem je kvalita zapsaných algoritmů programátorem. Zpravidla známým faktem je zapsání jednoho konkrétního řešení problému různými algoritmy. Tedy k řešení jedné situace může popisovat více různých zapsaných programátorských řešení.

Java

Java je dnes druhým nejrozšířenějším jazykem na světě. V nezávislosti na platformě a zároveň jednoduchosti a nízkým nárokům na programátora je využívána v mnoha projektech jak v prostředí world wide web tak i pro klasické aplikace, které běží přímo na PC.

Perl

Perl, jakožto jazyk, který v sobě přímo implementuje awk, sed a grep (možná *nIXové utility pro práci s textem), umožňuje dokonalou a snadnou práci s libovolným textem a za použití regulárních výrazů. Jeho objektová orientace umožňuje snadnou práci s kódem a rychlé přenesení objektového návrhu do praxe. Zároveň nenutí programátora využívat svých objektově orientovaných vlastností.

Obsahuje integrace pro práci se SŘBD, spolupracuje s jazyky pro WWW stránky (HTML, XML), podporuje UNICODE (standard pro kódování znaků). Interpret Perlu je možno zabudovat do mnoha zařízení, což přináší flexibilitu a nezávislost k platformě.

PHP

PHP je dnes jedním z nejpoužívanějších programovacích jazyků nejen pro tvorbu dynamických aplikací nebo WWW stránek. Funguje na mnoha platformách, operačních systémech a hardwarových zařízeních se softwarovými servery pro hypertext. Je nabízen zdarma. Je umožněno placeným způsobem převádět interpretované zdrojové kódy do binární formy za pomoci ZEND kompilátoru, která je pak rychlejší.

Hlavním směrem působení PHP je tvorba dynamických world wide web stránek – aplikací. Umožňuje také samostatné skripty, který mohou sloužit k jiným účelům. Například generování grafu do obrázku formátu JPEG nebo PNG. Dokumentace k tomuto jazyku včetně příkladů a krokových manuálů pro začátečníky je k dispozici na internetu. Jeho expanzní nárůst v používání však vedl k vzniku nevhodných postupů, které obsahují velká bezpečnostní rizika.

V polovině roku 2007 bylo PHP na více než 20 milionech WWW stránek a to z něj učinilo cíl pro hackery, kteří se snaží nabourat se do neošetřených mezer v kódech, které jsou psány neopatrně. Většinu z těchto bezpečnostních rizik je možné se zcela vyhnout při zkušeném programování.

Phyton

Phyton byl vyvinut jako jazyk pro neprogramátory, tedy jeho syntaxe je velice jednoduchá a snadno pochopitelná. Přebírá spoustu vlastností z funkcionálního programování a také plně implementuje objektový model. Koncepte tohoto jazyka umožňuje snadné a rychlé naučení.

Ruby

Ruby je moderním programovacím jazykem, plně implementuje objektový model a snaží se o co největší jednoduchost v nárocích kladených na programátora. V posledních letech oslabuje pozici PHP jako hlavního jazyka pro tvorbu world wide web aplikací.

4.1. Programovací jazyky potřebné pro aplikaci 4U (viz 6.)

Pro potřeby vývinu a naprogramování World wide web aplikace, která je dále popsána, bylo potřeba se seznámit s celou řadou jazyků a technologií. Pro aplikaci 4U („foryou“) byly vybrány tyto:

- UML jazyk je určen pro potřeby vývinu návrhů. Používají jej především návrháři, analytici a vývojáři. Jeho jednotná metodika vede k nezaměnitelnosti smyslu a tím je ideální pro případné komunikace ve skupině. Funguje stejně univerzálně jako periodická soustava prvků pro chemiky.
- PHP skriptovací jazyk, ve kterém je naprogramována samotná aplikace. Tento jazyk je zpracováván na straně serveru. [2, 7]

- HTML je hypertextový značkovací jazyk, kterým se pomocí předdefinovanými značkami (tagy) formátuje dokument. Tomuto jazyku následně rozumí prohlížeče a zobrazují dokument zformátován. [1, 5]
- CSS je jazykem, který je určen pro grafickou úpravu zformátovaných dokumentů. Umí upravit zobrazování dokumentu dle výstupního zařízení a dokument oživit různými efekty. [4]
- JAVASCRIPT je skriptovacím jazykem, který ovlivňuje chování dokumentu na straně klienta(v prohlížeči) a může měnit chování dokumentu i vzhledu. Použití technologie známé jako AJAX je možné měnit obsah již zobrazeného dokumentu pomocí změny DOM¹⁴. [6]
- SQL je dotazovací jazyk, kterým se formulují dotazy směřující na SŘBD. [3]

¹⁴ DOM - Document Object Model.

5. Současné aplikace

Za WWW aplikaci se dá považovat každá webová stránka, která svůj obsah dynamicky mění v čase. A to v souvislosti s dotazy, které zpracovává, nebo s daty které publikuje. Statická stránka není považována za WWW aplikaci.

Aplikací je celá řada, proto je zde uvedeno jen několik známějších WWW aplikací. Vlastním přínosem této práce bude charakterizování obecného postupu tvorby vlastní WWW aplikace.

Seznam.cz

Toto je nejznámější WWW aplikace z českých vyhledávačů. V současné době není pouze vyhledávačem, ale funguje jako informační portál. Na své titulní stránce implicitně obsahuje formulář, pomocí kterého můžeme začít hledat v obsahu WWW stránek, která byly již indexovány.

Obsahuje přihlášení do aplikace email.cz (další WWW aplikace – ovládání emailového boxu), stručný přehled zpravodajských novinek, informací ze sportu, počasí, horoskopu, televizního programu i kurzů měn.

Seznam.cz slouží jako rozcestník na velké množství dalších WWW aplikací.

Google.com

Vyhledávač google.com je jedničkou mezi fulltextovými vyhledávači. Jeho vlastnosti se těžko porovnávají s jinými, protože je mu těžké konkurovat.

Společnost, která tento vyhledávač vlastní, rozvíjí velké množství dalších WWW aplikací nejrůznějšího charakteru a svoji aktivitou tvoří vrásky v konkurenčním boji i velkým softwarovým společnostem.

Gmail.com

Patří do velké rodiny google aplikací. Tato aplikace slouží k obsluze emailových boxů (schránek), umožňuje klasickou emailovou korespondenci, kvalitní antispamové služby, vedení emailových boxů i jiným doménám.

Dále obsahuje velké množství doplňků, jako je plánovací kalendář, který lze vnitropodnikově sdílet nebo Jabber chat s možností archivace rozhovorů i tvorbu dokumentů různých charakterů – dokumentů, tabulek i prezentací.

Doplňky se stále vyvíjejí, každým dnem se objevují další funkční aplikace z dílny google.com

Hroch.czu.cz

Tato WWW aplikace je známa všem kdo zavádili o ČZU. Je určena nejen studentům při organizaci studia a studijních záležitostech, ale i vyučujícím, kteří například zapisují výsledky zkoušek. V období během roku není tolik vytížena jako v období zkoušek koncem letních a zimních semestrů.

CMS systémy

Jedná se obecně o systémy pro jednoduché spravování obsahu stránek WWW. Těchto systémů je celá řada, jsou jak komerční tak open source, které jsou volně šiřitelné.

V zásadě se tento typ WWW aplikace stáhne jako balíček, který se rozbalí a pomocí FTP se umístí na server do připravené složky. Proběhne základní konfigurace, nastaví se heslo pro hlavního správce stránek a spojení s SŘBD (obvykle s MySQL). Po této konfiguraci už je obvykle umožněn vstup administrátora do systému, který běží odděleně za veřejně přístupnými stránkami. Tímto systémem je ovládán obsah celého veřejného webu, který je dynamicky vyvoláván pomocí zobrazovací WWW aplikace.

Za CMS známý systém na půdě ČZU je WWW aplikace moodle.czu.cz.

phpMyAdmin

WWW aplikace, která umožňuje příjemnou správu SŘBD (MySQL) prostřednictvím intuitivního GUI (*Graphical User Interface*) zobrazeného uvnitř okna prohlížeče.

OGame

„OGame is a text-based, resource-management and space-war themed massively multiplayer online browser game with over two million accounts. OGame was created in 2000 and is produced and maintained by Gameforge AG. OGame is available in multiple different languages, while different nationalities have their own communities. The game does not differ between the nationality communities except in rare cases. Players are generally informed of news and other things such as changes in rules or new versions through the official forums.“ [8]

„OGame won the bronze Superbrowsergame Award in 2006.“ [8]

E-banking

Jsou aplikace určené k vzdáleným přístupům k bankovním účtům. Umožňují především zadávání plateb a dalším doplňkovým funkcím, které jsou rozdílné v závislosti na jednotlivých bankovních společnostech. Patří k WWW aplikacím, které jsou na internetu k dispozici veřejnosti a jsou nejvíce střeženy proti zneužití.

6. Návrh a realizace vlastní webové aplikace

Před tím, než jakákoliv WWW aplikace vznikne, musí být splněno několik primárních hledisek. Tato hlediska jsou milníky, kterými by měl proces zrodu World wide web aplikace projít dříve než bude zahájena jakákoliv fyzická činnost.

- Jedním z nejdůležitějších hledisek je stanovení si konkrétního, jasně a přesně definovaného cíle. Vhodnou definicí cíle je určen smysl a účel aplikace.
- Je vhodné věnovat pozornost doprovodným faktorům, které ovlivní tvorbu WWW aplikace. Z ekonomického hlediska nemá smysl tvořit aplikaci, která je již vytvořena, pokud ji můžeme použít pro náš záměr. Výjimkou může být vznik konkurenční aplikace. V případě tvorby konkurenční WWW aplikace musí být kladen zřetel na vytvoření viditelných odchylek od již existující aplikace.
- WWW aplikace by měla mít stanovenou cílovou skupinu, pro kterou bude tvořena – zaručí poptávku po aplikaci.
- Doplnění trhu je ideální možnost pro vznik nové WWW aplikace. Bude vybráno, pro koho aplikace bude stavěna, co bude umět nového a jaký bude přínos oproti aplikacím současným.
- Volby možností komercializace funkcí aplikace je velice rozsáhlé téma, kterému v této práci nebude věnován prostor.
- Přizpůsobení aplikace možnostem lokalizace vycházející z jazykových odlišností a zvyklostí pro danou lokaci. V závislosti na kulturních předpokladech a myšlenkovém poznání, které přichází v souvislosti s jiným jazykem.
- Udržitelnost by neměla být zanedbána. Umožní efektivní využití aplikace vzhledem k času a její správu. Zde platí pravidlo čím méně vynaložených prostředků, tím menší náklady spojené s udržováním.

- Zákony a nařízení v dané zemi, mohou ovlivnit vývoj aplikace. Většinou vedou k sepsání ustanovení, s kterým musí koncový uživatel souhlasit, aby mohl pracovat s aplikací.
- Bezpečnost a opatření k zajištění citlivých dat.

6.1. Návrh

Stanovení návrhu a zaznamenání myšlenek, kterými bude WWW aplikace popsána a charakterizována. Tento bod je klíčový pro další kroky, je jím zaznamenána určitá základna pro WWW aplikaci a v případě modifikace tohoto kroku musí být zkontrolovány kroky následující, tak aby byly vyřešeny problémy spojené s kompatibilitou.

Pro potřeby této práci vzniká aplikace „foryou“ (4u).

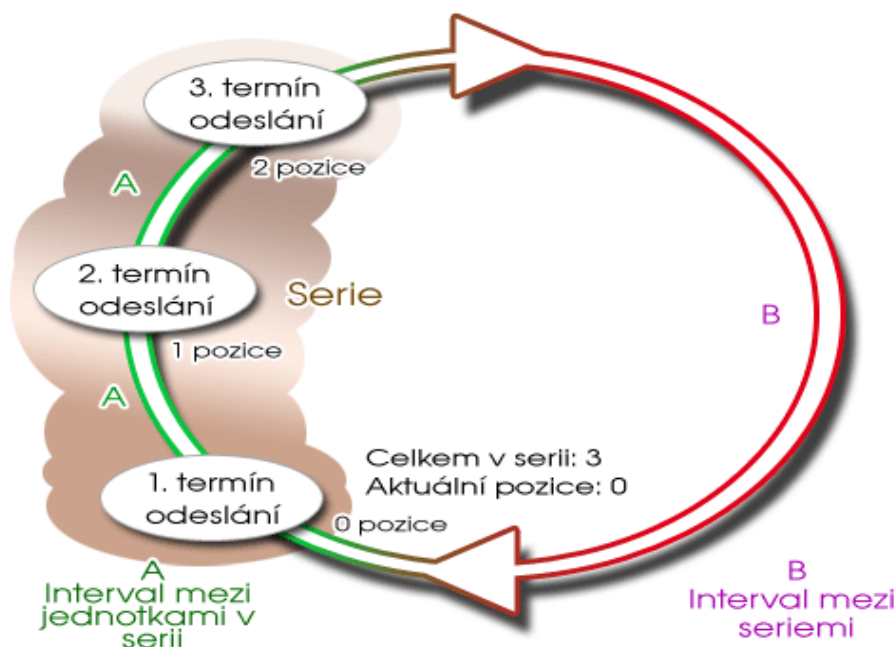
Hlavní funkce a myšlenky doprovázející návrh aplikace:

- Pravidelně odesílat upozorňovací zprávy na aplikaci tabletek hormonální antikoncepce.
- Odesílání upozorňovacích zpráv na mobilní zařízení.
- Zobecnění upozorňování na antibiotika, technické kontroly, servis zařízení, výročí a na ostatní periodicky se opakující události.
- Umožnit měnit znění těchto zpráv.
- Vytvoření individuálních cyklů pro odesílání těchto zpráv.
- Zabránění zneužití tohoto systému.
- Vytvoření algoritmů pro udržování systému.
- Zdarma.

6.2. Iluminace

V závislosti na obecném návrhu bylo rozhodnuto, že aplikace bude mít dvě poloroviny. První polorovina bude výkonným jádrem, které se bude starat o včasné rozesílání zpráv. Druhá polorovina bude GUI, se kterou budou uživatelé komunikovat a jejímž prostřednictvím nastavovat vlastní preference v rámci možností systému WWW aplikace.

Bylo potřeba vymyslet vhodné flexibilní nastavení algoritmu, jednoduše popsatelného jak po stránce matematické, tak srozumitelného pro člověka a zároveň umožnit definování pravidelných výjimek. Bylo vybráno popsání série a cyklu, ve kterém se série budou opakovat. Série bude obsahovat „n“ upozornění po časovém intervalu „A“ a cyklus bude charakterizovat časový interval „B“, který bude popisovat časovou vzdálenost konce předešlé série k začátku budoucí. Obrázek 1 - Schéma zobrazení série a cyklu zobrazuje sérii o třech prvcích s příslušnými intervaly.

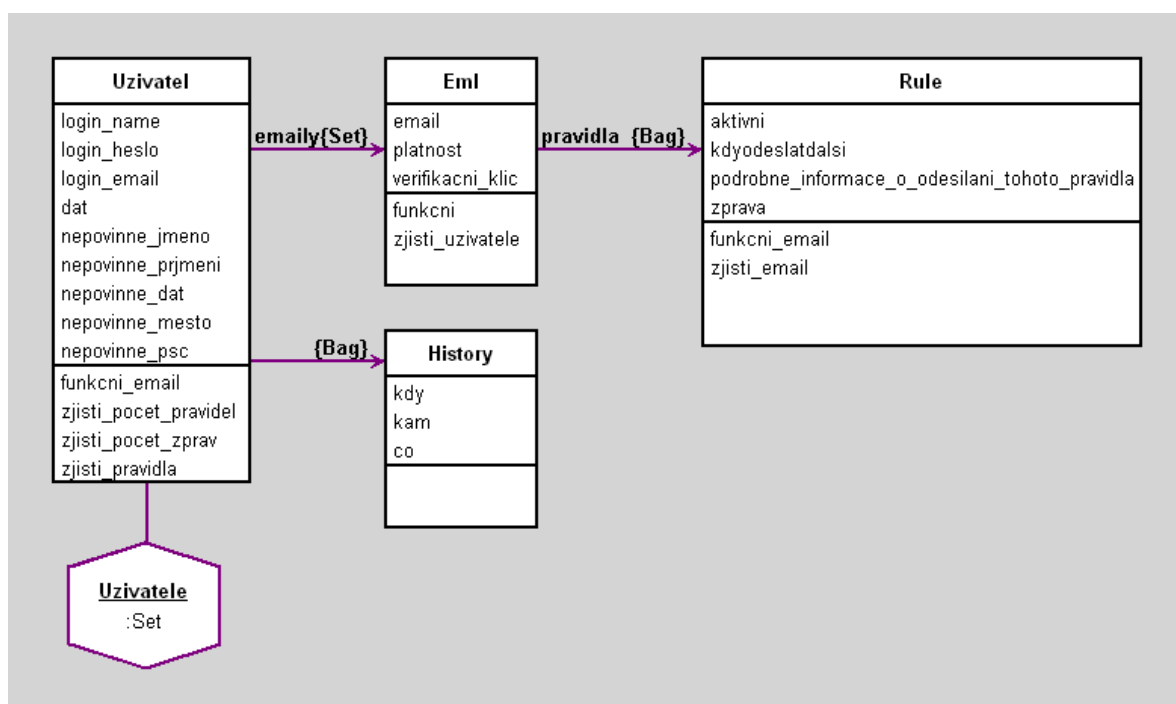


Obrázek 1 - Schéma zobrazení série a cyklu

Tímto způsobem je možné popsat nejrůznější druhy hormonálních antikoncepcí i antibiotik. Je možné tímto mechanismem logicky definovat všední dny od víkendu.

6.3. UML

Sestavení vhodného UML modelu nám nezávisle na volbě realizace umožňuje popsat funkce aplikace i data, která budou uchováována a tato data vhodně roztřídit a charakterizovat. Zároveň nám takto vytvořený model umožňuje zaručit datovou normalizaci. Je možné se při tvorbě UML přiklánět i k návrhovým vzorům. Obrázek 2 - UML - foryou popisuje UML model aplikace „foryou“.



Obrázek 2 - UML - foryou

Charakterizuje nám uživatele, na které se vážou individuální emailové adresy. Na tyto adresy se dále vážou „pravidla“, která popisují cyklus odesílání každé jednotlivé zprávy. Dále na uživatele navazuje informace o již odeslaných zprávách – historie. V tomto případě je vedena v pozadí aplikace z justičních a statistických důvodu.

6.4. Volba realizace

Z obecného modelu UML lze volit formu realizace. Forma realizace se stává konkrétní a ne vždy se zpravidla lze vrátit k zobecněnému návrhu v UML beze ztrát. V současné době je plno kvalitních prostředků – programovacích jazyků, ve kterých je možné UML věrně zachytit a zpětně se k němu vrátit.

Při volbě realizace má konečné slovo investor, který by měl být obeznámen s možným úskalím jednotlivých variant realizace.

Pro WWW aplikaci „foryou“ byl vybrán programovací jazyk PHP verze 4, který je neobjektový a relační systém řízení báze dat MySQL. Obě tyto volby minimalizují nákladové položky.

V závislosti na tomto výběru bude vytvořen skript pro zastoupení první poloroviny, který při korektním provozu nebude vykazovat žádný výstup. Pouze proběhne. Pro druhou polorovinu bude vytvořen framework s moduly sloužící jako GUI. Jednotlivé moduly budou funkčně nezávislé, ale budou sdílet data uložená v MySQL.

Takto rozdělená aplikace může mít skript první poloroviny fyzicky na jiném stroji než na stroji obsluhujícím framework s GUI, ke kterému budou mít přístup uživatelé WWW aplikace. Pouze tyto dva stroje budou přistupovat do společného databázového prostoru.

6.5. Návrh jádra

Jádro, jak už je možné chápat z názvu, je motorem, který se bude především starat o funkci rozesílání příslušných zpráv. Je potřeba jádro navrhnout tak, aby bylo co nejjednodušší na provedení, protože toto jádro je využíváno každou minutu a tím je zaručen prakticky kontinuálně funkční odesílací systém.

Jak už bylo řečeno, aplikace „foryou“ bude pravidelně rozesílat zprávy. Každá z těchto zpráv bude definována prostřednictvím pravidla, podle kterého se bude řídit její odeslání. Jádro bude tedy každou minutu vyhodnocovat všechna tato pravidla a bude z nich vybírat ta, s kterými bude v příslušný okamžik pracovat.

U těchto pravidel připraví příslušné znění zpráv, které odešle. Vypočte jejich další termín pro odeslání a pravidlo aktualizuje o tuto informaci. Dále jádro bude zajišťovat kontrolu validity emailových adres. V případě blížícího se vypršení odešle

upozornovací zprávu. Tato funkce je blíže specifikována v kapitole Návrhu GUI, modulu 4. Následně zapíše informace o všech odeslaných zprávách.

6.6. Návrh GUI (*Graphical User Interface*)

Tato kapitola by měla ideálně shrnout požadavky na zobrazení WWW aplikace pro její klienty. V zásadě by se měla pokusit mít intuitivní charakter, vyrovnanou koncepci barev, dodržení kontrastů a využití standardizovaných stylistických norem v textech, které ulehčují orientaci. Za kvalitní se dá považovat takové GUI, ke kterému běžný uživatel nebude mít otázky a bude schopen se sám orientovat v dané WWW aplikaci. Mezi možné obohacení GUI je volba vhodných stručných nápovědí s příkladem.

Rozložením rozsáhlejších textů do kapitol a kroků je možné WWW aplikaci funkčně rozdělit do částí, které uživatel snáze pochopí.

Aplikace „foryou“ byla rozdělena do následujících kapitol:

- Informace na úvod
(Jsou obsaženy informace k WWW aplikaci, je zde uvedena historie a možnosti, co aplikace nabízí.)
- Krok 1 – Registrace
(Obsahuje formulář, kterým lze založit uživatelský účet)
- Krok 2 – Aktivace uživatelského účtu
(Obsahuje formulář, do kterého uživatelé vloží kód, který jim byl doručen závěrem prvního kroku. Tímto krokem se stávají uživateli WWW aplikace.)
- Krok 3 – Přihlášení do systému
(Uživatelé aplikace se zde mohou přihlásit do aplikace nebo mohou použít formulář v záhlaví GUI z titulní strany)

- Krok 4 – Emailové účty
(Přihlášený uživatel si v aplikaci může přidávat emaily, na které bude chtít zasílat upozorňovací emaily. Zároveň je veden k jejich schválení.)
- Krok 5 – Tvorba pravidel
(Přihlášený uživatel si může vytvářet pravidla pro jednotlivé emaily. Prostřednictvím těchto pravidel mu budou zasílány upozorňovací zprávy)
- Statistiky
(Mají informativní charakter, obsahují čísla a grafy.)
- Donate – Věnovat darem
(Dobrovolná finanční podpora vývinu a provozu WWW aplikace)

6.7. Vzhled

WWW aplikace jsou tvořeny pro lidi, kteří je používají. Lidé jsou zmanipulovatelnými subjekty a zpravidla dají na první dojem. První dojem neodhalí kvality aplikace jako takové, ale hodnotí pouze vzhled a schopnost orientace. Vzhled aplikace by měl být na jiné logické úrovni než samotná aplikace a to tak, aby mohl být nezávisle na aplikaci měněn a upravován. Vzhled by měl vypovídat o aplikaci.

Nemohou být opomenuty schopnosti zobrazovacího zařízení. WWW aplikace byly určeny především pro PC¹⁵, ale i dnes je možné s nimi pracovat v prohlížečích textového charakteru, u kterých je minimální zobrazení grafických prvků. Taktéž je lze zobrazit na miniaturních přenosných zařízeních, známých jako PDA¹⁶, či na mobilních telefonech. Tato zařízení jsou omezena rozměrem zobrazovacího displeje, a proto by aplikace měly umožňovat různá zobrazení dle zobrazovacího zařízení.

¹⁵ PC - Personal computer

¹⁶ PDA - Personal Digital Assistant

V současnosti je nejefektivnější použití šablon¹⁷, do kterých se jednotlivé prvky vygenerované webovou aplikací doplní a následně se jako celek pošlou na výstup. Aplikace může měnit svoje výstupy v souvislosti na zdroj dotazů, které dostává. Výstup dle standardů je již v závislosti na prohlížeči zpracován a vyobrazen.

Vzhled aplikace zobrazované prohlížečem se stává GUI. Přehlednost zvyšuje jakost aplikace jako takové a umožňuje snazší ovladatelnost. Může být doplněna o grafické prvky. Ikony, fotky, náčrtky i grafy ve vhodné míře mohou zkvalitnit orientaci i zprostředkování informací. Vhodné použití animací je na rozhodnutí osoby zodpovědné za design aplikace. Pozornost člověka může být směřována buď na efekty nebo na obsah WWW aplikace.

6.8. Přenositelnost a dokumentace

Nezávisle na samotné world wide web aplikaci by měla být vedena kvalitní dokumentace obsahující informace o jednotlivých algoritmech a tvorbě samé. Součástí dokumentace by měl být postup informující o způsobu přenositelnosti aplikace. Přenositelnost informující o jednotlivém nastavení a skrytých souvislostech, jejichž nevědomost by vedla k nezdárnému přenosu aplikace na jiný server.

6.9. Realizace jádra jazykem PHP

Ukázka možnosti realizace jádra jazykem PHP. Tento PHP skript je připraven tak, aby mohl být volán pravidelně každou minutu a tím zajistil bezproblémové rozesílání zpráv. Obsahuje informace o vytvoření v hlavičce souboru.

Zdrojový kód 1 – index.php – výňatek z přílohy

```
<?
// #####
// ##### Header ##### Created by jossss @ 2k9; jossss@badl.org #
// #####
```

¹⁷ Šablony jsou známé jako templates.

Nastavuje informace, které se mají odeslat v záhlaví dokumentu. Vyžaduje načtení souboru `olv.php`, který především obsahuje informace pro připojení k MySQL databázi.

Zdrojový kód 2 – `index.php` – výňatek z přílohy

```
Header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
Header("Last-Modified: ".GDate("D, d M Y H:i:s")." GTM");
Header("Pragma: No-cache");
Header("Cache-Control: no-cache");

// #####
require_once("olv.php"); // ##### Load $OLV(array)#####
```

Dále vyžaduje načtení souboru `class_MySQL.php`, který obsahuje třídu zprostředkovávající komunikaci mezi PHP a MySQL (viz příloha). Nastavuje ukazatel, který v algoritmu bude používat pro komunikaci s databází.

Zdrojový kód 3 – `index.php` – výňatek z přílohy

```
// ##### Load fce paste(); ###
require_once($GLOBALS["OLV"]["basedir_functionsANDclasses"]."class_MySQL.php");
// #####
// ##### MySQL #####
if (($GLOBALS["OLV"]["mysql"]["comp"]!="") &&
($GLOBALS["OLV"]["mysql"]["username"]!="")) {
    $C_MySQL = new C_MySQL($GLOBALS["OLV"]["mysql"]["comp"],
        $GLOBALS["OLV"]["mysql"]["username"],
        $GLOBALS["OLV"]["mysql"]["userpass"],
        $GLOBALS["OLV"]["mysql"]["startDB"]);
} else {
    if ($GLOBALS["OLV"]["mysql"]["show_err"]=="1") {
        echo "Nebyl uskutečněn pokus o navázání spojení s MySQL. Toto chybové hlášení
můžete vypnout v olv.php (mysql-show_err)";
    }
}
// #####

// #####
// ##### PAGE #####
// #####
```

Prvním SQL dotazem vybírá informace o pravidlech, která bude v tomto kroku zpracovávat.

Zdrojový kód 4 – index.php – výňatek z přílohy

```
Set_Time_Limit(0);

// naplneni ukolu / poslani zprav podle RULE tabulky
$q = "
SELECT
    r.owner,
    r.id,
    r.email,
    r.active,
    r.serie_celkem,
    r.serie_pozice,
    r.serie_mezi_jednotkami,
    r.serie_mezi_seriemi,
    r.serie_limit,
    r.next_send,
    r.msg,
    e.address,
    e.exp_dat
FROM
    2nd_rule As r
    JOIN 2nd_email as e ON
        r.owner = e.owner
        AND
        r.email = e.id
WHERE
    (
        r.enable_from IS null
        OR
        r.enable_from < NOW()
    )
    AND
    (
        r.enable_until IS null
        OR
        r.enable_until > NOW()
    )
    AND
    (
        r.next_send <= NOW()
    )
    AND
    (
        r.next_send <> \"0000-00-00 00:00:00\"
    )
;
";
$exec = @$C_MySQL->query($q);
while ($l = @$C_MySQL->next_row_array()) {
    $ZAZNAM[] = $l;
}
```


Definuje funkci, pro získání informací o aktuálním svátku. Tato funkce je volána pouze jednou. Po prvním volání nastavuje informaci o svátku do proměnné, která je použita pro další události během tohoto vyvolání skriptu.

Zdrojový kód 5 – index.php – výňatek z přílohy

```

//////////////////////////////////// SVATEK
$GLOBALS["svatek"] = null;
function svatek($pokuscislo = 0) {
    $maxpocetpokusy = 5;
    Set_Time_Limit(30+$pokuscislo);

    if ($GLOBALS["svatek"] == null) {
        if ($GLOBALS["svatek"] == null) {
            sleep($pokuscislo);
            if ($f =
file("http://stmb.s.kvalitne.cz/svatky/index.php?inb=svatekdnes")) {
                foreach ($f as $item) {
                    $stream .= trim($item);
                }
                $string = Explode('<div class="svatky">', $stream);
                $string = Explode('<div class="napovdole">', $string[1]);
                $radky = Explode('<td valign="top">', $string[0]);
                $dnes = Explode("<b>", str_replace("/", "", strip_tags($radky[3],
"<b>")));
                $zitra = Explode("<b>", str_replace("/", "", strip_tags($radky[4],
"<b>")));
                $GLOBALS["svatek"] = trim($dnes[0]." ".$dnes[2]." ".$zitra[0]."
".$zitra[2]."");
            } else {
                if ($pokuscislo < $maxpocetpokusy) {
                    $GLOBALS["svatek"] = $this->svatek($pokuscislo++);
                } else {
                    $GLOBALS["svatek"] = "Svatek se nepodarilo najit.";
                }
            }
        }
        $GLOBALS["svatek"] = StrTr($GLOBALS["svatek"],
            "áčďéěěíňóřšťúůýžÁĀČĎĚĚĚĪŇÓŘŠŤÚŮÝŽ",
            "aacdeeeinoorstuuuyzAACDEEEINOORSTUUUYZ");
    }
    return $GLOBALS["svatek"];
}
//////////////////////////////////// SVATEK konec

```

Skript začne procházet záznamy vyhovující prvnímu dotazu, připraví znění zpráv dle individuálního nastavení každého pravidla a odešle upozorňovací email. Následně zapíše informaci o poslání tohoto emailu do historie v databázi.

Zdrojový kód 6 – index.php – výňatek z přílohy

```

For($i = 0; $i < Count($ZAZNAM); $i++) {
    ...
}

```

Skript pokračuje ověřováním informací, zkoumá a informuje o případném vypršení doby, která informuje o aktuálnosti informací. O případných nalezených informacích informuje a aktualizuje jejich současný stav nebo je vyřazuje z činnosti.

Zdrojový kód 7 – index.php – výňatek z přílohy

```
// posilani oznameni o vyprseni validity emailu
$HODINDopredu[] = 336;
$HODINDopredu[] = 168;
$HODINDopredu[] = 72;
$HODINDopredu[] = 48;
$HODINDopredu[] = 24;
$HODINDopredu[] = 12;
$HODINDopredu[] = 1;
For ($i = 0; $i < Count($HODINDopredu); $i++) {
    $q = "SELECT address
        FROM 2nd_email
        WHERE exp_dat LIKE '".DATE("Y-m-d H:i:", (time() + ($HODINDopredu[$i] * 60 *
60)))."%:'";
    $exec = @$C_MySQL->query($q);
    while ($l = @$C_MySQL->next_row_array()) {
        if (!(@mail(trim($l["address"]), "4u.zvyk.com", "U tohoto emailu vyprsi za
".$HODINDopredu[$i]." hodin/u validace, prodluzte si ji. 4u.zvyk.com", "From:
4u@zvyk.com"))) echo "Error: ".trim($l["address"]).', '.Date("Y-m-d H:i:s")."<br>";
    }
}

// Propadla validace
$q = "SELECT e.owner as owner, e.id as id, e.address as address
    FROM 2nd_email as e, 2nd_rule as r
    WHERE
        e.owner=r.owner AND
        e.id=r.email AND
        e.exp_dat < now() AND
        r.active = '1'
    ";
$exec = @$C_MySQL->query($q);
while ($l = @$C_MySQL->next_row_array()) {
    $VALIDACE[] = $l;
}
For($i = 0; $i < Count($VALIDACE); $i++) {
    if (!(@mail(trim($VALIDACE[$i]["address"]), "4u.zvyk.com", "U tohoto emailu vyprsel
a validace, prodluzte si ji. 4u.zvyk.com", "From: 4u@zvyk.com"))) echo "Error:
".trim($VALIDACE[$i]["address"]).', '.Date("Y-m-d H:i:s")."<br>";
    $q = "UPDATE 2nd_rule SET active = '0' WHERE email = ".$VALIDACE[$i]["id"]." AND
owner LIKE '". $VALIDACE[$i]["owner"]."'";
    $exec = @$C_MySQL->query($q);
}

// propadla validace u pokusu o registraci
$q = "DELETE FROM 2nd_w8_4_verify WHERE dat < '".DATE("Y-m-d H:i:s", Time()-
(24*60*60))."'";
$exec = @$C_MySQL->query($q);

// SET active = 0 ((( limit )))
$q = "UPDATE 2nd_rule SET active = '0' WHERE serie_limit > 0 AND serie_limit <=
counter AND active LIKE '1'";
$exec = @$C_MySQL->query($q);

Die();
?>
```

6.1. Část databázové struktury

Ukázka definice části databázové struktury, kterou používá jádro popsané v předešlé kapitole. Pro jádro jsou především potřebné informace o emailech a o pravidlech, která má kontrolovat či vykonávat. Jedná se především o databázové tabulky „2nd_rule“ a „2nd_email“, které jsou specifikovány dle následujícího zdrojového kódu MySQL definice tabulek:

Zdrojový kód 8 - MySQL definice tabulek

```
--
-- Struktura tabulky `2nd_email`
--

CREATE TABLE IF NOT EXISTS `2nd_email` (
  `owner` varchar(32) NOT NULL default '',
  `id` int(11) NOT NULL auto_increment,
  `address` varchar(60) NOT NULL default '',
  `counter` int(11) NOT NULL default '0',
  `exp_dat` datetime NOT NULL default '0000-00-00 00:00:00',
  `verf_key` varchar(10) NOT NULL default '',
  PRIMARY KEY (`owner`,`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

-----

--
-- Struktura tabulky `2nd_rule`
--

CREATE TABLE IF NOT EXISTS `2nd_rule` (
  `owner` varchar(32) NOT NULL default '',
  `id` int(11) NOT NULL auto_increment,
  `email` int(11) default NULL,
  `active` char(1) NOT NULL default '1',
  `enable_from` datetime default NULL,
  `enable_until` datetime default NULL,
  `serie_celkem` mediumint(9) NOT NULL default '0',
  `serie_pozice` mediumint(9) NOT NULL default '0',
  `serie_mezi_jednotkami` int(11) NOT NULL default '0',
  `serie_mezi_seriemi` int(11) NOT NULL default '0',
  `serie_limit` int(11) NOT NULL default '0',
  `counter` int(11) NOT NULL default '0',
  `last_send` datetime NOT NULL default '0000-00-00 00:00:00',
  `next_send` datetime NOT NULL default '0000-00-00 00:00:00',
  `msg` varchar(100) NOT NULL default '',
  PRIMARY KEY (`owner`,`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

První tabulka „2nd_email“ spojuje uživatele definovaného řetězcem „owner“ s jeho emailovými adresami, které mají identifikační číslo v atributu „id“. Obdobně jsou provázána pravidla k jejich majitelům a k jednotlivým emailovým adresám. V UML modelu tyto tabulky odpovídají třídám „Eml“ a „Rule“.

7. Závěr

V praxi se podařilo vyvinout funkční WWW aplikaci, která je úspěšně nasazena do provozu a je nyní stále aktivní. Aplikace slouží k pravidelnému rozesílání upozorňovacích zpráv na mobilní telefony s informativním charakterem. Připomínají různé, uživatelsky definované, periodicky se opakující události. Zvoleným logickým způsobem zapisování pravidel pro odesílání zpráv lze zaznamenat prakticky jakoukoliv periodicky se opakující událost. Plně pokrývá potřeby všech uživatelů systému podle jejich rozmanitých požadavků. Vhodným příkladem jsou pravidla pro definování termínů aplikace hormonální antikoncepce (upřednostňováno především uživatelkami této aplikace). Umožňuje grafické znázornění termínů v budoucnosti, kdy zprávy budou posílány na předdefinovanou emailovou adresu. Aplikace se jeví v průběhu času jako stabilní, plně funkční a je hojně využívána. Aplikace je s úspěchem realizována a poskytována uživatelům zdarma.

Závěrem práce lze konstatovat, že zmíněným zobecněným postupem může být vyvinuta a naprogramována world wide web aplikace. Existuje velká řada standardů vhodných pro world wide web aplikace a jejich dodržení ulehčuje práci programátorům, kteří tyto aplikace tvoří.

Tvorbě kvalitních WWW aplikací slouží celá řada jazyků a norem, které svojí existencí a dodržováním zaručují určitý stupeň jakosti. Další rozvoj těchto jazyků a norem je klíčem k tvorbě rychlejších, bezpečnějších a stabilnějších aplikací. Proto se vyplatí tvořit aplikace přehledné a tím snadno udržovatelné.

Obecně formulovaný postup popsaný v této práci je možné uplatnit nejen na tvorbu webových aplikací. Zahrnuje myšlenky a poznatky, které mohou doprovázet různé reálné projekty. Vynáší na světlo aspekty, které by mohly být zanedbány při tvorbě projektu a následně by jejich opomenutí mohlo způsobit nežádoucí problémy. Samotným akceptováním těchto aspektů v průběhu vzniku aplikace je minimalizován prostor pro vznik chyb.

V současné době je po celém světě vyvinuto velké množství kvalitních i nekvalitních aplikací. Doplnují každodenní život lidí, kteří si na ně zvykli. V dnešní době je těžko představitelné, že by tyto aplikace zde nebyly. Lidstvo si na ně zvyklo a tím se aplikace jako takové staly nedílnou součástí života kulturních civilizací.

Aplikace mohou a ovlivňují lidskou psychiku a tento fakt by neměl být opomíjen. Rizikem aplikací pro simulace a zábavu je možný vznik závislosti, ke které jsou náchylní především mladí lidé. Z těchto důvodů by vývoj aplikací měl být korigován a neměl by umožňovat vznik aplikací s manipulativním charakterem. Aplikace mají být lidem užitečným pomocníkem, nikoliv zotročujícím nástrojem.

V historii vývinu aplikací lze vidět odraz vývoje civilizace a jejích požadavků na aplikace, které v dané době vznikly. Na jedné straně požadavky a na druhé straně nápady k řešení dávají podnět ke vzniku nových aplikací i technologií, kterými je svět zaplavován. Internet jako síť umožňuje propojovat lidi a jejich tvorbu nezávisle na zeměpisné poloze, vyznání či náboženství.

Programátoři věnující se WWW aplikacím musí být schopni vidět tuto problematiku v souvislostech. Měli by si uvědomovat citlivost dat, se kterými manipulují prostřednictvím jejich algoritmů. Musí zpravidla být otevřeni pro nové technologie, které se neustále rodí. Být ochotni se je učit a používat. Stále by měli mít na paměti, že vytváří aplikace pro rozličné lidi, z nichž někteří s tématem nemusí být tak dobře obeznámeni jako oni sami. Výsledky dnešní práce programátorů zůstávají dalším generacím jako jediný skutečný důkaz existence jejich tvůrců, který zde po sobě zanechají v podobě svého díla.

8. Seznam literatury

- [1] KOSEK, Jiří. *HTML tvorba dokonalých WWW stránek*. GRADA Publishing, a.s., 1998. ISBN 80-7169-608-0
- [2] KOSEK, Jiří. *PHP, tvorba interaktivních internetových aplikací*. GRADA Publishing, a.s., 14.04.1999. ISBN 80-7169-373-1.
- [3] ŠIMŮNEK, Milan. *SQL, kompletní kapesní průvodce*. GRADA Publishing, a.s., 24.03.1999. ISBN 80-7169-692-7.
- [4] McFARLAND, David Sawyer. *CSS, chybějící manuál*. GRADA Publishing, a.s., 10.08.2007. ISBN 978-80-247-2122-4.
- [5] PÍSEK, Slavoj. *HTML – tvorba jednoduchých internetových stránek*. GRADA Publishing, a.s., 2001. ISBN 80-247-1767-8.
- [6] FLANGAN, David. *JavaScript - Kapesní příručka - Pro aktivní webové stránky*. Helion, 2004. ISBN 80-7361-466-4.
- [7] ROSEBROCK, Eric – FILSON Eric. *Linux, Apache, MySQL a PHP - Instalace a konfigurace prostředí pro pokročilé webové aplikace*. GRADA Publishing, a.s., 23.08.2005. ISBN: 80-247-1260-1
- [8] OGame (Wikipedia) [online]. [Cit 19.4.2009]. Datum poslední revize: 11 April 2009, 18:28 (UTC). Dostupné na url: <http://en.wikipedia.org/wiki/OGame>
- [9] About the Apache HTTP Server Project [online]. [Cit 19.4.2009]. Dostupné na url: http://httpd.apache.org/ABOUT_APACHE.html
- [10] PHP: What is PHP? [online]. [Cit 19.4.2009]. Datum poslední revize: 17 April 2009. Dostupné na url: <http://www.php.net/manual/en/intro-what-is.php>
- [11] Zend Framework: About [online]. [Cit 19.4.2009]. Dostupné na url: <http://framework.zend.com/about/overview>
- [12] Rapid PHP Editor [online]. [Cit 19.4.2009]. dostupné na url: <http://www.blumentals.net/rapidphp/>

9. Přílohy

9.1. Skript index.php (jádro)

```

<?
// #####
// ##### Header ##### Created by jossss @ 2k9; jossss@badl.org #
// #####

Header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
Header("Last-Modified: ".GDate("D, d M Y H:i:s")." GTM");
Header("Pragma: No-cache");
Header("Cache-Control: no-cache");

// #####
require_once("olv.php"); // ##### Load $OLV(array)#####
// ##### Load fce paste(); ###
require_once($GLOBALS["OLV"]["basedir_functionsANDclasses"]."class_MySQL.php");
// #####
// #####
// ##### MySQL #####
if (($GLOBALS["OLV"]["mysql"]["comp"]!="") &&
($GLOBALS["OLV"]["mysql"]["username"]!="")) {
    $C_MySQL = new C_MySQL($GLOBALS["OLV"]["mysql"]["comp"],
                          $GLOBALS["OLV"]["mysql"]["username"],
                          $GLOBALS["OLV"]["mysql"]["userpass"],
                          $GLOBALS["OLV"]["mysql"]["startDB"]);
} else {
    if ($GLOBALS["OLV"]["mysql"]["show_err"]=="1") {
        echo "Nebyl uskutečněn pokus o navázání spojení s MySQL. Toto chybové hlášení
můžete vypnout v olv.php (mysql-show_err)";
    }
}
// #####

// #####
// ##### PAGE #####
// #####

Set_Time_Limit(0);

// naplneni ukolu / poslani zprav podle RULE tabulky
$q = "
SELECT
    r.owner,
    r.id,
    r.email,
    r.active,
    r.serie_celkem,
    r.serie_pozice,
    r.serie_mezi_jednotkami,
    r.serie_mezi_seriemi,
    r.serie_limit,
    r.next_send,
    r.msg,
    e.address,
    e.exp_dat
FROM
    2nd_rule As r
    JOIN 2nd_email as e ON
        r.owner = e.owner
    AND
        r.email = e.id
WHERE
    (
        r.enable_from IS null
    OR
        r.enable_from < NOW()

```

```

)
AND
(
  r.enable_until IS null
  OR
  r.enable_until > NOW()
)
AND
(
  r.next_send <= NOW()
)
AND
(
  r.next_send <> \"0000-00-00 00:00:00\"
)
;
";
$exec = @$C_MySQL->query($q);
while ($l = @$C_MySQL->next_row_array()) {
  $ZAZNAM[] = $l;
}

//////////////////////////////////// SVATEK
$GLOBALS["svatek"] = null;
function svatek($pokuscislo = 0) {
  $maxpocetpokusu = 5;
  Set_Time_Limit(30+$pokuscislo);

  if ($GLOBALS["svatek"] == null) {
    if ($GLOBALS["svatek"] == null) {
      sleep($pokuscislo);
      if ($f =
file("http://stmb.s.kvalitne.cz/svatky/index.php?inb=svatekdnes")) {
        Foreach ($f as $item) {
          $stream .= trim($item);
        }
        $string = Explode('<div class="svatky">', $stream);
        $string = Explode('<div class="napovdole">', $string[1]);
        $radky = Explode('<td valign="top">', $string[0]);
        $dnes = Explode("<b>", str_replace("/", "", strip_tags($radky[3],
"<b>"))));
        $zitra = Explode("<b>", str_replace("/", "", strip_tags($radky[4],
"<b>"))));
        $GLOBALS["svatek"] = trim($dnes[0]." ".$dnes[2]." ".$zitra[0]."
".$zitra[2]."");
      } else {
        if ($pokuscislo < $maxpocetpokusu) {
          $GLOBALS["svatek"] = $this->svatek($pokuscislo++);
        } else {
          $GLOBALS["svatek"] = "Svatek se nepodarilo najit.";
        }
      }
    }
    $GLOBALS["svatek"] = StrTr($GLOBALS["svatek"],
      "áäčďéěíňóřšťúúýžÁÄČĎĚĚĚÍŇÓŘŠŤÚÚÝŽ",
      "aacdeeeinoorstuuuyzAACDEEEINOORSTUUUYZ");
  }
  return $GLOBALS["svatek"];
}

//////////////////////////////////// SVATEK konec

For($i = 0; $i < Count($ZAZNAM); $i++) {
  $q1 = null;
  $q2 = null;
  $q3 = null;

  if ($ZAZNAM[$i]["serie_pozice"] == ($ZAZNAM[$i]["serie_celkem"] - 1)) {

```



```

        $new_next_dat = Date("Y-m-d H:i:s",
(strtotime($ZAZNAM[$i]["next_send"])+$ZAZNAM[$i]["serie_mezi_seriemi]));
        $new_serie_pozice = 0;
    } else {
        $new_next_dat = Date("Y-m-d H:i:s",
(strtotime($ZAZNAM[$i]["next_send"])+$ZAZNAM[$i]["serie_mezi_jednotkami]));
        $new_serie_pozice = $ZAZNAM[$i]["serie_pozice"] + 1;
    }
    $actual = $ZAZNAM[$i]["serie_pozice"] + 1;
    $q1 = "UPDATE 2nd_rule SET serie_pozice = ".$new_serie_pozice
        .", last_send = '". $ZAZNAM[$i]["next_send"]
        ."', next_send = '". $new_next_dat.'"';

    if (($ZAZNAM[$i]["active"] == "1") AND ($ZAZNAM[$i]["exp_dat"] > DATE("Y-m-d
H:i:s"))) {

        // a - aktualni
        $ZAZNAM[$i]["msg"] = Str_replace("%a%", ($actual), $ZAZNAM[$i]["msg"]);
        // z - zbyva
        $ZAZNAM[$i]["msg"] = Str_replace("%z%", ($ZAZNAM[$i]["serie_celkem"]-$actual),
$ZAZNAM[$i]["msg"]);
        // c - celkem
        $ZAZNAM[$i]["msg"] = Str_replace("%c%", ($ZAZNAM[$i]["serie_celkem"]),
$ZAZNAM[$i]["msg"]);
        // H
        $ZAZNAM[$i]["msg"] = Str_replace("%H%", (Date("H")), $ZAZNAM[$i]["msg"]);
        // i
        $ZAZNAM[$i]["msg"] = Str_replace("%i%", (Date("i")), $ZAZNAM[$i]["msg"]);
        // d
        $ZAZNAM[$i]["msg"] = Str_replace("%d%", (Date("d")), $ZAZNAM[$i]["msg"]);
        // m
        $ZAZNAM[$i]["msg"] = Str_replace("%m%", (Date("m")), $ZAZNAM[$i]["msg"]);
        // f / festival
        $ZAZNAM[$i]["msg"] = Str_replace("%f%", ucfirst(svatek()), $ZAZNAM[$i]["msg"]);

        $q1 .= ", counter = (counter + 1)";
        $q2 = 'INSERT INTO 2nd_history (dat, email, msg) VALUES(NOW(),
"'.$ZAZNAM[$i]["address"].'", "'.((trim($ZAZNAM[$i]["msg"])!="") ? $ZAZNAM[$i]["msg"] :
"NULL").'");';
        $q3 = "UPDATE 2nd_email SET counter = (counter + 1) WHERE owner LIKE
"'.$ZAZNAM[$i]["owner"]."' AND id = ".$ZAZNAM[$i]["email"]."';";

        // Odesilani zprav
        if (trim($ZAZNAM[$i]["msg"])!="")
            if (!(@mail(trim($ZAZNAM[$i]["address"]), "4u.zvyk.com", $ZAZNAM[$i]["msg"],
"From: 4u@zvyk.com")))
                echo "Error: ".trim($ZAZNAM[$i]["address"].', '.Date("Y-m-d
H:i:s")."<br>";
    }
    $q1 .= " WHERE owner LIKE "'.$ZAZNAM[$i]["owner"]."' AND id =
"'.$ZAZNAM[$i]["id"]."';";

    $exec = @$C_MySQL->query($q1);
    $exec = @$C_MySQL->query($q2);
    $exec = @$C_MySQL->query($q3);

    if (trim($q3)!="") writelog("".Date("Y-m-d H:i:s")."; ".$GLOBALS["C_Auth"]-
>_Auth_user_id."; ".$q3);
}

// posilani oznameni o vyprseni validity emailu
$HODINDopredu[] = 336;
$HODINDopredu[] = 168;
$HODINDopredu[] = 72;
$HODINDopredu[] = 48;
$HODINDopredu[] = 24;
$HODINDopredu[] = 12;
$HODINDopredu[] = 1;
For ($i = 0; $i < Count($HODINDopredu); $i++) {

```

```

    $q = "SELECT address
        FROM 2nd_email
        WHERE exp_dat LIKE '".DATE("Y-m-d H:i:", (time() + ($HODINDopredu[$i] * 60 *
60)))."%';";
    $exec = @$_MySQL->query($q);
    while ($l = @$_MySQL->next_row_array()) {
        if (!@mail(trim($l["address"]), "4u.zvyk.com", "U tohoto emailu vyprsi za
".$HODINDopredu[$i]." hodin/u validace, prodluzte si ji. 4u.zvyk.com", "From:
4u@zvyk.com")) echo "Error: ".trim($l["address"]).', '.Date("Y-m-d H:i:s")."<br>";
    }
}

// Propadla validace
$q = "SELECT e.owner as owner, e.id as id, e.address as address
    FROM 2nd_email as e, 2nd_rule as r
    WHERE
        e.owner=r.owner AND
        e.id=r.email AND
        e.exp_dat < now() AND
        r.active = '1'
    ";
$exec = @$_MySQL->query($q);
while ($l = @$_MySQL->next_row_array()) {
    $VALIDACE[] = $l;
}
for($i = 0; $i < Count($VALIDACE); $i++) {
    if (!@mail(trim($VALIDACE[$i]["address"]), "4u.zvyk.com", "U tohoto emailu vyprsel
a validace, prodluzte si ji. 4u.zvyk.com", "From: 4u@zvyk.com")) echo "Error:
".$VALIDACE[$i]["address"]').', '.Date("Y-m-d H:i:s")."<br>";
    $q = "UPDATE 2nd_rule SET active = '0' WHERE email = ".$VALIDACE[$i]["id"]." AND
owner LIKE '". $VALIDACE[$i]["owner"]."';";
    $exec = @$_MySQL->query($q);
}

// propadla validace u pokusu o registraci
$q = "DELETE FROM 2nd_w8_4_verify WHERE dat < '".DATE("Y-m-d H:i:s", Time()-
(24*60*60))."';";
$exec = @$_MySQL->query($q);

// SET active = 0 ((( limit )))
$q = "UPDATE 2nd_rule SET active = '0' WHERE serie_limit > 0 AND serie_limit <=
counter AND active LIKE '1';";
$exec = @$_MySQL->query($q);

Die();
?>

```

9.2.Skript class_MySQL.php

```

<?
// ### <B>C_MySQL</B>
// ### V1.1
// ### vytvoří MySQL spojení(ODBC) v prostredi PHP se serverem
// ### Created by jossss @ 2k3, jossss@centrum.cz

if (!define("C_MySQL_defin", "class")) return;

class C_MySQL {
    var $_MySQL_connection = 0;
    var $_MySQL_comp = "localhost";
    var $_MySQL_username = "";
    var $_MySQL_userpass = "";
    var $_MySQL_dbname = "";
    var $_MySQL_result = "";

    // konstruktor

```

```

function C_MySQL($comp, $username, $userpass, $dbname = "") {
    return $this->create_connection($comp, $username, $userpass, $dbname);
}

function create_connection($comp, $username, $userpass, $dbname) {
    $this->_MySQL_comp = $comp;
    $this->_MySQL_username = $username;
    $this->_MySQL_userpass = $userpass;
    $this->_MySQL_dbname = $dbname;
    if ($this->_MySQL_connection) {
        $this->close_connection();
    }
    return $this->connect($this->_MySQL_dbname);
}

function connect($db) {
    if (!(@$this->_MySQL_connection = MySQL_Connect($this->_MySQL_comp,
                                                    $this->_MySQL_username,
                                                    $this->_MySQL_userpass))) {
        die("\n<br>Neporadilo se navazat spojeni s MySQL.<br><br>\n");
    }
    if ($db != "") $this->select_db($this->_MySQL_dbname);
    return $this->_MySQL_connection;
}

function close_connection() {
    @MySQL_Close($this->_MySQL_connection);
    $this->_MySQL_connection = 0;
}

function select_db($db) {
    $this->_MySQL_dbname = $db;
    if ($this->_MySQL_connection) {
        return @MySQL_Select_DB($this->_MySQL_dbname, $this->_MySQL_connection);
    }
}

function query($query) {
    if ($this->_MySQL_connection) {
        @$this->_MySQL_result = MySQL_Query($query, $this->_MySQL_connection);
    }
    return $this->_MySQL_result;
}

function next_row_array() {
    return @MySQL_Fetch_Array($this->_MySQL_result);
}

function how_much_rows() {
    return @MySQL_Num_Rows($this->_MySQL_result);
}
}
?>

```

10. Seznam obrázků

Obrázek 1 - Schéma zobrazení série a cyklu	21
Obrázek 2 - UML - foryou.....	22

11. Seznam zdrojových kódů

Zdrojový kód 1 – index.php – výňatek z přílohy	26
Zdrojový kód 2 – index.php – výňatek z přílohy	27
Zdrojový kód 3 – index.php – výňatek z přílohy	27
Zdrojový kód 4 – index.php – výňatek z přílohy	28
Zdrojový kód 5 – index.php – výňatek z přílohy	29
Zdrojový kód 6 – index.php – výňatek z přílohy	29
Zdrojový kód 7 – index.php – výňatek z přílohy	30
Zdrojový kód 8 - MySQL definice tabulek	31