UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2009 Lukáš Šindelář

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Systém pro rozesílání začátku sportovních utkání

Lukáš Šindelář

Bakalářská práce

2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10.5.2009

.....

Jméno Příjmení

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš ŠINDELÁŘ**

Studijní program: **B2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Systém pro rozesílání začátku sportovních utkání**

Zásady pro vypracování:

Úkolem bude navrhnout systém pro automatické rozesílání začátků a místa sportovních utkání elektronickou poštou. Začátek a místo utkání je nutné odeslat rozhodčím, zástupcům klubu a řídícímu orgánu soutěže. Aplikace by měla obsahovat minimálně následující uživatele: * administrator - neomezený přístup * správce soutěží - zadává do databáze datum utkání, deadline pro zadání začátku a místa, pořádající družstvo a rozhodčí. * zástupce družstva - zadá čas a místo utkání v případě, že je pořádajícím družstvem
Cíle teoretické části: * zhodnocení současných systémů na trhu * zhodnocení současných technologií
Cíle aplikační části: * návrh vhodné databáze * návrh způsobu rozesílání začátků utkání * návrh způsobu řešení změny termínu utkání
* návrh rozesílání upomínek v případě, že zástupce družstva začátek a místo utkání nezadá.
* webová aplikace * manuál - instalace * uživatelský manuál

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Hugh E. Williams & David Lane - PHP a MySQL: Vytváříme webové databázové aplikace (Computerpress)

Vedoucí bakalářské práce: **RNDr. Josef Rak**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**

L.S.

doc. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

ANOTACE

Práce popisuje možnosti tvorby aplikace pro získávání a pozdější sledování a rozesílaní výsledků sportovních utkání. Práce zaměřuje na nástroje pro tvorbu dynamických www stránek, způsoby importu dat a periodické spouštění aplikací. Aplikace zajišťuje rozesílání informací o sportovních utkáních a kontrolu správného zadávání dat pomocí emailu.

KLÍČOVÁ SLOVA

sport, sportovní utkání, hlášenky sportovních utkání, sportovní výsledky

TITLE

The system for sending the beginning of sporting events.

ANNOTATION

The work describes the possibility of making an application for the acquisition and subsequent monitoring and distribution results of sport competitions. The work focuses on tools for creating dynamic websites, ways to import data and run applications periodically. The application provides information on the distribution of sporting events and checking the correct data entry by email.

KEYWORDS

sports, sport competitions, information report of sport competitions, sports results

Obsah

1.Úvod

Internet je v poslední době jedním z nejdůležitějších zdrojů informací. Část Internetu známá jako World Wide Web, tedy systém mezi sebou prolinkovaných stránek, přístupných pomocí prohlížeče, dnes nabízí možnost vytvoření vlastní webové prezentace. Každá webová prezentace má nějaký účel. Ať už je to poskytnutí informací, předání argumentů nebo třeba prodej výrobků, vždy je účel jasně dán. Samotný účel webu závisí jen na jeho tvůrci. Dle průzkumu serveru www.lupa.cz [1]. Dnes elektronicky komunikuje 83 procent lidí ve věku 15-30 let. S rostoucím počtem uživatelů rostou také nároky na dostupnost a kvalitu poskytovaného obsahu. Jednoduché stránky lze vytvořit za několik dní, složitější často projektují celé vývojové týmy. Specifickým druhem webových stránek jsou webové informační systémy (IS). Informační systémy jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. Takový systém pak mj. řeší problematiku autentizace, auditování, modularizace, serializace, oprávnění, proměnlivého datového modelu, personalizace, hiearchického zařazení uživatelů a samodokumentace. Obsahem této práce je takový systém navrhnout, implementovat a otestovat v reálném prostředí.

2.Zadání projektu

2.1. Obecné požadavky

Vytvořit internetové stránky pro automatické rozesílání hlášenek na sportovní utkání. Stránky budou přístupné jak pro zúčastněné kluby a organizace řídící soutěže tak i pro ostatní návštěvníky.

2.2. Architektura aplikace

Jedná se o databázovou aplikaci. V databázovém schématu je třeba zachytit tyto aspekty a požadavky na uložená data:

- Uložení informací o registrovaných uživatelích
- Uložení informací o oprávněních uživatele (uživatel návštěvník, uživatel správce soutěží, uživatel zástupce klubu, uživatel administrátor)
- Ukládání informací o utkáních, které zajišťuje zástupce klubu
- Ukládání informací o utkáních, které zajišťuje správce soutěží
- Uložení informací o soutěžích a o kategorii, v které ji uživatel najde
- Uložení historie

2.2.1. O klubech

- Stránky by měly představit každý tým, budou obsahovat informace o adrese klubu, adresách hracích míst, přehled soutěží hraných klubem
- Přítomna bude rozpiska zápasů a jejich výsledků.
- Budou vyplňovat místo a čas začátku utkání a výsledky utkání
- Při opomenutí bude zástupce klubu informován prostřednictvím e-mailu

2.2.2. O správcích soutěží

- Seznamy klubů, rozhodčích a kategorií pro daný sport jenž spadá pod daného správce
- Přidávají a odebírají jednotlivé kluby, rozhodčí a kategorie
- Přiřazují kluby do jednotlivých soutěží
- Plánují termínovou listinu sportovních utkání
- Delegují rozhodčí k utkání

2.2.3. O administrátorech

- Přidávají a odebírají jednotlivé správce soutěží

2.2.4. Grafické požadavky

- Jednoduchost, přehlednost
- Požadavek na manuál

2.2.5. Historie

- Ukládání veškerých dat, poskytovaných uživateli

3.Trendy vývoje webových stránek

3.1. Statické WWW stránky

Tvorba www stránek vychází ze struktury HTML, jedná se o text doplněný značkami, které používá prohlížeč k formátování výstupu [2]. Statické www stránky jsou nevhodné pro rychlou správu a údržbu obsahu, ale stále se s nimi v praxi setkáváme. Uživateli neumožňují vkládání dat (komentáře, diskuze apod.) s výjimkou uploudu souborů. Jejich ideální využití je pro čistě informační stránky, jejich vytvoření je jednoduché díky tzv. WYSIWYG [3] (What you see is what you get, česky „co vidíš to dostaneš“) editorům. Z nejznámějších například MS Word nebo PSPad, kde napíšete libovolný text a dokument uložíme jako webovou stránku.

3.2. Dynamické webové aplikace

Díky nutnosti generovat výslednou WWW stránku v závislosti na aktuálně dostupných informací/dat začaly vznikat skriptovací a programovací jazyky, které vygenerují obsah stránek až na žádost uživatele. Ve spojení s databázovým systémem tak vzniká nástroj pro vytvoření libovolného informačního systému.

Obecně lze prostředky pro tvorbu dynamických stránek rozdělit na dvě základní skupiny:

- serverové
- klientské

Serverové prostředky, nejčastěji skriptovací jazyky, umožňují vygenerovat unikátní obsah stránky odpovídající přesně dotazu uživatele. Vygenerovaný obsah je poslán na stranu klienta, kde je zobrazen. Na klientské straně jsou minimální nároky na hardwarové a softwarové vybavení. Do této skupiny patří například technologie PHP, ASP.NET, Python a další.

Mezi klientské prostředky můžeme zařadit například JavaScript. aby tato technologie mohla správně pracovat, musí být podpora s klientské stanice. Kód, který je součástí webových stránek se tedy vykonává až na straně uživatele, což zvyšuje nároky na schopnosti prohlížeče a zvyšují se nároky na výkon hardwaru.

4.Prostředky tvorby redakčního systému

Hlavní část této práce se věnuje realizaci redakčního systému za použití skriptovacího jazyka PHP ve spojení s databází MySQL [4]. Tato kapitola by měla přiblížit další možné jazyky a nástroje, které se dnes používají k tvorbě webových aplikací a na závěr porovnat jejich přednosti a nevýhody.

4.1. Jazyky a nástroje pro tvorbu dynamických www stránek

V současné době je trendem uchovávat a zveřejňovat informace na internetu prostřednictvím internetových stránek. Existuje mnoho jazyků a nástrojů pro tvorbu těchto WWW stránek. Základním stavebním kamenem je jazyk HTML. Tento jazyk by však pro tvorbu dynamických WWW stránek nestačil, proto se používají další prostředky a jazyky rozvíjející dynamičnost stránek. Tyto nástroje a jazyky můžeme rozdělit do dvou základních skupin. První skupinou jsou klientské, které pracují na straně klienta (interpret prohlížeče). Většinou se jedná o událostmi řízený běh programu. Druhou skupinou jsou jazyky, jejichž kód je vykonáván na straně serveru, jako je PHP, ASP a některé druhy JavaScriptu. Na straně serveru se velmi často nutné uchovávat data a proto je nutné seznámit se i s různými druhy databází, jako je například stále oblíbenější MySql.

4.1.1. HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext [5]. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.

Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Jazyk HTML je od verze 2.0 aplikací SGML (připravovaná verze HTML5 ale již závislost na SGML obsahovat nebude) [6]. Je charakterizován množinou značek a jejich atributů definovaných pro danou verzi. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky (<a>). Část dokumentu tvořená oteví-

rací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. element (prvek) dokumentu. Například je otevírací značka pro zvýraznění textu a Text tučně je element obsahující zvýrazněný text. Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu.

Značky (zvané tagy) jsou obvykle párové (v XHTML jsou párové všechny), přičemž koncové značka je shodná se značkou počáteční, jen má před názvem znak lomítko.

Dokument v jazyku HTML má předepsanou strukturu:

- Deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE.
- Kořenový element – element html (značky <html> a </html>) reprezentuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu).
- Hlavička elementu – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení a jiné informace o dokumentu. Hlavička je uzavřena mezi značky <head> a </head>. Element head je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.
- Tělo dokumentu – obsahuje vlastní text dokumentu. Vymezuje se značkami <body> a </body>. Element body je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

4.1.2. Druhy značek

Značky lze z hlediska významu rozdělit na tři základní skupiny [5]:

Strukturální značky rozvrhují strukturu dokumentu, příkladem jsou odstavce (<p>), nadpisy (<h1>, <h2>). Dodávají dokumentu formu.

Popisné (sémantické) značky popisují povahu obsahu elementu, příkladem je nadpis (<title>) nebo adresa (<address>). Současný trend je orientován právě na

sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk XML.

Stylistické značky určují vzhled elementu při zobrazení, typickým příkladem je značka pro tučné písmo (<b>). Od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu. Mezi důvody pro neužívání těchto značek patří především to, že tyto značky jsou orientovány na prohlížení na obrazovce počítače, příliš se však nepočítá s používáním dokumentu jiným způsobem – alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.

4.1.3. PHP

PHP je programovací jazyk, který pracuje na straně serveru. S PHP můžete ukládat a měnit data webových stránek. PHP původně znamená *Personal Home Page* a vzniklo v roce 1996, od té doby prošlo velkými změnami a nyní tato zkratka znamená *Hypertext Preprocesor* [7].

PHP není nijak těžké pochopit a už se základy si lze vystačit [8]. Umí ukládat, měnit a mazat data. Vše se odehrává na webovém serveru (kde jsou uloženy zdrojové kódy webových stránek). PHP skript se nejprve provede na serveru a potom odešle prohlížeči pouze výsledek (to je rozdíl oproti JavaScriptu, který počítá přímo v prohlížeči). Zdrojový kód PHP narozdíl od JavaScriptu a HTML nezobrazíte.

Webová stránka s prvky PHP má nejčastěji koncovku .php. Avšak je možné použít i .php3, php4, php5 a phtml. Nejlépe je používat koncovky .php.

Pomocí PHP je možné vytvořit diskusní fórum, knihu návštěv, počítadlo, anketu, graf a dokonce si pomocí jednoduchého kódu můžete zlikvidovat celý obsah webu. Navíc máte možnost propojit vaše stránky s databázemi, např. MySQL.

Je rozhodně alespoň jedna funkce PHP, která se hodí snad do každého webu [9]. Na webových stránkách se obvykle opakují některé části, hlavička s odkazy, menu, patička. S PHP si můžete snadno vytvořit šablonu pro web, do které se budou vkládat soubory s menu, patičkou atd. Můžete tedy mít menu jen jednou zapsané a do dalších stránek ho pouze kopírovat. Až budete chtít menu změnit bude to nesmírně jednoduché.

4.1.4. ASP.NET

Active Server Pages je technologie nezávislá na programovacím jazyce, která umožňuje vykonávání kódu na straně serveru a následné odeslání výsledku uživateli [10]. To znamená, že webová stránka s příponou **.asp** obsahuje kód, který se vykoná na IIS serveru. Ten prohlížeči odešle pouze výsledek ve značkovacím jazyce HTML, který tento html kód umí bez problému zobrazit. Programovací jazyky, které se nejvíce používají: VBScript a JScript. ASP pak tedy není ničím jiným než VBScriptem nebo JScriptem, jenž běží na serveru a generuje výsledný kód v HTML, kterému rozumí každý prohlížeč.

ASP.NET je nadstavba .NET Frameworku firmy Microsoft pro tvorbu webových aplikací a služeb. Je nástupcem technologie ASP. ASP.NET je založen na CLR, který je sdílen všemi aplikacemi postavenými na *.NET Frameworku* přímým konkurentem JSP (Java Server Pages).

Programátoři tak mohou realizovat všechny své projekty v jakémkoliv jazyce podporujícím CLR, např. Visual Basic.NET, JScript.NET, C#, Managed C++, ale i mutace Perlu, Pythonu a další [11]. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika DLL souborů.

Koncept ASP.NET WebForms ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je tedy možné používat ovládací prvky jako tlačítko (Button), nápis (Label) a další.

4.1.5. MySql

MySQL je multiplatformní *databáze* [12] jenž je založena na komunikaci za pomocí jazyka SQL. Přesněji se jedná o dialekt tohoto jazyka s některými rozšířeními. Existuje ve dvou licencích, GPL nebo komerční licence. Právě díky bezplatné GPL licenci se tento druh databází ujal u mnoha uživatelů. Vyznačuje se snadnou implementovatelností a výkonem. MySQL bylo od začátku vyvíjeno především s ohledem na rychlost, i za cenu některých zjednodušení, jako například jednoduchého způsobu zálohování. MySql také až donedávna nepodporovalo pohledy, uložené procedury a databázové triggery. Tyto vlastnosti byly implementovány až u posledních verzí (od verze 5.0). MySql ukládá data do databázových tabulek, které se

liší svým použitím, možnostmi a způsobem ukládání dat do souborů. Nejpoužívanějším způsobem je MyISAM, který nepodporuje transakce, dále pak InnoDB, které transakce podporuje. Dále to jsou BerkeleyDB, MEMORY, NDB Cluster (od verze 5.0), ARCHIVE a CVS pro ukládání dat v prostých textových souborech.

4.1.6. Zhodnocení možných nástrojů pro tvorbu

ASP nebo PHP musíme si zaprvé uvědomit že většina webů běží na PHP. Programovací jazyk ASP je svými možnostmi velmi podobný jazyku PHP. ASP je též serverový jazyk, tzn. umožňuje pracovat s databází, ukládat data, dynamicky generovat webové stránky. Rovněž je potřeba webový server např. Microsoft ISS.

ASP je vyvíjeno Microsoftem, oproti tomu PHP je opensource - tj. zdrojové kódy jsou každému k dispozici a každý pokročilý programátor si jej může dle libosti upravit. Jazyk se díky tomu může snadno dál rozvíjet.

ISS - server podporující ASP rovněž není zadarmo (jako Apache pro PHP). Problém může nastat až budete hledat webhosting zdarma pro ASP stránky. Nabízí se ASPweb a ASP2, jinak webhosting zdarma a webhosting vůbec je těžší sehnat pro ASP než pro PHP.

Pokud se rozhodujete pro použití PHP nebo ASP, mějte na paměti, že pro malé a běžné stránky je vhodnější PHP. ASP je vhodné jedině pro obrovské weby. V ASP prostředí lze používat další doplňky (.NET aplikace) a tak vytvořit aplikaci, která se spíše blíží klasickým programům.

MySQL je primitivní databáze, jednoduchý filesystém, která je v konečném důsledku pomalá a nepodporuje správné programátorské zvyky a postupy. Ona MySQL totiž není databáze. Je to jednoduchý filesystém. Kdyby se lidi weboví chtěli naučit trošku více SQL a PL/SQL a začali používat (taky open-source) PostgreSQL či Firebird, namísto MySQL, řada složitých aplikací by běžela mnohem rychleji. Někomu prostě primitivnost MySQL vyhovuje. Je třeba si ale také uvědomit že 99% freewebu a dokonce velký počet komerčních webhostingu nabízí na linuxové platformě pravě toto.

4.2. RSS Kanál

RSS je rodina XML formátů určených pro čtení novinek na webových stránkách a obecněji syndikaci obsahu [13].

Technologie RSS umožňuje uživatelům Internetu přihlásit se k odběru novinek z webu, který nabízí RSS zdroj (*RSS feed*, též RSS kanál, *RSS channel*). Tento zdroj se většinou vyskytuje na stránkách, kde se obsah mění a přidává velmi často.

Původně tento formát sloužil pouze k předávání aktuálních novinek mezi jednotlivými servery, které se takto jednoduše mohly odkazovat na aktuální články na jiných serverech.

RSS formát poskytuje obsah celého článku, případně jeho část, odkaz na původní článek a také jiná metadata. Tyto informace jsou posílány jako XML soubor nazývaný RSS zdroj, webový zdroj, RSS stream, RSS feed nebo RSS kanál.

4.2.1. Možnosti použití RSS kanálu

Software určený k práci s RSS kanály se označuje jako *RSS čtečka*. Může se jednat o samostatný specializovaný program, o plugin do jiného programu (typicky webového prohlížeče) tato funkce může být v jiném programu rovnou vestavěna, případně se může jednat o webovou aplikaci poskytující tuto funkčnost (např. Google Reader či NetVibes) [14].

Pokud některý webový server nabízí RSS kanály, obvykle to indikuje ikonkou, která vede přímo na URL příslušného zdroje, který uživatel zadá do čtečky a ta poté zobrazí seznam všech takto zpřístupněných článků (či jiných odkazů). Čtečka pak pravidelně kontroluje toto URL a zobrazuje nové položky.

4.2.2. Atom (standard)

Atom (plným názvem *Atom Syndication Format*) je webový standard pro publikování syndikovaného obsahu [15]. Je nástupcem formátu RSS. Kromě něj je přijat také *Atom Publishing Protocol* (zkráceně APP či AtomPub) umožňující vytváření a aktualizaci webových zdrojů ve formátu Atom pomocí HTTP.

Oproti RSS i jednotlivé položky musí mít uvedeny svůj název, jedinečný identifikátor (URI) a datum poslední změny. Povinné je rovněž jméno autora u každé položky, pokud není vyplněno pro všechny z nich. Zatímco u RSS bylo možné v poli

description uvést jak souhrn, tak plný obsah, u Atomu je pro souhrn vyčleněn element *summary*, zatímco pro plný obsah se používá *content*. V RSS nebylo možné zvolit, jaký formát obsahu je použit. Atom rozeznává např. prostý text, escapované HTML, XHTML, XML binární obsah v kódování Base64 či odkaz na jiný webový zdroj.

4.2.3. Přehled RSS čteček

I když ke čtení RSS lze použít běžný internetový prohlížeč, k naplnění účelu RSS kanálů je samozřejmě více než vhodné využít speciální čtečku [16].

Čtečka	RSS Point	RSS Tracker	Blog Ex-press	Feed Demon	Feed Reader-CZ	Sharp Reader	Abilon
Podpora RSS	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Podpora Atom 0.3 Feed	Ano	Ne	Ano	Ano	Ano	Ano	Ano
Import z OPML	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Export do OPML	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Řazení kanálů do složek	Ano	Ne	Ano	Ne	Ano	Ano	Ano
Integrovaný prohlížeč	Ano	Ano	Ano	Ano	Ano	Ano	Ano

Tabulka 01 – přehled RSS čteček

4.2.4. Integrovaná čtečka

RSS čtečka se pomalu ale jistě stává standardním nebo alespoň volitelným doplňkem řady jiných programů. RSS čtečku mají integrovanou prohlížeče Opera a Netscape nebo nadstavby pro Internet Explorer Maxthon a Deepnet Explorer.

RSS čtečkou disponuje i e-mailový klient Thunderbird, dvojka mezi internetovými prohlížeči Firefox zase nabízí tzv. živé záložky, které lze rovněž považovat za RSS čtečku. Pro Firefox ale existují lepší RSS čtečky více se blížící klasickým čtečkám, ve formě rozšíření.

K dalším programům majícím RSS čtečku jako doplněk patří Deskop Sidebar nebo s patřičným plug-inem i univerzální instant messaging klient Miranda.

5.Realizace projektu

Každý sport má svoje specifické pravidla. Aby náš systém mohl být kompatibilní ze všemi druhy sportů musíme uchovávat pouze ty informace které sou společné všem sportům.

5.1. Vymezení potřeb systému

Rozesílání informací o jednotlivých utkáních soupeřům, rozhodčím a řídícím orgánům. Vkládání výsledků utkání a kontrola jejich správnosti. Upozornění na zapomenutí vložení údajů o utkání, nebo opomenutí vložení výsledku utkání do daného data.

5.2. Výběr vhodného Webhostingu

Výběr vhodného webhostingu je jedna ze základních věcí při tvorbě každé internetové aplikace [17]. Implementace se musí přizpůsobit tomuto webovému serveru. Nejedná se zde jenom o to, zda je nebo není na stránkách umístěna reklama, ale hlavně o použité technologie. Každý webový server poskytuje různé technologie a různou podporu. Proto jako první věc musíme najít vhodný webhosting.

5.2.1. Webový server

Webhostingové služby znamenají, když někdo nabízí místo a s tím spojené služby na serveru. Tento server je připojen k velmi rychlému internetu. Každý zákazník má na serveru vyhrazeno určité místo. Limity se nejčastěji pohybují u placených hostingů v jednotkách gigabajtů. Mezi další služby nabízené hostincem patří nejčastěji e-mail, ftp, různé statistiky atd.

Webový server přijímá požadavky od klienta (pomocí protokolu HTTP). Tyto požadavky vyřizuje a vrací odpověď, kterou je nejčastěji HTML stránka. Ta je buď předem připravená (statický obsah), nebo se vytváří podle požadavků klienta (dynamický obsah). Statické stránky jsou vráceny rychleji. Dynamické stránky ale mohou lépe reagovat na klientské požadavky. V praxi se oba způsoby kombinují.

5.2.2. Výběr webhostingových služeb

Na začátku si vybereme vlastní doménu u placeného hostingu, pokud se rozhodneme pro hosting neplacený vybereme si jednu z nabízených domén. Dalším požadavkem je spouštění skriptu v daném časovém intervalu. To se ukázalo jako největší problém u těchto poskytovatelů a značně to zúžilo možný výběr. Mé požadavky byly:

- PHP (PHP ve verzi 5)
- MySQL (PHPMyAdmin)
- Mod_rewrite (optimalizace URL pro SEO)
- Minimum 50 MB místa
- E-mail
- CRON (nutnost spouštění skriptů v daném intervalu)

5.3. Řešení komunikace systému s uživateli

Systém komunikuje prostřednictvím e-mailu. Veškerá komunikace probíhá automaticky na základě vložených dat.

5.4. Rozdělení uživatelských rozhraní

Pokud by zmíněné funkce měly být implementovány do jediného rozhraní, stejně by bylo nutné zahrnout mnoho speciálních příkazů pro vybrané uživatele, kterým mělo být umožněno provádět úpravy. Pokud mají být tyto funkce v rukou určité skupiny uživatelů je výhodné rozdělit projekt do více odlišných rozhraní. Lze tak nastavit rozdílnou úroveň zabezpečení a vzhledu obou částí.

5.4.1. Uživatelské role

V systému byly identifikovány čtyři základní uživatelské role. Liší se právy pro přístup k administrátorskému rozhraní a dalšími právy.

5.4.2. Neregistrovaný uživatel

Je uživatel který se systému neprokázal platnými přihlašovacími údaji. Jeho přístup je omezen pouze na veřejnou část webu. Smí prohlížet jednotlivé stránky, ale není mu dovoleno nijak přidávat data.

5.4.3. Registrovaný uživatel

Uživatel který se zaregistroval a prokázal platnými přihlašovacími údaji. V našem případě se jedná o kluby. Kromě práv neregistrovaného uživatele má možnost přidávat čas a místo utkání a výsledky utkání.

5.4.4. Správce soutěží

Uživatel který se zaregistroval a byla mu administrátorem přidána správcovská práva. Po přihlášení má všechna práva jako neregistrovaný člen a navíc mu je umožněno vytvářet nové kategorie sportů, přidávat kluby a rozhodčí. Provádět rozlosování zápasů a nastavit datum utkání.

5.4.5. Administrátor

Uživatel které mu je umožněno přidávat nové správce.

5.5. Historie komunikace

5.5.1. Korespondence

Potřeba komunikace mezi kluby, rozhodčími a řídícími orgány soutěže existovala samozřejmě již dříve, za první komunikační prostředek můžeme považovat korespondenci.

Shrňme její základní výhody:

- snadná dostupnost
- jednoduchá manipulace
- přehlednost

Nevýhody korespondence:

- časová náročnost
- využití placené služby
- problém z aktualizací informací

5.5.2. Webové stránky

Obsah stránek byl přístupný všem, kteří znali jejich adresu.

Základní výhody:

- využívá internetu
- stránky přístupné odkudkoliv přes prohlížeč

Nevýhody:

- aktualizace vyžadovala spoustu času a znalost psaní stránek
- četné úpravy kódu vedly k chybám
- obsluhu mohl provádět pouze člověk s oprávněním

5.5.3. Email

Složitá aktualizace a úprava stránek vedla k přetíženosti správce, který tak prováděl aktualizace pouze několikrát za rok. S odstupem času se ukázalo, že je nutné ukončit výměnu informací pomocí korespondence a začít využívat lepší a efektivnější nástroj komunikace. Nástupcem korespondence je e-mail. Pro každého návštěvníka stránek je jednodušší, když za informacemi nemusí chodit on, ale informace chodí za ním. Máte jistotu, že email s nějakou novinkou nebo zprávou dojde tomu, kdo o něj má zájem.

Výhody e-mailu:

- časová nenáročnost
- zdarma
- jednoduchá a rychlá možnost aktualizace informací
- vzdálený přístup

Nevýhody emailu:

- časté odesílaní informací

5.5.4. Systém automatického rozesílání hlášenek

Stále tedy zůstává nutnost fyzické přítomnosti při odesílání e-mailů. Je potřeba najít způsob jak náročnost agendy ještě více rozložit a zjednodušit. Výsledkem je návrh webu, který by minimalizoval nutnost obsluhy a pracoval by nezávisle pouze

na základě vložených dat. Řešení se nabízí v podobě využití znalostí databází, HTML a PHP.

Je nutné upřesnit konkrétní požadavky na to, jak by měl nově vzniklý systém vypadat a co by měl svým uživatelům nabízet. Základním požadavkem je funkčnost a jednoduchost. Systém je určen pro kolektivní sporty a bude uchovávat veškeré nutné informace o jednotlivým klubech jakožto uživatelích a řídících orgánech jakožto správcích.

Systém umožňuje vkládání informací o klubech, utkání a rozhodčích. Veškeré nutné data pro chod systému budou vkládány přes webové rozhraní. Stránky budou zároveň určeny i veřejnosti k prohlížení výsledku jednotlivých sportovních utkání.

Výhody automatického rozesílání hlášenek:

- téměř nulová časová náročnost
- rychlá a jednoduchá možnost aktualizace informací
- uživatel vkládá pouze nutné minimum dat

5.5.5. Shrnutí možností komunikace

Systém automatického rozesílání hlášenek uživateli značně ulehčuje práci. Většinu práce kterou musel dříve vykonávat, za něj provede systém. Ale vraťme se na začátek. Korespondence zde musel zástupce klubu zadávat veškerá data sám, musel znát adresu příjemce a toto vše v několika kopiích a toto samé platí i pro správce soutěží. Webové stránky zde naopak je režie zástupce klubu minimální a naopak je zde obrovská kumulace práce pro správce webu na začátcích jednotlivých sezon a se zadáváním údajů o sportovních utkání (čas, místo, výsledek apod.). Jisté zjednodušení přichází až z emailem a kombinaci webu a emailu. Zde už je vše o něco jednodušší a hlavně rychlejší. Informace lze předávat během několika vteřin o čem se nám dříve mohlo nechat jenom zdát. Stále ale zástupce klubu potře buje znát některé informace (adresář emailových adres) aby tento systém mohl fungovat. Naprosté zjednodušení přináší Systém automatického rozesílání hlášenek. Jedinou informaci, kterou je potřeba znát je adresa webu a přihlašovací údaje. Časová náročnost obsluhy bude v řádech minut a to v průměru jednou týdně pro každého uživatele, ale záleží na četnosti utkání. Jedinou kumulaci práce má správce zápasů při vytváření rozpisů utkání, které se obvykle provádí jednou až dvakrát během sezony. Zástupci klubu

vkládají pouze čas a místo utkání a výsledky. O komunikaci se v maximální míře stará systém sám a nepotřebuje žádnou obsluhu.

Závěrem jasně vyplývá, že systém automatického rozesílání hlášenek obrovským způsobem ulehčuje práci jak správcům soutěží, tak hlavně zástupcům klubů. Zároveň systém kontroluje zda všichni uživatelé vložili data potřebná pro chod systému v čas a pokud ne informuje je o tom.

5.6. Možnosti rozesílání e-mailu

5.6.1. Nastavení emailu

K odeslání nám slouží funkce - *mb_send_mail* [18]. Pokud před ni uvedeme zavináč, zabráníme případné chybové PHP hlášce, která je nevzhledná.

Parametry funkce jsou: příjemce, předmět, zpráva a nepovinně hlavičky:

```
mb_send_mail($to, $subject, $message, $from)
```

Nemusíte vkládat jen přes proměnné, můžete napevno stanovit kteroukoliv z těchto položek.

- příjemce

```
$to = "jmeno@server.cz"
```

Pokud je více příjemců, oddělujeme čárkami.

- předmět

```
$subject = "zpráva z WWW – ".$hlavicka." "
```

Můžete zadat jen název proměnné; nemusíte psát nějaký další text.

- zpráva

```
$message = $zprava
```

- odesílatel

```
$from = "from: ".$odesilatel." "
```

5.6.2. Odeslání e-mailu

Funkci MB_SEND_MAIL zkombinujeme s funkcí IF, protože potom máme pod kontrolou hlášky při (ne)odeslání zprávy. Viz. následující kód:

```
function email($prijemce,$hlavicka,$zprava,$odesilatel) {

$to = $prijemce;
$subject = $hlavicka;
$message = $zprava;
$from = "from: ".$odesilatel." \n";

mb_language('Neutral');
mb_internal_encoding("UTF-8");
mb_http_input("UTF-8");
mb_http_output("UTF-8");
if( mb_send_mail($to, $subject, $message, $from) )
  {echo 'OK';}
else
   {echo 'CHYBA';}

}
```

5.7. Periodické spouštění aplikací

Další částí, kterou se zabývá tato práce, je spouštění skriptů, či aplikací.. Periodické spouštění aplikací je důležité v situacích, kdy chceme v pravidelných intervalech extrahovat informace z HTML stránek. Jedním z nejčastějších řešení tohoto problému, bez ohledu na způsob implementace, je periodický časovač *cron*. Tímto programem a dalšími možnostmi se podrobně zabývá tato kapitola.

5.7.1. Periodický časovač Cron

Téměř v každém vyspělém webovém systému je nutné používat periodicky naplánované akce. V mém případě to jsou situace, kdy kontroluji jestli sou vložena všechna potřebná data. K tomuto účelu je možné použít periodický časovač. Démon *cron* je systémový nástroj pro Unix a Linux. Tento démon spouští v předem naplánovaných časech a intervalech zaregistrované aplikace.

Cron však není určen pouze pro webové systémy, je jej možné použít pro spouštění všech aplikací a příkazů [19]. U webhostingu, kde se *cron* také hojně využívá, nebudeme mít k programu přímý přístup. Pokud však pracujeme na vlastním systému, máme přístup ke *cronu* přímo přes příkazovou řádku nebo přes jeho grafic-

kou nadstavbu. Program však neumí spouštět aplikace či skripty v intervalu kratší jedné minuty. V těchto případech je výhodnější nechat aplikaci běžet permanentně.

Pro nastavení spuštění aplikace pomocí *cronu* můžeme využít jednu ze dvou možností. První spočívá v tom, že nakopírujeme aplikaci do jednoho z těchto adresářů */etc/cron.daily*, */etc/cron.hourly, /etc/cron.weekly* a */etc/cron.monthly*. Poté je aplikace spouštěna v intervalech, jež vyplývají z názvu adresářů. Tento způsob je jednoduchý, ale nevýhodný v případech, kdy požadujeme pouze jedno spuštění nebo spouštění v jiném čase nebo jiných intervalech. Proto existuje další varianta. Pro tuto druhou možnost nastavování je nutné použít utilitu, která se nazývá *crontab*. Je to vlastně seznam úloh pro démona *cron*. Každý uživatel má přístup ke svému vlastní tabulce. Přistoupit k této tabulce můžeme pomocí textového editoru, nebo pomocí příkazu crontab -e. Záznam v tabulce se skládá ze šesti údajů oddělených mezerami nebo tabulátorem. Pořadí a hodnoty jednotlivých záznamů jsou popsány na konci tohoto odstavce. Pokud chceme říci, že se má aplikace spouštět každou hodinu, nahradíme v záznamu číslo hodiny symbolem "*****" (hvězdička). Pokud chceme aby byla aplikace spuštěna dvakrát do hodiny, nastavíme minuty spuštění a oddělíme je pomocí znaku "**,**" (čárka). Jednotlivé záznamy jsou od sebe odděleny pomocí konců řádku. Takto to funguje se všemi údaji v tabulce.

- minuta (0 – 59 nebo *)
- hodina (0 – 23 nebo *)
- den v měsíci (1 – 31 nebo *)
- měsíc (1 – 12 nebo *)
- den v týdnu (0 – neděle ... 6 – sobota)
- cesta k aplikaci, která se spustí

5.7.2. Ukázka několika záznamů v Crontab

0 3 * * * /bin/php -f /www/spravaspotrotnichutkani/skript.php
- vždy ve tři hodiny ráno spustí *skript.php*

0,30 3,5 * * * /www/spravaspotrotnichutkani/skript.php
- vždy ve tři hodiny, v půl čtvrté, v pět a v půl šesté ráno spustí *skript.php*

0 2 3 * */program/save_me
- vždy třetí den v měsíci ve dvě hodiny ráno je spuštěn program *seve_me*

* 16 * * * wget http://www.spravaspotrotnichutkani.cz

- každou minutu v od čtyř do pěti odpoledne každý den spustí program *wget*

Problém však nastává při spouštění PHP skriptů. Aby bylo možno spouštět skript jako v prvním ukázkovém záznamu (pomocí PHP interpretu), je nutno, aby na serveru bylo kompilováno CGI (tedy existuje spustitelný program PHP). Pokud tomu tak není, je možno použít program *wget*, který skript spustí podobně jako internetový prohlížeč. Toto řešení však není ideální, protože skript příliš vytěžuje webový server a také se uplatňuje timeout, který může skript předčasně ukončit. Spouštění z příkazové řádky je rychlejší a efektivnější. Je-li tedy kompilováno CGI je možno spouštět PHP skripty jako v prvním ukázkovém záznamu, bez nutnosti zásahu do skriptu. Druhou možností, kterou demonstruje druhý ukázkový záznam, je přidat specifikaci "#!/bin/php" do skriptu. Potom se skript chová jako spustitelný.

Pokud chceme používat Cron pro internetové aplikace a nemáme k dispozici přístup k příkazové řádce serveru a tudíž i k tabulce záznamů, můžeme například použít *cron* na jiném serveru nebo počítači, který má přístup do sítě internetu a spouštět tento skript pomocí programu *wget*. Nicméně většina poskytovatelů webhostingu *cron* zpřístupňuje, ale jeho nastavování je individuální.

5.7.3. Další možnosti periodického spouštění aplikací

Pokud nemáme *cron* k dispozici, a chceme využívat jeho služby pro spouštění PHP skriptů, můžeme využít služby *WebCron*. Tato služba je volně dostupná a je možné ji nastavovat přes přehledné webové rozhraní, to je dostupné na adrese *http://www.webcron.org/*.

Periodický časovač je výhodný, ale nelze ho vždy použít. Jedním z dalších možností je spuštění aplikací pomocí plánování úloh v OS Windows. Další možností je permanentně spuštěná aplikace. To se hodí pro velmi krátké intervaly (méně než jedna minuta). čekání mezi akcemi je možno zajistit například pomocí uspání aplikace pomocí příkazu a metod jako jsou sleep, pause nebo delay. Například v jazyku C# .NET 2.0 je možno použít třídu Timer, které za pomocí delegáta přiřadím funkci, jež bude spuštěna v pravidelných intervalech. Podobná komponenta je k dispozici například i v C++, Borland Delphi nebo Borland Builder C++.

5.7.4. Plánování úloh v OS Windows

Jedním z řešení je použít hostingu např. *http://www.ASpone.cz* [20]. ASPone jako jediný ASP.NET 3.5 freehosting zdarma nabízí všem možnost vyzkoušet si no-

vý Windows Server 2008 zdarma. Tato aplikace nabízí možnost pravidelného spouštění daných stránek na webovém prostoru ve zvolený čas. Díky této aplikaci můžete v pravidelných intervalech rozesílat novinky svým zákazníkům, partnerům nebo například aktualizovat data v databázi.

Aplikace umožňuje nadefinování několika druhů úkolů:

- **Periodické** - pravidelné spouštění úkolů v určeném časovém intervalu každou hodinu.
- **Každý den** - pravidelné spuštění úkolů v danou hodinu a minutu.
- **Týdenní** - pravidelné spuštění úkolů v daný den v týdnu, hodinu a minutu.
- **Každý měsíc** - pravidelné spuštění úkolů v daný den v měsíci, hodinu a minutu.

6.Databázový model

6.1. Vymezení pojmů

6.1.1. Databázový systém

Databáze (DB) je kolekce trvalých informací (dat) uložených na paměťovém médiu [21]. Softwarové prostředky pro správu a přístup k uloženým datům jsou označovány jako Systém řízení báze dat (DBMS) a slouží pro řízení přístupu k datům a správu reprezentace dat na úrovni paměťového média. DB a DBMS tvoří databázový systém.

Databázový systém zahrnuje:

- datové prvky – elementární hodnoty
- vztahy mezi prvky dat – složitější datové struktury, které tvoří jednotlivé záznamy
- integritní omezení – podmínky, které musí data uložená v databázi splňovat
- logické schéma – popis záznamů a jejich vazeb z uživatelského hlediska
- fyzické schéma – řeší uložení dat na paměťovém médiu

V současné době jsou nejvíce rozšířené relační a objektově-relační databázové systémy.

6.1.2. Relační databáze

Počátkem 70 let vznikají první databáze, které pohlížejí na data jako na tabulky [22]. Kolem roku 1974 vzniká první verze dotazovacího jazyka SQL (Structured Query Language), který byl vyvinut právě pro komunikaci s *relačními databázemi*. Pomocí SQL popisujeme, jaká data chceme získat a nikoliv jakým způsobem. Výsledky dotazů provedených nad relační databází se vrací v podobě *tabulek*.

6.2. Databáze

Základem informačního systému je databáze. Po prostudování požadavků na systém jsem provedl návrh databáze. Ta se skládá z tabulek jejichž názvy a krátký

popis je uveden v následující kapitole. Základními nosnými tabulkami jsou UZIVATELE a KATEGORIE. U nich většina relací začíná.

6.2.1. Tabulka UZIVATELE

- UZIV_JMENO - primární klíč tabulky a přihlašovací údaj
- JMENO - reálné jméno
- PRIJMENI - reální příjmení
- MESTO - informace o adrese
- ULICE - informace o adrese
- CP - informace o adrese
- PSC - informace o adrese
- TELEFON - telefonní kontakt, nemusí být uveden
- EMAIL - emailový kontakt, musí být unikátní
- HESLO - přihlašovací heslo formou md5 hashe
- ID_ROLE - cizí klíč do tabulky ROLE
- ID_MISTA - cizí klíč do tabulky MISTA
- VLASTNÍK - uživatelské jméno správce kontaktu, pro administrátora a správce soutěže je jeho hodnota NULL

6.2.2. Tabulka KATEGORIE

- ID_KATEGORIE - primární klíč
- NÁZEV_KATEGORIE - název kategorie
- ID_NADKATEGORIE - cizí klíč pro spojení sama na sebe a vytvoření struktury kategorií
- VLASTNIK - uživatelské jméno správce soutěží, který danou kategorii vytvořil

6.2.3. Tabulka ROZPIS_ZAPASU

- ID_ZAPASU - primární klíč
- JMENO_DOMACI - uživatelské jméno domácího týmu
- JMENO_HOSTE - uživatelské jméno hostujícího týmu
- ID_KATEGORIE – cizí klíč pro spojení s tabulkou kategorie
- CAS - čas konání utkání
- DATUM - datum konaní utkání
- MISTO - místo konání utkání

6.2.4. Tabulka ROZPIS_SOUTEZI

- ID_SOUTEZE – část primárního klíče, identifikace soutěže
- UZIV_JMENO – část primárního klíče, identifikace uživatele

6.2.5. Tabulka ROZHODCI_ROZPIS

- ID_ZAPASU - část primárního klíče, identifikace zápasu
- EMAIL_ROZHODCI - část primárního klíče, identifikace rozhodčího

6.2.6. Tabulka ROZHODCI

- EMAIL - primární klíč tabulky
- JMENO - reálné jméno
- PRIJIMENI- reální příjmení
- MESTO - informace o adrese
- ULICE - informace o adrese
- CP - informace o adrese
- PSC - informace o adrese
- TELEFON - telefonní kontakt, nemusí být uveden

- VLASTNIK - uživatelské jméno správce soutěží, který daného rozhodčího vytvořil

6.2.7. Tabulka HRACI_MISTA

- ID_MISTA - primární klíč tabulky
- MESTO - informace o adrese
- ULICE - informace o adrese
- CP - informace o adrese
- PSC - informace o adrese
- UZIV_JMENO - uživatelské jméno klubu, který si dané hrací místo zaregistroval

6.2.8. Tabulka VYSLEDKY

- ID_ZAPASU - cizí klíč do tabulky ROZPIS_ZAPASU
- SKORE_DOMACI - body domácí
- SKORE_HOSTE - body hosté
- CAS_VLOZENI - datum a čas vložení záznamu do tabulky

6.2.9. Tabulka VYSLEDKY_POM

- ID_ZAPASU - cizí klíč do tabulky ROZPIS_ZAPASU
- SKORE_DOMACI - body domácí
- SKORE_HOSTE - body hosté
- UZIV_JMENO - uživatelské jméno týmu, který daný záznam vložil

6.2.10. Tabulka ROLE

- ID_ROLE - primární klíč
- JMENO_ROLE - jméno role(administrator – přidává a maže správce soutěží, správce soutěží – administruje soutěže, zástupce klubu – spravuje vlastní zápasy)

6.3. Navržené databázové schéma:

Viz Příloha 1 - Obrázek 01 – schéma databáze

6.3.1. Popis schématu

Jak je vidět z E-R diagramu:

- Všichni uživatelé jsou v jedné tabulce, v které je jejich uživatelské jméno a heslo pro přihlášení do systému, dále jejich kompletní adresa a klíč k tabulce ROLE, kde je zaznamenána jejich role.
- Tabulka UZIVATELE nesplňuje 3. Normální formu. Sloupce PSC a MESTO jsou tranzitivně závislé na primárním klíči tabulky (UZIV_JMENO), s měnícím se PSC se mění i Město. Tyto sloupce jsou v jedné tabulce z praktických důvodů, abychom nemuseli udržovat tabulku s číselníkem měst a jejich PSČ.
- Jeden uživatel může mít pouze jednu roli (správce soutěží, zástupce klubu, administrátor(přidává a odebírá správce soutěží)).
- Každá soutěž patří do nějaké kategorie
- Kategorie mají stromovou strukturu
- Každá Kategorie, Klub, Rozhodčí má svého vlastníka a pouze on je může spravovat

V databázi je uložena kompletní historie zápasů Vyřazení klubu z vede ke smazání jeho kompletní historie. Smazání kategorie vede ke smazání všech podkategorií a všech informací s těmito kategoriemi spjatými.

6.4. Použitá syntaxe pro přístup k databázi

Ukázka použitého kódu pro přístup do databáze:

```
function spojeni() {

$conn = mysql_connect( "mysql.ic.cz", "ic_bobika_web", "nofear" )
or die ( "Zpojení selhalo: ". mysql_error() );

//nastavení znakové sady
mysql_query("SET CHARACTER SET utf8");
@mysql_query("set character_set_client=utf-8");
```

```
@mysql_query("set character_set_connection=utf-8");
@mysql_query("set character_set_results=utf-8");

if (!$conn) {
   die('Could not connect: ' . mysql_error());
}
mysql_select_db( "ic_bobika_web", $conn )
or die ( mysql_error() );
return $conn;
};
```

6.5. Popis stěžejního kódu php

```
<?php
session_start();
include('./secure/spojeni.php');
//odhlaseni
if (isset($_GET['odhlasit'])) {
  unset($_SESSION['login']);
  unset($_SESSION['user']);
  unset($_SESSION['role']);
}
?>
<?php include("header.php");
?>
<div class="menu_pravy">
        <div class="login">
                <?php include("login.php"); ?>
 </div>
                <?php if (isset($_SESSION['role'])) {
                        }?>
</div>
<div class="menu_levy">
<?php
        echo '<div class="menu_levy_nadpis">';
                echo ' Kategorie';
        echo '</div>';
        echo '<div class="menu_levy_kategorie">';
                if (!isset($_SESSION['role']))
                {
                        include("./menu/2.php");  //neprihlaseny uzivatel
                } else {
                        include('./menu/'.$_SESSION['role'].'.php');
                }
        echo '</div>';
?>
</div>
<div class="telo">
<?php
        if (isset($_GET['page'])) {
                if (isset($_SESSION['role'])) {
                include ('./include/'.$_SESSION['role'].'/'.$_GET['page'].'.php');
```

```
                } else {
                        include ('./include/2/'.$_GET['page'].'.php');
                }
        } else {
                if (isset($_GET['staticpage'])) {
                        include ('./static/'.$_GET['staticpage'].'.php');
                } else {
                        //include naky uvodni stranky
                        if (isset($_SESSION['role'])) {
                                include
('./include/'.$_SESSION['role'].'/index'.$_SESSION['role'].'.php');
                        } else {
                                include ('./include/2/index2.php');
                        }
                }
        }
?>
</div>
<?php include("footer.php"); ?>
```

Jak je vidět, vždy se načítá stránka *index.php*, která na základě parametru page, případně staticpage a role includuje vždy jiný obsah a jiná menu.

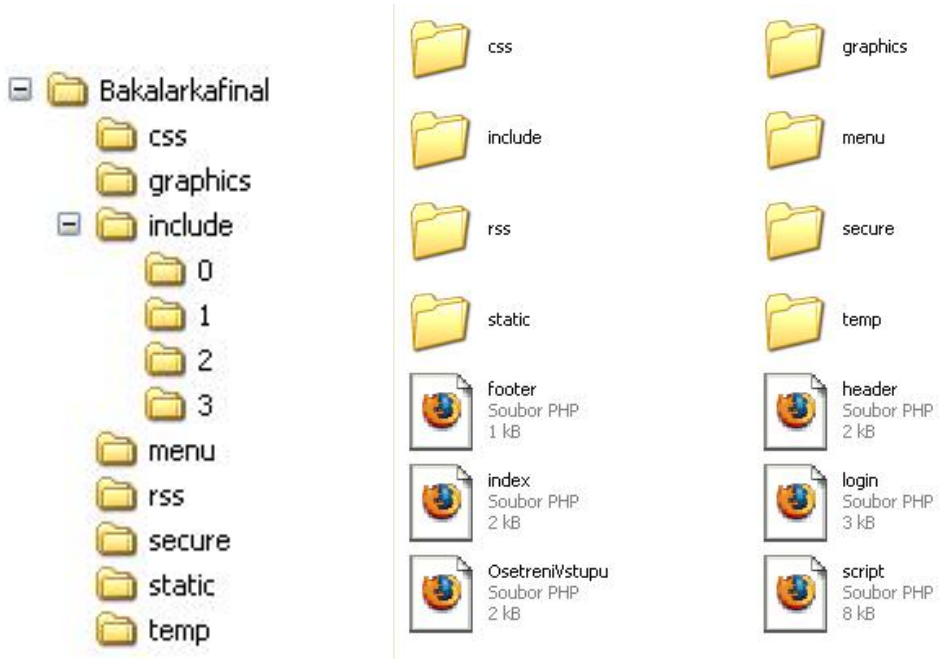
Do pravého menu je includována stránka *login.php* určená pro autorizovaný přístup, opět ji schématicky uvedu:

```
if (isset($_POST['user']) and isset($_POST['pass']) and !isset($_SESSION['login']))
{
        $conn = spojeni();
        $heslo=md5($_POST['pass']);
        $s = mysql_query("SELECT * FROM `UZIVATELE` WHERE `HESLO` =
        '" .$heslo. "' and `UZIV_JMENO` = '" . $_POST['user'] . "' ");
        if ($s === false)
        {
                echo ('Dotaz do databaze se nezdaril</br>');
                echo mysql_error($s);
        }
                $poct = mysql_num_rows( $s );
                if ( $poct == 1 ){
                        while ($r = mysql_fetch_array($s)) {
                                $_SESSION['login']='ano';
                                $_SESSION['user']=$r['UZIV_JMENO'];
                                $_SESSION['role']=$r['ID_ROLE'];
                        }
                }else {
                        $hlaska='<b>Chyba při přihlašování.<br/>Neplatné uživatel-
                        ské jméno nebo heslo.</b>';
         }
}mysql_close($conn);
<form>...</form>
```

7.Adresářová struktura a přístupová práva

7.1. Adresářová struktura

Z hlediska přehlednosti a bezpečnosti je nutné navrhnout správnou adresářovou strukturu systému a předem si ujasnit, kde budou která data a skripty uloženy. Výpis adresářové struktury vypadá následovně:



Obrázek 02 a 03 – adresářová struktura

- / kořenový adresář
- */css/* obsahuje použité kaskádové styly
- */include/* je klíčovým adresářem jádra a obsahuje všechny sdílené skripty a funkce. Veškerý obsah je chráněn proti samostatnému spuštění.
- */menu/* stránky menu, dle uživatelských rolí.
- */graphics/* slouží jako úložiště dat, která není možné uložit do databáze.
- */secure/* nastavení přístupu do databáze.
- */static/* obsahuje statické HTML stránky.
- */rss/* tvorba RSS kanálu.
- */temp/* dočasné generované soubory

7.2. Popis adresářové struktury a přístupových práv

Pojďme si nyní podrobně rozepsat, k čemu který adresář slouží.

„/“ je kořenový adresář, který obsahuje pouze základní soubory *index.php, header.php, footer.php, login.php, OsetreniVstupu.php*. Princip jejich činnosti si podrobněji popíšeme později.

V **root**u je již popsaná úvodní stránka *index.php a login.php*, dále hlavička a pata stránky *index.php* a *Osetrenivstupu.php*, kde jsou funkce pro ošetření vstupních formulářových polí.

V adresáři **css** jsou definice použitých kaskádových stylů.

V adresáři **include/číslo** jsou includované stránky pro jednotlivé role uživatelů, tedy 0-správce soutěží, 1-zástupce klubu, 2-návštěvník, 3-administrátor.

V adresáři **menu** jsou stránky levého menu *0.php, 1.php, 2.php, 3.php* pro includování menu podle rolí.

V adresáři **graphics** jsou obrázky ve formátu jpg (např. logo použité u RSS kanálu).

V adresáři **secure** je soubor *spojeni.php*, kde je funkce pro spojení s databází, je zde uloženo i heslo pro přístup do databáze a soubor *mailtest.php* obsahující funkci pro odesílání emailů Tato složka by neměla být zobrazitelná a soubory v ní čitelné, povoleno by mělo být pouze spouštění webovým serverem.

V adresáři **static** jsou statické HTML stránky jako „Manuál“ nebo „Autor“.

V adresáři **temp** jsou dočasné soubory generované stránkami.

V adresáři **rss** jsou soubory pro generování RSS kanálů.

Obecně by měli být přístupová práva nastavena tak aby si uživatel nemohl prohlédnout kódy *php* stránek, zvlášť v adresáři *secure*, čímž se zvyšuje šance na hack systému. Pouze adresáře **css**, **static** by měli být přístupné pro čtení.

7.3. Databázová abstrakční vrstva

Pro vykonávání SQL příkazů jsou v tomto případě použity funkce, které nám nabízí PHP. Jedná se zejména o:

- *mysql_connect()* - inicializace připojení k databázi
- *mysql_select_db()* - výběr konkrétní databáze
- *mysql_query()* - vykonání SQL příkazu / dotazu
- *mysql_fetch_array()* - získání jednoho řádku z vrácené tabulky dat
- *mysql_num_row()* - počet vrácených řádků tabulky

Z pohledu PHP se jedná o základní funkce pro práci s databází MySQL, takže se jedná o velice jednoduchou abstrakční vrstvu, která neumí pracovat s jinými databázemi ani neumí sama provádět složitější operace s výsledky dotazů.

7.4. Spojení s databázovým systémem

7.4.1. Možnosti přístupu k datovým zdrojům

Nativní funkce. Všechny DBMS jsou vybaveny nativními funkcemi, které zajišťují základní komunikační procesy – jedná se o komunikaci na nejnižší úrovni. Tyto funkce jsou programátory koncových aplikací málo využívány, není-li kladen na aplikaci požadavek extrémní rychlosti. Tyto funkce jsou tedy jakousi mezivrstvou, kterou používají zapouzdřené funkce.

Zapouzdřené funkce zachovávají strukturu daného programovacího jazyka a na pozadí volají funkce nativní. Programátor potom pracuje s funkcí či komponentou podle svých zvyklostí a již se nezajímá o komunikaci s databází. PHP podporuje velké množství databází. Problém je, že funkce pro práci s jednotlivými databázemi jsou pojmenovány různě, takže ztrácíme obecnost. Řešením je použití některého obecného propojovacího protokolu nebo abstrakční vrstvy.

Obecný propojovací protokol ODBC. Tento protokol vznikl kvůli potřebě jednoduchého přechodu z jedné databáze na jinou. ODBC obsahuje standardizované API, pomocí kterého je možné přistupovat k databázím shodným způsobem. Velkou nevýhodou je nízká výkonnost oproti standardním zapouzdřeným funkcím.

Abstrakční vrstva. Jde o sadu funkcí pro přístup k databázím, které podle typu používají příslušné zapouzdřené funkce. Jedná se o zobecnění zapouzdřených funkcí. Nevýhodou je možnost volání pouze těch zobecněných funkcí, které podporují všechny databáze a omezení společného SQL příkazu na společný standard.

7.4.2. MD5 (Message-Digest Algorithm)

Je to funkce, která vytváří ze vstupního (libovolně velkého) množství dat výstupní množství dat fixní délky (otisk). Tento otisk má velikost 128 bitů. Někdy je otisk označován jako hash nebo miniatura. Jeho zásadní vlastností je, že malá změna na vstupu provede velkou změnu na výstupu (zcela odlišné otisky). Algoritmus MD5 je popsán standardem RFC1321 [23].

Byl vytvořen Ronaldem Rivestem v roce 1991 a nahradil stávající MD4. V roce 1996 se objevila vada v návrhu a i přes to, že nebyla zásadní, kryptologové doporučili používat jiné algoritmy. V roce 2004 jej kryptologové z Japonska (později z ČR) označili jako nebezpečný, neboť objevili zpětný algoritmus, kterým je možno vysledovat přibližnou množinu vstupních znaků.

8. Webové rozhraní databáze

8.1. Klíčové vlastnosti

Při návrhu prvního z obou rozhraní bylo na prvním místě vytvořit stabilní a bezpečný systém pro správu dat v databázi. Důraz byl kladen na datovou nenáročnost a jednoduché ovládání. Stylování pomocí CSS. Každá stránka musí obsahovat identifikaci právě přihlášeného uživatele a jeho práva, možnost odhlásit se ze systému vždy přístupné menu barevně odlišené odkazy pro ulehčení orientace.

8.2. Bezpečnost systému

Co se týče zabezpečení, je počítáno s ochranou před běžnými útoky typu *SQL injection* [24] a *session hijacking*. Na minimum bude omezeno množství informací uložených v počítači. Pro tento účel budou využity relace (*sessions*).

8.3. Zápasy

Pro aktuální kategorii dojde k zobrazení všech zápasů (číslo utkání, soupeři, datum a čas, výsledek). Domácí tým má možnost měnit čas a místo utkání a vkládat výsledky utkání. Hostující tým vkládá výsledek utkání. Správce soutěže přidává samotné zápasy, přiděluje k nim rozhodčí a nastavuje datum utkání.

8.4. Uživatelé

Obsahují jednotlivé kluby, správce soutěží a administrátora. Nalezneme zde osobní informace s možností jejich úpravy. Kluby mají odkaz na svůj profil. Jako dostatečné množství informací pro přidání uživatele stanovuji: jméno, příjmení, název klubu ,uživatelské jméno, ulice, číslo popisné, PSČ, telefon ,heslo a email. Povinné údaje se liší dle předpokládané role uživatele. Formáty veškerých vkládaných údajů se ověřují.

Kromě tabulky s uživateli je nutné vytvořit stránku pro úpravu profilu.Uživatel si je může upravit svépomocí po přihlášení:

- Změna uživatelských údajů

8.5. Kluby

Stránka bude obsahovat seznam všech uložených týmů pro daného správce soutěže. Než bude tým možné přidat do rozpisu soutěží, musí být nejprve vytvořen. Pro již přidané týmy opět platí, že je lze odebrat, ale následuje smazaní všech dat souvisejících s týmem. Jednoznačná identifikace týmu je zajištěna pomocí uživatelského jména.

8.6. Správce soutěží

Je správcem svého sportu. Vkládá všechny data umožňující konaní jeho sportu : kluby, rozhodčí, soutěže, termínovou listinu.

8.7. Soutěže

Jednotlivé kluby jsou vkládány do kategorií. Klub může hrát více kategorii. Kategorie mají charakter stromu. Pokud smažeme nějakou kategorii, smažou se i všechny podkategorie včetně souvisejících dat.

8.8. Rozhodčí

Obsahují jednotlivé rozhodčí. Nalezneme zde osobní informace. Rozhodčí mají odkaz na svůj profil. Jako dostatečné množství informací pro přidání uživatele stanovuji: jméno, příjmení, ulice, číslo popisné, PSČ, telefon a email. Formáty veškerých vkládaných údajů se ověřují.

8.9. Rozpisy

Po přidání všech soupeřů v aktuální kategorii vygenerujeme automaticky rozpis utkání. Základní systém je každý z každým doma a venku. Lze ale přidat i jednotlivá utkání. Kluby poté vloží čas a místo konaní utkání. Ke každému zápasu je přidělen správcem soutěže jeden nebo více rozhodčích.

9.Závěr

Podařilo se mi vytvořit fungující webový informační systém, který splňuje zadání. Provedl jsem analýzu požadavků na systém s ohledem na specifika sportovních utkání. Systém byl vytvořen na základě obecných požadavků sportovních utkání. Myslím, že splňuje jejich požadavky na funkčnost, bezpečnost a jednoduchost ovládání. Je navržen s maximální obecností a bylo by možné jej s minimálními úpravami, využívat pro libovolný sport. Pro uvedení do praxe je potřeba ještě systém otestovat v plném provozu. Domnívám se že by poté bylo možné tento systém plně využívat.

Seznam literatury

[1] SAK, Petr. SAKOVÁ, Karolína. Jak Internet mění společnost [online]. 2007, [cit. 2009-01-10]. Dostupné na WWW: < http://www.lupa.cz/clanky/zpusob-vyuzivani-osobniho-pocitace-a-internetu >.

[2] VANĚK, Jiří. Statické stránky vs. dynamické. Jaký je rozdíl? [online]. Publikováno listopad 2008. [cit. 2009-04-10]. Dostupné na WWW: < http://jirivanek.eu/staticke-stranky-vs-dynamicke-jaky-je-rozdil >.

[3] WYSIWYG - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-04.08]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/WYSIWYG >.

[4] Redakční systémy [online]. [cit. 2009-05.10] Dostupný z: < http://www.artic-studio.net/webove-stranky/redakcni-system/ >.

[5] HTML - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-04-05]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/HTML >.

[6] SALVET, Pavel. Kam kráčí HTML, publikováno 2007-09-07. [cit. 2009-04-12]. Dostupné na WWW: < http://www.lupa.cz/clanky/kam-kraci-html/ >.

[7] PHP - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-04.08]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/php >.

[8] ZAJÍC, P. Historie a Budoucnost v PHP [online]. Poslední aktualizace 2004-05-27. [cit. 2009-04.06]. Dostupné na WWW: < http://www.linuxsoft.cz/article.php?id_article=171 >.

[9] SAMEK, Michal. PHP:Knihovna Funkcí[online]. Publikováno 1999-06-028. [cit. 2009-04-07]. Dostupné na WWW: < http://www.root.cz/clanky/php-knihovna-funkci/ >.

[10] BLAHOUT, Michal. Jemný úvod do ASP [online]. Publikováno 2002. [cit. 2009-04.17]. Dostupné na WWW: < http://www12.brinkster.com/mibla >.

[11] DOČEKAL, Daniel. Co všechno byste chtěli vědět o ASP+ a přitom nevěděli kde hledat. Publikováno 2002-10-25. [cit. 2009-04.05]. .Dostupné na WWW: < http://www.pooh.cz/pooh/a.asp?a=2002977&db=1001 >.

[12] MySQL - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-03-06]. Dostupné na WWW: < http://encyklopedie.seznam.cz/heslo/482131-mysql >.

[13] XML - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-04.19]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/RSS >.

[14] BOUŠKA, Petr. Jak vytvořit RSS kanál v PHP [online], publikováno 2006-04-04. [cit. 2009-03.08].. Dostupné na WWW: < http://www.samuraj-cz.com/clanek/jak-vytvorit-rss-kanal-v-php/ >.

[15] Atom (standard) - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-05-03]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/Atom_(standard) >.

[16] MACICH, Jiří. Přehled RSS čteček [online]. Publikováno 2005-06-06. [cit. 2009-04-20]. Dostupné na WWW: < http://www.lupa.cz/clanky/prehled-rss-ctecek/ >.

[17] Služba webhostingu - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-04-19]. Dostupné na WWW: < http://wikipedia.infostar.cz/w/we/web_hosting.html >.

[18] Rozesílání emailů. Publikováno 10.4.2004. [cit. 2009-03-01]. Dostupné na WWW: < http://www.owebu.cz/php/vypis.php?clanek=379 >.

[19] KOCMAN, Jiří. Jak na démona Cron [online], publikováno 2002-04-21. [cit. 2009-04-11]. Dostupné na WWW: < http://interval.cz/clanky/jak-na-demona-cron/ >.

[20] ASPone s.r.o. - Plánovač úloh [online]. [cit. 2009-04.17] Dostupné na WWW: < http://www.aspone.cz/80500/Applications-Scheduled-Tasks.aspx >.

[21] VALADE, Janet. PHP and MySQL For Dummies, 2nd Edition, John Wiley & Sons. April 2004. ISBN 978-0-7645-5589-3

[22] SKŘIVAN, Jaromír. Databáze a jazyk SQL [online] 2000. [cit. 2009-04-18]. Dostupný z: < http://interval.cz/clanky/databaze-a-jazyk-sql/ >.

[23] MD5 - Wikipedie, otevřená encyklopedie [online]. [cit. 2009-05-03]. Dostupné na WWW: < http://cs.wikipedia.org/wiki/MD5 >.

[24] ANLEY, Chris. Advanced SQL Injection In SQL Server Applications. NGSSoftware Insight Security Research. 2002 http://www.ngssoftware.com/papers/

Seznam použitých zkratek

ASP - Active Server Pages

Atom - Atom Syndication Format

CGI - Common Gateway Interface

CLR - Common Language Runtime

DB - Database

DBMS - Database Management Systém

DLL - Dynamically Linked Library

DOCTYPE – Dokument Type Definition

HTML - Hypertext Markup Language

HTTP - Hypertext Transfer Protocol

IS - Information System

MD5 - Message-Digest Algorithm

MyISAM - Once of Storage Engine

MySQL – Multiplatform Database

ODBC - Open Database Connectivity

OS - Operační Systém

PHP - Personal Home Page

RSS - Really Simple Syndication

SQL - Structured Query Language

SGML - Standard Generalized Markup Language

URL - Uniform Resource Locator

WYSIWYG - What you see is what you get

WWW - World Wide Web

XHTML - Extensible Hypertext Markup Language

XML - eXtensible Markup Language

Seznam obrázků

Seznam tabulek

Seznam příloh

Příloha 1

Příloha 2

Obsah přiloženého CD

Na přiloženém CD se nachází zdrojové kódy realizovaného redakčního systému, data pro naplnění databáze a text této práce ve formátu PDF:

<dir> /doc – text bakalářské práce a uživatelský manuál

<dir> /data – zdrojové kódy aplikace

<dir> /databaze – defaultní data pro naplnění databáze