

Turnajový systém

Tournament system

Bc. Jan Bula

Diplomová práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan BULA**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**

Téma práce: **Turnajový systém**

Zásady pro vypracování:

1. Vytvořte literární rešerši na zadané téma.
2. Definujte analogii a účelnost použitých technologií.
3. Popište nejnovější verze jazyků PHP, XHTML a CSS a jejich budoucí využití.
4. Navrhněte a popište relační model databáze.
5. Vytvořte turnajový systém a otestujte kompatibilitu na různých platformách a internetových prohlížečích.
6. Porovnejte systém s již vytvořenými systémy.
7. Součástí práce budou provedeny a vyhodnoceny testy funkčnosti včetně diskuze.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Michael Kofler, Bernd Öggl, PHP 5 a MySQL 5 – Průvodce webového programátora, Computer Press, říjen 2007, ISBN: 978-80-251-1813-9
2. Elizabeth Castro, HTML, XHTML a CSS – Názorný průvodce tvorbou WWW stránek, Computer Press, duben 2007, ISBN: 978-80-251-1531-2
3. Luboslav Lacko, Ajax – Hotová řešení, Computer Press, květen 2008, ISBN: 978-80-251-2108-5
4. <http://seo.jasminka.cz/> – Optimalizace pro vyhledávače -- SEO
5. <http://www.php.net/> – domovská stránka PHP

Vedoucí diplomové práce:

Ing. Tomáš Podoba

Ústav aplikované informatiky

Datum zadání diplomové práce:

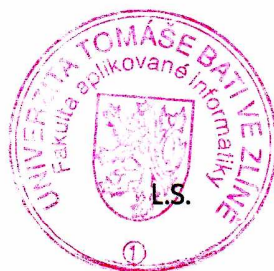
20. února 2009

Termín odevzdání diplomové práce:

27. května 2009

Ve Zlíně dne 13. února 2009


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je vytvořit turnajový systém určený primárně pro počítačové hry, ale nejen pro ně. Systém je naprogramován v jazyce PHP s využitím databáze MySQL a výstupem do XHTML formátovaného pomocí CSS.

Teoretická část se zabývá charakteristikou použitých prostředků jako je PHP, HTML, CSS a další. U každé je uveden popis, historie, vývoj a přehled jednotlivých verzí.

Náplní praktické části je popis vytvořeného systému. První část je zaměřena na ovládání a funkcionalitu, následuje rozbor jednotlivých souborů a funkcí, přičemž důraz je kladen na ty, které tvoří jádro systému. Samostatná kapitola je věnována návrhu a popisu databázového modelu. V závěru je probráno zabezpečení systému, kompatibilita a nakonec srovnání s konkurencí.

Klíčová slova: turnajový systém, turnaj, liga, playoff, WWW, PHP, MySQL, XHTML, CSS, SEO

ABSTRACT

The aim of this thesis is to create a tournament system designed primarily for computer games, but not limited to them. The system is programmed in PHP using MySQL database and output to XHTML formatted using CSS.

The theoretical part deals with the characteristics of the devices such as PHP, HTML, CSS and others. For each, there is a description, history, development and overview of each version.

The main of the practical part is a description of the system created. The first part is focused on control and functionality, followed by analysis of files and functions, with an emphasis on those which form the core of the system. A separate chapter is devoted to design and description of the database model. In the conclusion is discussed security of the system, compatibility, and finally comparison with competitors.

Keywords: tournament system, tournament, league, playoff, WWW, PHP, MySQL, XHTML, CSS, SEO

Děkuji Ing. Tomáši Podobovi, vedoucímu diplomové práce, a Ing. Jiřímu Gieslovi, konzultantovi diplomové práce, za odborné vedení, cenné rady a připomínky, které mi poskytovali při řešení a realizaci této diplomové práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.
V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 WWW	11
1.1 WORLD WIDE WEB CONSORTIUM.....	11
2 HTTP	12
2.1 ZABEZPEČENÉ HTTP (HTTPS).....	14
3 PHP	15
3.1 VERZE PHP	15
4 APACHE	19
5 MYSQL	20
5.1 HISTORIE MYSQL.....	21
6 HTML / XHTML	22
6.1 HTML.....	22
6.1.1 Verze jazyka HTML.....	22
6.1.2 Koncepce jazyka HTML	23
6.2 XHTML.....	25
6.2.1 Verze jazyka XHTML.....	25
6.2.2 Rozdíly mezi XHTML a HTML	26
7 CSS	28
7.1 VERZE CSS	28
7.2 SYNTAXE.....	29
7.2.1 Selektor	29
7.2.2 Třída	29
7.2.3 Identifikátor	30
8 SEO	31
8.1 POUŽÍVANÉ METODY SEO.....	32
8.2 ZAKÁZANÉ METODY SEO.....	33
II PRAKTICKÁ ČÁST	35
9 STRUKTURA TURNAJOVÉHO SYSTÉMU	36
9.1 OVLÁDÁNÍ A POPIS FUNKCIONALITY SYSTÉMU	36
9.1.1 Registrace uživatele.....	37
9.1.2 Přihlášení.....	37
9.1.3 Registrace týmu	38
9.1.4 Založení turnaje.....	38
9.1.5 Administrace turnaje	38
9.1.6 Registrace do turnaje.....	39
9.1.7 Účast v turnaji z pohledu hráče	39

9.1.8	Členství v týmu	39
9.1.9	Přepínání týmu	40
9.1.10	Administrace týmu	40
9.1.11	Administrace uživatele.....	40
9.1.12	Možnosti supervizora.....	40
9.1.13	Možnosti administrátora.....	40
9.2	STRUKTURA SOUBORŮ SYSTÉMU	40
9.3	SOUBOR FUNCTIONS.PHP	43
10	DATABÁZE.....	56
11	ZABEZPEČENÍ	65
11.1	UŽIVATELSKÁ PRÁVA	65
11.2	OCHRANA FORMULÁŘŮ PROTI PODVRŽENÍ DAT	65
11.3	OŠETŘENÍ UŽIVATELSKÝCH VSTUPŮ Z FORMULÁŘŮ.....	65
11.4	OCHRANA PROTI PHP INJECTION.....	66
11.5	OCHRANA PROTI SESSION STEAL	66
11.6	OCHRANA PHP SKRIPTŮ POMOCÍ .HTACCESS.....	67
11.7	OBRANA PROTI NEŽÁDOUCÍMU VÍCENÁSOBNÉMU ODESLÁNÍ FORMULÁŘE	67
12	KOMPATIBILITA	68
12.1	KOMPATIBILITA V RŮZNÝCH PROHLÍZEČÍCH.....	68
12.2	KOMPATIBILITA NA STRANĚ SERVERU	68
13	POROVNÁNÍ S KONKURENČNÍMI SYSTÉMY.....	70
13.1	ELECTRONIC SPORTS LEAGUE	70
13.2	CLANBASE A GLOBAL GAMING LEAGUE.....	71
13.3	GAMEPARK	72
13.4	SROVNÁNÍ S NAŠÍM SYSTÉMEM	73
13.4.1	Návrhy rozšíření	73
	ZÁVĚR	74
	CONCLUSION	75
	SEZNAM POUŽITÉ LITERATURY.....	76
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	80
	SEZNAM OBRÁZKŮ	81
	SEZNAM TABULEK.....	82

ÚVOD

Jednou z věcí, která v posledních dvaceti letech změnila náš život nejvíce, je internet a především jeho služba WWW. Od svého zrodu na konci osmdesátých let minulého století, kdy sloužila pro výměnu a prezentaci informací mezi vědci a byla čistě statická, se postupem času vyvinula až do dnešní podoby dynamických interaktivních webových stránek. Současné webové technologie tak umožňují vytvářet téměř plnohodnotné aplikace typu klient–server. Největší výhodou tohoto řešení je fakt, že na straně klienta není potřeba žádný speciální software, přístup probíhá přes webový prohlížeč, který má v počítači nainstalovaný každý uživatel. Daná aplikace je tím pádem jednoduše přístupná milionům uživatelů po celém světě. Hlavní nedostatek zatím spočívá v tom, že webové technologie ještě nedokáží nabídnout a realizovat to, co speciální klientský program, případně je naprogramování takové aplikace velmi komplikované a složité. Důležitým aspektem je také zabezpečení aplikace, především jedná-li se o systém pracující s citlivými daty jako jsou elektronické obchody, bankovníctví apod. Způsobů, jak napadnout webovou aplikaci, je mnohem více, než v případě klientského programu. Útočník má totiž možnost nahlédnout do zdrojového kódu stránky a vyčíst tak spoustu potenciálně důležitých informací o fungování celé aplikace. Velkou část úsilí při tvorbě webové aplikace je tak třeba věnovat eliminaci možných rizik a zabránit tak případným útokům ještě dříve, než se naskytnou.

Jedním z mnoha druhů webových aplikací jsou turnajové systémy. Drtivá většina z nich je však učena pro multiplayerové počítačové, případně konzolové hry. Hráči si navíc mohou vybrat jen z turnajů, které vypíše správci, a většinou nemají možnost založit si vlastní turnaj podle svých potřeb. Proto vznikla myšlenka na turnajový systém, který by nebyl orientován pouze na počítačové hry, ale byl by využitelný i pro jiné sporty nebo soutěže a zároveň by dal uživateli možnost vytvořit si a spravovat vlastní turnaje dle přání. Vytvořit takový systém bylo cílem této diplomové práce.

I. TEORETICKÁ ČÁST

1 WWW

World Wide Web (WWW, také pouze zkráceně web), ve volném překladu „celosvětová pavučina“, je označení pro aplikace internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů. V češtině se slovo web často používá nejen pro označení celosvětové sítě dokumentů, ale také pro označení jednotlivé soustavy dokumentů dostupných na tomtéž webovém serveru nebo na téže internetové doméně nejnižšího stupně (internetové stránce). Dokumenty umístěné na počítačových serverech jsou adresovány pomocí URL, jehož součástí je i doména a jméno počítače. Název naprosté většiny těchto serverů začíná zkratkou www, i když je možné používat libovolné jméno vyhovující pravidlům URL. Protokol HTTP je dnes již používán i pro přenos jiných dokumentů, než jen souborů ve tvaru HTML a výraz World Wide Web se postupně stává pro laickou veřejnost synonymem pro internetové aplikace. Autorem Webu je Tim Berners-Lee, který jej vytvořil při svém působení v CERNu. Navrhl jazyk HTML a protokol HTTP, napsal první webový prohlížeč WorldWideWeb a koncem roku 1990 spustil první webový server na světě info.cern.ch. V říjnu roku 1994 založil World Wide Web Consortium (W3C), které dohlíží na další vývoj Webu. [5]

1.1 World Wide Web Consortium

World Wide Web Consortium (W3C) je mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy pro World Wide Web. Konsorcium založil Tim Berners-Lee v říjnu roku 1994. Před založením konsorcia nabízely různé firmy různé upravené verze jazyka HTML které byly nekompatibilní s verzemi od ostatních výrobců. Konsorcium sjednotilo verze od různých výrobců a dohodlo se s nimi na základních principech a komponentách nových standardů. Cílem konsorcia je „*Rozvíjet World Wide Web do jeho plného potenciálu vývojem protokolů a směrnic které zajistí dlouhodobý růst Webu*“. W3C se také zabývá vzděláním a přístupností, vyvíjí software a nabízí otevřenou diskuzi o Webu prostřednictvím fóra. Konsorciu předsedá jeho zakladatel Tim Berners-Lee. [6]

2 HTTP

HyperText Transfer Protocol (HTTP) je internetový protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v RFC 2616. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor (podobně jako e-mail), používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu.

Pokud uživatel bude mít po chvíli další dotaz na stejný server (např. proto, že uživatel v dokumentu kliknul na hypertextový odkaz), bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, jestli tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká bezstavový protokol – protokol neumí uchovávat stav komunikace, dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho „nákupního košíku“ apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

HTTP definuje několik metod, které se mají provést nad uvedeným objektem (dokumentem). <metoda> <objekt> HTTP/<verze>:

GET

Požadavek na uvedený objekt se zasláním případných dat (proměnné prohlížeče, session id, atd.). Výchozí metoda při požadavku na zobrazení hypertextových stránek, RSS feedů aj. Celkově nejpoužívanější.

HEAD

To samé jako metoda GET, ale už nepředává data. Poskytne pouze metadata o požadovaném cíli (velikost, typ, datum změny, ...).

POST

Odesílá uživatelská data na server. Používá se například při odesílání formuláře na webu. S předaným objektem se pak zachází podobně jako při metodě GET. Data může odesílat i metoda GET, metoda POST se ale používá pro příliš velká data (více než 512 bajtů, což je velikost požadavku GET) nebo pokud není vhodné přenášena data zobrazit jako součást URL (data předávaná metodou POST jsou obsažena v HTTP požadavku).

PUT

Nahrává data na server. Objekt je jméno vytvářeného souboru. Používá se velmi zřídka, pro nahrávání dat na server se běžně používá FTP nebo SCP/SSH.

DELETE

Smaže uvedený objekt ze serveru. Jsou na to potřeba jistá oprávnění, jako u metody PUT.

TRACE

Odešle kopii obdrženého požadavku zpět odesílateli, takže klient může zjistit, co na požadavku mění nebo přidávají servery, kterými požadavek prochází.

OPTIONS

Dotaz na server, jaké podporuje metody.

CONNECT

Spojí se s uvedeným objektem před uvedený port. Používá se při průchodu skrze proxy pro ustanovení kanálu SSL. [7]

2.1 Zabezpečené HTTP (HTTPS)

HTTPS je v informatice nadstavba síťového protokolu HTTP, která umožňuje zabezpečit spojení mezi webovým prohlížečem a webovým serverem před odposloucháváním, podvržením dat a umožňuje též ověřit identitu protistrany. HTTPS používá protokol HTTP, přičemž přenášená data jsou šifrována pomocí SSL nebo TLS a standardní port na straně serveru je 443.

Protokol HTTPS využívá asymetrické šifrování. Obě strany si před zahájením komunikace vygenerují pár klíčů (privátní a veřejný). Při zahájení komunikace si vymění veřejné klíče, které by obě strany měly ověřit pomocí jiného komunikačního kanálu. Ověření může proběhnout kontrolou výtahu (otisk, miniatura, hash) veřejného klíče u protistrany například pomocí telefonu nebo lze použít princip přenosu důvěry, kdy nám protistrana předá veřejný klíč, který je digitálně podepsaný (nejlépe certifikační autoritou, které důvěřujeme a jejíž veřejný klíč máme v důvěryhodném úložišti).

Zatímco samotné šifrování ochrání komunikaci před odposloucháváním, bez ověření autenticity veřejných klíčů jsou komunikující strany vystaveny riziku útoku Man in the middle.

Za certifikáty vydané certifikačními autoritami, které mají svůj veřejný klíč v úložišti, které je dodáváno s webovým prohlížečem, je nutné platit. Existuje však možnost vytvoření certifikátu, který si vydavatel sám sobě podepíše, avšak v takovém případě musí protistrana přidat do úložiště veřejný klíč sama (a ověřit ho jinak). [8]

3 PHP

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací. PHP lze použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou většinou prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti (interpret PHP skriptu je možné volat pomocí příkazové řádky). Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...).

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl. [9]

3.1 Verze PHP

PHP/FI

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval Personal Home Page Tools. Protože byla potřeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro web. Rasmus se rozhodl uvolnit zdrojový kód PHP/FI pro všechny, takže kdokoli ho může používat, stejně jako opravovat chyby a vylepšovat kód.

PHP/FI, což znamená Personal Home Page/Forms Interpreter, obsahovalo něco ze základní funkcionality PHP, jak ho známe dnes. Mělo proměnné perlovského typu, automatickou interpretaci formulářových proměnných a syntaxi vloženou do HTML. Syntaxe samotná byla podobná jazyku Perl, přestože mnohem omezenější, jednodušší a v něčem nekonzistentní.

PHP/FI 2.0

V roce 1997 se PHP/FI 2.0, druhá implementace psaná v C, stala kultovní záležitostí pro (odhadem) tisíce uživatelů po celém světě a přibližně 50 000 domén oznamujících nainstalované PHP/FI, což čítalo zhruba 1 % všech domén na Internetu. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

PHP/FI 2.0 bylo oficiálně uvolněno až v listopadu 1997, poté, co strávilo většinu svého života v betaverzích. Krátce nato bylo následováno první alfa verzí PHP 3.0.

PHP 3

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný celek poté, co shledali PHP/FI 2.0 výrazně "poddimenzovaným" pro vývoj svých aplikací pro e-komerci. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI, rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0 a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. Kromě pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly především možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.

Nový jazyk byl uvolněn pod novým názvem, který odstranil implikaci omezeného osobního použití, kterou neslo označení PHP/FI 2.0. Byl nazván pouze PHP, což je rekurzivní akronym - PHP: Hypertext Preprocessor.

Na konci roku 1998 vyrostlo PHP do rozsahu instalací v řádu (odhadem) desítek tisíc uživatelů a stovek tisíc webů. V době svého vrcholu bylo PHP 3.0 instalováno na přibližně 10 % všech WWW serverů na Internetu.

PHP 3.0 bylo oficiálně uvolněno v červnu 1998 poté, co strávilo cca 9 měsíců ve veřejném testování.

PHP 4

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přepisu jádra PHP. Cílem návrhu bylo zlepšit modularitu kódové báze PHP a zvýšit výkon pro složité aplikace. Tyto aplikace byly schopny pracovat s PHP 3.0 (díky novým možnostem a podpoře široké škály databází a API od jiných tvůrců), ale PHP 3.0 nebylo navrženo pro efektivní práci tak náročných aplikací.

Nový engine, nazvaný Zend Engine (název byl sestaven z jejich křestních jmen - Zeev a Andi), úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho různých WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

PHP 5

Na vývoji PHP 5 se začalo pracovat již v roce 2002. Základem této verze je zcela přepracovaný Zend Engine 2, který jednak přinesl vyšší výkon kritických PHP aplikací, jednak umožnil zakomponovat do PHP řadu pokročilejších programovacích struktur, čímž především otevřel cestu kvalitnější podpoře objektově orientovaného programování v PHP.

První betaverze PHP 5 byly veřejnosti k dispozici na jaře roku 2003 a první oficiální verze PHP 5 byla uvolněna 13. července 2004.

PHP 5 je do značné míry zpětně kompatibilní s PHP 4. V nové verzi jazyka byly především posíleny bezpečnostní mechanismy (což může vést k nefunkčnosti některých špatně postavených aplikací pro PHP 4) a uveden nový, podstatně kvalitnější objektový model, umožňující používat PHP jako skutečný objektově orientovaný jazyk. [10]

PHP 6

O nové verzi PHP se mluví již od roku 2005, nicméně do dnešního dne ještě nebylo oficiálně vydáno. Mezi hlavní změny patří přechod na plně objektový model, který poběží na novém Zend Framework 3, samozřejmě bude zachována i zpětná kompatibilita. Odstraněny budou některé direktivy, jako například `register_globals`, `magic_quotes_gpc`, `safe_mode` a další. Bude tedy nutné upravit skripty, které zmíněné direktivy používají. Naopak přibýt má nová direktiva `allow_url_include`, která bude ovlivňovat používání vzdálených souborů při vkládání PHP kódu. Další novou vlastností je plná podpora Unicode, spíše drobnou záležitostí je vznik datového typu `int64`.

Do finální verze se však mohou některé věci změnit, takže na konečný výsledek si musíme ještě počkat.

4 APACHE

Apache HTTP Server je multiplatformní softwarový webový server s otevřeným kódem. V současné době dodává prohlížečům na celém světě většinu internetových stránek.

Vývoj Apache začal v roce 1993 v NCSA (National Center for Supercomputing Applications) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. V dalším roce však vývojářský tým opustil hlavní programátor Rob McCool, tím došlo ke zpomalení vývoje a poté, v roce 1998, k úplnému zastavení.

NCSA HTTPd však mezitím už používali správci webových serverů a dodávali k němu vlastní úpravy – patche (patch = záplata). Hlavní úlohu v dalším vývoji sehráli Brian Behlendorf a Cliff Skolnick, kteří založili emailovou konferenci a začali sběr úprav a jejich distribuci koordinovat. První veřejná verze s označením 0.6.2 byla vydána v dubnu 1995. Následovalo kompletní přepsání kódu (Apache2 už neobsahuje nic z původního NCSA HTTPd) a založení Apache Group, která je dnes základem vývojářského týmu.

Od dubna 1996 byl Apache nejpopulárnější server na internetu. V květnu 1999 běžel na 57% všech serverů a v listopadu 2005 jeho používanost dosáhla 69% (výsledky měření Netcraft).

Název vznikl z úcty a obdivu k domorodému kmenu nativních Američanů - Apačů anebo anglického slovního spojení „A patchy server“ (patchovaný server, kdysi byl Apache pouze sada patchů pro jiný web server). Jako indiánský symbol je ve znaku ptačí pero. [11]

5 MYSQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet.

MySQL nabízí několik typů databázových tabulek (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů:

- MyISAM - nejpoužívanější, bez podpory transakcí
- InnoDB - podpora transakcí
- BerkeleyDB (BDB)
- MEMORY - práci s daty v paměti
- NDB Cluster - úložiště pro clusterované databáze (od verze 5.0)
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech [12]

5.1 Historie MySQL

- První vydání MySQL 23. dubna 1995
- První verze pro Windows 95 a NT - srpen 1998
- Verze 3.23: beta verze v červnu 2000, vydáno v lednu 2001
- Verze 4.0: beta verze v srpnu 2002, vydáno v březnu 2003 (podpora union)
- Verze 4.1: beta verze v červnu 2004, vydáno v říjnu 2004 (R-stromy, B-stromy, poddotazy)
- Verze 5.0: beta verze v březnu 2005, vydáno v říjnu 2005 (cursors, triggers(spouštěče), pohledy, XA transakce)
- 26 února 2008 Sun Microsystems získala MySQL AB
- Verze 5.1: vydáno v listopadu 2008 (kalendář událostí, partitioning(rozdělování), plugin API, row-based replication, server log tables) [12]

6 HTML / XHTML

6.1 HTML

HyperText Markup Language, označovaný zkratkou HTML, 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).

6.1.1 Verze jazyka HTML

HTML 0.9

Byla vydána v roce 1991. Nepodporuje grafický režim.

HTML 2.0

Zachycuje stav jazyka v polovině roku 1994. Standard vydala komunita IETF (Internet Engineering Task Force). Je to první verze, která odpovídá syntaxi SGML. Přidává k původní specifikaci interaktivní formuláře a podporu grafiky.

HTML 3.2

Byla vydána 14. ledna 1997 a zachycuje stav jazyka v roce 1996. Standard už vydalo W3C, stejně jako následující verze. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

HTML 4.0

Byla vydána 18. prosince 1997. Do specifikace jazyka přibyly nové prvky pro tvorbu tabulek, formulářů a nově byly standardizovány rámy (frames). Tato verze se snaží dosáhnout původního účelu – prvky by měly vyznačovat význam (sémantiku) jednotlivých částí dokumentu, vzhled má být ovlivňován připojovanými styly. Některé prezentační elementy byly zavrženy.

HTML 4.01

Byla vydána 24. prosince 1999. Tato verze opravuje některé chyby verze předchozí. Podle původního předpokladu se mělo jednat o poslední verzi, po které by se přešlo na XHTML.

HTML 5

7. března 2007 byla založena nová pracovní skupina HTML, jejíž cílem je vývoj nové verze HTML. V květnu 2007 bylo odhlasováno, že základem nové specifikace se stanou Web Applications 1.0 a Web Forms 2.0 ze specifikace WHATWG. Jako název nové specifikace bylo odhlasováno HTML 5. Specifikace by měla být hotova do roku 2012.

6.1.2 Koncepce jazyka HTML

Jazyk HTML je od verze 2.0 aplikací SGML (připravovaná verze HTML 5 ale již závislost na SGML obsahovat nebude). 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 otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. element (prvek) dokumentu. Součástí obsahu elementu mohou být další vnořené elementy. Tagy mohou obsahovat atributy, které popisují jejich vlastnosti nebo nesou jinou informaci.

Značky (zvané tagy) jsou obvykle párové, přičemž koncová značka je shodná se značkou počáteční, jen má před názvem znak lomítka. Některé značky jsou nepárové – nemají žádný obsah a nepoužívají koncovou značku.

Pro každou verzi existuje definice pravidel DTD (Document Type Definition). Od verze 4.01 musí být odkaz na deklaraci DTD v dokumentu uveden pomocí klíčového slova DOCTYPE. DTD definuje pro určitou verzi elementy a atributy, které lze používat.

Dokument může mimo značkování obsahovat další prvky:

- Direktivy – začínají znaky <!, jsou určeny pro zpracovatele dokumentu (prohlížeč).
- Komentáře – pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje).
- Kód skriptovacích jazyků.
- Definice událostí a kód pro jejich obsluhu.

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. 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.

Značky lze z hlediska významu rozdělit na tři základní skupiny:

- 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>`). 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 (``). Od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu. [13]

6.2 XHTML

XHTML (eXtensible HyperText Markup Language) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý W3C. Původně se předpokládalo, že se stane nástupcem jazyka HTML, jehož vývoj byl verzí 4.01 ukončen. V roce 2007 však došlo k založení pracovní skupiny, která má za cíl vytvořit novou verzi HTML, která ponese označení HTML 5 a její XML variantu XHTML 5. Vedle toho paralelně pokračuje i vývoj XHTML 2.0.

6.2.1 Verze jazyka XHTML

XHTML 1.0

První specifikace, jejíž cílem bylo převedení staršího jazyka HTML tak, aby vyhovoval podmínkám tvorby XML dokumentů a přitom byla zachována zpětná kompatibilita. Existuje ve třech verzích: Strict, Transitional a Frameset.

XHTML 1.0 Strict

Používá se, pokud chceme strukturovaný dokument osvobozený od formátovacích značek souvisejících s rozvržením stránky. Předpokládá se jeho užití společně s CSS, které vám umožní dosáhnout potřebných grafických efektů. Nicméně i tato verze obsahuje formátovací elementy, například `` nebo `<i>` a naopak zavrhuje některé sémantické elementy, například `<menu>`.

XHTML 1.0 Transitional

Je přechodným DTD pro webové stránky, který umožní používat překonané tagy. Je vhodný pro formátování stránek vytvářených pro staré prohlížeče, které nerozumí kaskádovým stylům CSS nebo chcete-li používat ve svých dokumentech některé zavržené, ale sémantické elementy, například již zmíněné `<menu>`.

XHTML 1.0 Frameset

Umožňuje používat zastaralé značky jako XHTML 1.0 Transitional a přidává podporu pro rámce. V dnešní době by se mělo rámcům pokud možno vyhýbat.

Dalším krokem ve vývoji XHTML byla modularizace s cílem dosáhnout vyšší flexibility napříč uživatelskými agenty (WWW prohlížeče, mobilní zařízení, tiskárny, čtečky apod.).

XHTML Basic 1.1

Příklad minimální sady modulů potřebné k vytvoření XHTML dokumentu, která je cílená na mobilní aplikace.

XHTML Mobile Profile

XHTML Mobile Profile - někdy taky XHTML MP je postaveno na základě XHTML Basic a je určeno pro použití v mobilních telefonech. Někdy je také označováno jako WAP 2.0. XHTML MP podporuje na rozdíl od WAP 1.0 a WAP 1.1 barvu a barevné obrázky ve formátech GIF, JPEG a PNG.

XHTML 1.1

Příklad rozsáhlé sady modulů pro komplexnější tvorbu XHTML dokumentů. Vynechává již prakticky všechny prezentační vlastnosti. Je velice podobné XHTML 1.0 Strict, ale na rozdíl od něj může vzhledem ke své modularizaci sloužit jako základ budoucím rozšířeným dokumentům z rodiny XHTML. Je výslovně zakázáno odesílat takto zformátovaný dokument s MIME typem text/html, ale je nutno ho odesílat s MIME typem application/xhtml+xml.

XHTML – Print

Zaměření na tiskový výstup.

XHTML 2.0

Není zamýšleno tak, aby bylo zpětně kompatibilní se svými předchůdci.

XHTML 5

Součást specifikace HTML5. Oproti předchozím verzím obsahuje řadu rozšíření (značky pro audio, video, možnost tvorby offline webových aplikací a další).

6.2.2 Rozdíly mezi XHTML a HTML

V XHTML na rozdíl od HTML musí být všechny tagy ukončené a to včetně nepárových jako jsou <meta>, <link>,
, <hr> nebo . Zápis může mít více podob. Buď použijeme klasické (a validní) nebo zkrácené nebo mírně upravené

. První způsob se nedoporučuje používat, zasíláme-li XHTML dokument s typem text/html. Druhý způsob, bez mezery, se nedoporučuje používat kvůli postarším prohlížečům, které by v takovém případě mohly vynechat poslední atribut, je-li nějaký uvedený.

V XHTML na rozdíl od HTML musí být všechny tagy a jejich atributy zapsány malými písmeny a to z toho důvodu, že jsou takto deklarované v odkazované DTD a X(HT)ML je case sensitive, tedy záleží na velikosti písem. Všechny hodnoty atributů musí být uzavřeny do uvozovek. Dokument musí začínat XML deklarací. Její použití není povinné, pokud je dokument kódován v UTF8 nebo pokud je kódování určeno vyšším protokolem (např. HTTP).

Pokud potřebujeme pracovat s rámy, můžeme deklarovat XHTML 1.0 Frameset a pro jednotlivé stránky XHTML 1.0 Transitional.

XHTML dokument bychom měli zasílat s jiným MIME typem než klasické HTML dokumenty. [14]

7 CSS

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

7.1 Verze CSS

CSS 1

První oficiální verze CSS specifikace byla vydána konsorciem W3C v prosinci 1996. Mezi podporované vlastnosti patřily například:

- Vlastnosti fontů – velikost, tloušťka, ...
- Barva textu a pozadí
- Mezery mezi písmeny, slovy a řádky
- Zarovnání textů, obrázků, tabulek a dalších elementů
- Vlastnosti margin, border, padding a pozicování

CSS 2

Druhá verze CSS 2 byla publikována v květnu 1998. Přinesla spoustu nových vlastností, mezi něž patřilo absolutní a relativní pozicování, podpora textu zprava doleva, nové vlastnosti fontů a také zavedla koncept stylů pro různá média.

CSS 2.1

Revize CSS 2.1 opravila chyby CSS 2, byly odstraněny málo podporované vlastnosti a naopak přidány ty, které již prohlížeče využívaly. Ačkoli je ve vývoji již dlouhou řadu let, stále se nachází ve fázi Candidate Recommendation.

CSS 3

Třetí verze CSS je zatím ve fázi vývoje. CSS 3 má být modularizované a pravděpodobně se bude skládat z několika oddělených a samostatně schválených doporučení. [15]

7.2 Syntaxe

Předpis stylu se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu, nebo skupiny elementů. Pravidlo začíná selektorem, třídou nebo identifikátorem. Následuje seznam deklarácí, které určují vzhled daného prvku. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem (tj. za poslední deklarací středník už být nemusí).

7.2.1 Selektor

Selektor specifikuje element nebo skupinu elementů. Určuje, kterých tagů se bude pravidlo týkat.

```
div
{
    color: red;
}
```

Obrázek 1: Příklad selektoru

7.2.2 Třída

Deklarace třídy začíná tečkou. Ta vyjadřuje, že deklarace se nebude týkat HTML tagu, ale třídy. Atribut class (třída) se může použít u libovolného tagu a může se v dokumentu vyskytovat i vícekrát, na rozdíl od identifikátoru, který by měl být přítomen pouze jedenkrát.

```
.cerveny
{
    color: red;
}
```

Obrázek 2: Příklad třídy

7.2.3 Identifikátor

Pro jednoznačný popis nějakého elementu existuje univerzální atribut ID. I jemu se může přiřadit nějaká deklarace, na rozdíl od třídy však nezačíná tečkou, ale dvojkřížkem #. V těle dokumentu by se měl element s daným identifikátorem vyskytovat jen jednou. Z hlediska CSS se identifikátor chová stejně jako třída.

```
#cerveny
{
    color: red;
}
```

Obrázek 3: Příklad identifikátoru

8 SEO

Zkratka SEO pochází z anglického Search Engine Optimization, tedy optimalizace pro vyhledávače. Vyhledávače jako Google, Atlas či Seznam využívají postupů, které určují relevanci webu ve vztahu k vyhledávanému výrazu. Hledáte-li například sousloví „optimalizace pro vyhledávače“, vyhledávač najde ve své databázi všechny stránky, které tuto frázi obsahují, a interně si je ohodnotí. Ty, které mají největší hodnotu, poté zobrazuje jako první a ty s menší hodnotou pod nimi. [16]

Vyhledávače pomocí tzv. botů (crawlerů, robotů) stahují v různých časových intervalech obsah webových stránek a dalších dokumentů, které pak dále analyzují, indexují a ukládají do databáze.

Přesné metody a algoritmy, podle nichž vyhledávač stránky hodnotí, jsou obvykle přísně střeženým tajemstvím každé firmy. Základní mechanismus je zřejmý - pozice stránky ve výsledcích vyhledávání závisí na tom, do jaké míry podle algoritmu vyhledávacího stroje splňuje očekávání vyhledávajícího uživatele. Na základě obecně známých informací o těchto algoritmech lze odvodit, že pozici stránky ve výsledcích vyhledávání může napomoci, pokud na stránku odkazuje hodně dalších stránek, pokud se vyhledávaný výraz na stránce vyskytuje vícekrát, na důležitých místech (např. v titulku) apod. [17]

Vyhledávače navíc tyto algoritmy neustále zdokonalují, mění svá kritéria a následně i řazení výsledků, a to někdy i několikrát ročně. Stránky je proto nutné i nadále kontrolovat a vylepšovat. Účinná optimalizace webu pro vyhledávače je dlouhodobý proces, který se však bohatě vyplatí, protože přináší vysokou návštěvnost stránek, a tedy i mnoho nových zákazníků.

Z průzkumů vyplývá, že přes 90% uživatelů si vybere stránku, kterou navštíví, právě z výsledků zobrazených na první stránce výpisu. 80% z nich zaměří svou pozornost na prvních 5 odkazů, 20% uživatelů si prohlédne všechny odkazy na první stránce. [16]

Jestliže jsou například vaše stránky při vyhledávání určitého klíčového slova až na dvacáté stránce výpisu a přichází vám jeden člověk denně, lze očekávat přibližně desetinásobné zlepšení výsledku, pokud se vaše stránky pomocí optimalizace pro vyhledávače posunou na první stranu výpisu. [16]

8.1 Používané metody SEO

Kvalitní a unikátní obsah

Webová stránka by měla mít kvalitní a jedinečný obsah, pokud možno pravidelně aktualizovaný. Na každou stránku by měl vést alespoň jeden odkaz.

Používání (X)HTML značek (tagů) podle normových předpisů

Tvůrce stránky by měl dodržovat stanovený standard HTML či XHTML. Validní stránky jsou pro vyhledávače atraktivnější a řadí je výše.

Používání titulků, nadpisů a popisů (title, h1, alt)

Uvede-li autor v titulku konkrétní popis stránky či rovnou klíčová slova, získává stránka další významné plus. Vyhledávače připisují velkou váhu také nadpisům, proto by stránka měla být správně strukturovaná a v nadpisy by měly obsahovat důležitá slova. Tag alt by měl být u obrázků, protože vyhledávače obrázky neindexují.

Krátká a neměnná URL adresa

Uvedení krátké URL adresy podpoří zájem ostatních uživatelů o odkazování na stránku. Příliš dlouhá a lidem nesrozumitelná URL adresa naopak od odkazování spíše odrazuje. Přítomnost klíčového slova v URL adrese může u některých vyhledávacích strojů zvýšit umístění stránky ve výsledcích hledání těchto slov. Požadavek na neměnnost adres vyplývá z fungování vyhledávacích strojů i z požadavku na budování zpětných odkazů. [18]

Budování zpětných odkazů

Zpětnými odkazy (tzv. back-links) rozumíme odkazy směřující na váš web z jiných stránek na internetu. Zpětné odkazy považují vyhledávače za jakési pozitivní reference, které jim doporučují odkazovaný web. Čím více takových referencí na váš web směřuje, tím je pro uživatele v očích vyhledávače kvalitnější a přínosnější. S narůstajícím počtem a kvalitou odkazů směřujících na vaše stránky se stáváte pro vyhledávače zajímavějšími. [19]

Správné používání souboru robots.txt

Textový soubor robots.txt, umožňuje správci webu zakázat nebo povolit přístup indexovacích botů. Tento soubor se musí nacházet v kořenovém adresáři webu.

Používání meta tagů description, keywords, author

Vhodným používáním klíčových slov v meta description získá stránka v hodnocení robota další kladné body. Pokud se klíčová slova dále v obsahu stránky neopakují, nebo je jich naopak moc, ztrácí na prioritě. Meta description je použit u popisu stránky ve výsledcích vyhledávání SERP. Description a keywords by měly být na každé stránce jiné, a pokud možno by měly reflektovat obsah stránky. [18]

8.2 Zakázané metody SEO

Cloaking

Cloaking je technika podávání jiného obsahu vyhledávacímu botovi než uživateli. K tomu se využívá speciální software, který porovná IP adresu příchozí žádosti s databází IP adres známých vyhledávacích robotů. V případě, že IP adresa je nalezena v databázi, software podstrčí botovi speciálně optimalizovanou stránku.

Doorway pages

Stránka vytvořená speciálně a pouze za účelem zisku vysokého hodnocení ve vyhledávacích. Většinou se jedná o uměle seskládanou stránku obsahující desítky opakovaných klíčových slov. [20]

Duplicitní obsah

Tentýž obsah umístovaný na několik domén bývá ve vyhledávání penalizován negativním hodnocením daného webu. Typickým příkladem duplicitního obsahu jsou totožné stránky odkazované různými adresami (např. <http://example.com> a <http://www.example.com>).

Link farmy

Velký počet stránek, který je často generován i automaticky na základě slovníků a vzájemně prolinkován mezi sebou. Zpětnými odkazy se snaží o nekalé zvýšení pozice ve výsledcích. Často jsou podobné stránky umístovány na domény, které mají velmi dobré hodnocení, ale expirovaly. [21]

Skryté texty

Do stránek je umístováno velké množství slov, které nemají se zaměřením webu vůbec nic společného, ale patří k často vyhledávaným dotazům. Tento irelevantní text je skrýván

použitím shodné barvy pro text a pozadí v dokumentu nebo je minimalizována jeho velikost na jeden pixel.

Tapetování katalogů

Vkládání všech stránek z jednoho webu do vybraného katalogu. Pro katalogy kontrolované administrátorem tato taktika příliš fungovat nebude, nicméně při automatickém vkládání odkazů do katalogu jej může nepříjemně vytapetovat. [21]

II. PRAKTICKÁ ČÁST

9 STRUKTURA TURNAJOVÉHO SYSTÉMU

Cílem diplomové práce bylo vytvořit turnajový systém určený prvotně pro počítačové hry, ale nejen pro ně. Jedním z požadavků byla také dostupnost služby – to znamená, že turnaj si může založit kdokoli. Samozřejmě při velkém množství hráčů by se časem v turnajích nedalo vyznat, proto byl navržen dvojitý systém turnajů – veřejné a soukromé. Veřejné turnaje jsou jakousi obdobou dotovaných turnajů na konkurenčních serverech. Jsou viditelné v seznamu turnajů a může se do nich přihlásit kdokoli. Takový turnaj smí založit pouze supervizor nebo administrátor se schválením supervizora. Naopak soukromý turnaj si může kterýkoli uživatel, tento turnaj se však neobjeví v seznamu turnajů a přihlásit se do něj může jen ten, kdo o něm ví, nebo ho najde pomocí vyhledávání.

Na výběr jsou připraveny dva druhy turnajů – vyřazovací (playoff) a liga. Cílem bylo, aby průběh turnajů byl co nejvíce automatizovaný, což v případě ligy není problém, ale u playoff je třeba zajistit postup hráčů turnajovým pavoukem. Podařilo se dosáhnout toho, že do průběhu playoff není třeba nijak zasahovat, zápasy jsou provázány tak, že hráči automaticky postupují dál jen na základě zadaných výsledků. Správa turnaje je tak velmi snadná.

9.1 Ovládání a popis funkcionality systému

Pro rozvržení stránky je použit layout skládající se z hlavičky, menu, levého a pravého sloupce a patičky. Hlavička obsahuje grafický banner. Menu se mění v závislosti na tom, zda je uživatel přihlášen a jakou úroveň oprávnění má. V pravém sloupci se nachází přihlašovací / odhlašovací formulář a v případě přihlášeného uživatele také seznam týmů, kterých je členem, turnajů, kterých se účastní a administrovaných turnajů. Sloupec obsahuje též informace o verzi PHP, MySQL, webhostingu a validitě stránek. Největší část stránky zabírá levý sloupec, který slouží pro zobrazení podstránek. Stránku uzavírá patička s informací o autorovi. Na Obrázku 4 je zobrazena úvodní stránka systému.



Obrázek 4: Hlavní stránka systému

9.1.1 Registrace uživatele

Pokud chce uživatel využívat služeb systému, je třeba, aby si vytvořil účet. Na registrační stránce vyplní formulář, kam zadá uživatelské jméno, heslo, potvrzení hesla a email. Uživatelské jméno a email musí být unikátní. Nepovinné položky jsou křestní jméno a příjmení. Pokud jsou všechna pole správně vyplněna a registrace se zdaří, je na zadanou emailovou adresu odeslán email s odkazem na dokončení registrace. Teprve po potvrzení registrace je možné, aby se uživatel přihlásil do systému.

9.1.2 Přihlášení

Přihlásit do systému se lze buď formulářem v pravé části stránky nebo přes přihlašovací stránku zadáním uživatelského jména a hesla. Není-li uživatel ověřený nebo je-li blokový, nemůže se přihlásit.

9.1.3 Registrace týmu

Přihlášený uživatel může zakládat vlastní týmy. Stačí zadat název týmu, který se nesmí shodovat s žádným existujícím, a odeslat. Uživatel se automaticky stane administrátorem každého týmu, který založí.

9.1.4 Založení turnaje

Každý registrovaný uživatel smí vytvářet vlastní turnaje. Je třeba zadat název turnaje, popis, druh (liga / playoff), typ (single player / týmový), maximální počet hráčů, začátek a konec registrace. Běžní uživatelé smí zakládat pouze soukromé turnaje, tzn. že nebude vidět v seznamu turnajů, ale bude možné je vyhledat. Administrátoři a supervizoři mají možnost výběru, zda bude turnaj veřejný nebo soukromý. Veřejné turnaje založené administrátorem musí následně schválit supervizor.

9.1.5 Administrace turnaje

Turnaj se může nacházet v jedné z následujících fází:

Registrace hráčů do turnaje

V této fázi je možno měnit termín konce registrace (např. pokud není k danému termínu registrováno dostatek hráčů), rušit registrace přihlášených hráčů a ukončit fázi registrace a přejít k rozlosování zápasů.

Rozlosování zápasů

U ligy je lze nastavit počet vzájemných zápasů. U playoff je nastavení více – počet vítězných zápasů v každém kole a nasazení jednotlivých hráčů do pavouka turnaje. Pokud zůstane nevyplněné, nasadí se náhodně. Dále je možno nastavit datum a čas prvního zápasu a počet dní, po kterých se bude hrát – tyto hodnoty se následujícím kroku použijí pro vyplnění tabulky termínů zápasů, které je možno ručně upravit. Po rozlosování zápasů a stanovení termínů zápasů a zadání výsledku je možno turnaj spustit.

Turnaj probíhá

Když turnaj probíhá, spočívá role administrátora především v hlídání termínů zadání výsledků a řešení sporů o výsledek. U ligy je průběh jasný – účastníci všech zápasů jsou předem dáni, ale ani v případě playoff není nutné do průběhu turnaje nijak zasahovat –

postup turnajovým pavoukem probíhá automaticky, na základě výsledků zápasů. Prošlé i sporné zápasy jsou přímo viditelné v administraci a po rozkliknutí lze přímo zadat jejich výsledek, stejně tak u všech zápasů v turnaji. Turnaj je možné zakončit po odehrání všech zápasů nebo i předčasně nebo rovnou zrušit.

Turnaj skončil

Po skončení turnaje může administrátor i nadále zadávat výsledky zápasů.

Turnaj byl zrušen

Pokud je turnaj zrušen, není již možné do něj jakkoli zasahovat.

9.1.6 Registrace do turnaje

Je-li turnaj ve fázi registrace a není zatím přihlášen maximální počet hráčů, může se hráč zaregistrovat do turnaje. Jednotlivec se nemůže registrovat do týmových turnajů a naopak. Tým do turnaje může přihlásit pouze administrátor daného týmu. Totéž platí pro zrušení registrace.

9.1.7 Účast v turnaji z pohledu hráče

Běžný hráč, na rozdíl od administrátora turnaje, smí zadávat jen výsledky zápasů, kterých účastní. U týmových turnajů smí zadávat výsledky svého týmu jen administrátor daného týmu.

Hráč zadá výsledek a čeká se, až zadá výsledek i soupeř. Pokud se shodují, je výsledek platný, v opačném případě musí rozhodnout administrátor turnaje a u daného kola, kam zápas patří, se do té doby objeví vykřičník. Když hráč nezadá výsledek do patřičného termínu, bude mít u daného kola otazník. Otazník se zobrazí i v případě, že soupeř již zadal výsledek, ale hráč ještě ne. Po skončení turnaje již není možné výsledky zadávat.

9.1.8 Členství v týmu

Každý hráč může požádat o členství v kterémkoli týmu. Pak je na administrátorovi týmu, zda žádost přijme nebo zamítne. Hráč může zrušit své členství v týmu, aniž by byl nutný souhlas administrátora týmu.

9.1.9 Přepínání týmu

Je-li uživatel členem více týmů, může se mezi nimi jednoduše přepínat pomocí odkazů v pravém sloupci.

9.1.10 Administrace týmu

Do administrace týmu má pochopitelně přístup pouze administrátor týmu. Může vyškrtnout uživatele z týmu (ne však sebe), přijmout nebo zamítnout požadavky uživatelů na členství v týmu a také předat administraci týmu jinému uživateli.

9.1.11 Administrace uživatele

Uživatel má možnost editovat své křestní jméno, příjmení a email a také změnit heslo.

9.1.12 Možnosti supervizora

Supervizor je hlavní správce systému. Jeho úloha je tedy starat se o chod systému. Může vstupovat do veškerých administrací turnajů, týmů i uživatelů. Nemá možnost účastnit se turnajů, ani být členem týmů. Jako jediný má přístup ke správě uživatelů, kde může resetovat hesla a tzv. zabanovat (zakázat přihlášení) a zrušit ban (povolit přihlášení) uživatelů. Dalším úkolem je schvalování veřejných turnajů.

9.1.13 Možnosti administrátora

Administrátor má přístup ke správě všech turnajů, nikoli však týmů a uživatelů. Smí zakládat veřejné turnaje, které následně musí schválit supervizor. Na rozdíl od supervizora se může účastnit turnajů a být členem týmů.

9.2 Struktura souborů systému

Pro přehlednost a lepší orientaci jsou soubory systému rozděleny do složek. Soubory kaskádových stylů se nachází ve složce css/, obrázky a grafika jsou ve složce img/, adresář inc/ obsahuje především soubory s nastavením, zachytáváním akcí a funkcemi a dále soubory, které jsou vkládány do šablony stránky. Ve složce menu/ se nachází soubory nutné pro fungování menu stránky a konečně adresář pages/ obsahuje jednotlivé podstránky systému. Pro vysvětlení významu a funkce každého souboru slouží následující přehled:

/ (kořenový adresář systému)

- **.htaccess** – nastavení chybové zprávy 404 File Not Found a především nastavení `mod_rewrite` a prepisovacích pravidel pro tzv. SEO Cool URL.
- **favicon.ico** – ikona, zobrazující se v adresovém řádku prohlížečů
- **robots.txt** – nastavení chování vyhledávacích robotů
- **index.php** – šablona stránky, kam se pomocí `include` vkládají požadované soubory, jako je hlavička, menu, levý a pravý sloupec a patička; obsahuje též volání souborů s nastavením, funkcemi a zachytáváním akcí

/css

- **layout.css** – prostorové rozvržení stránky
- **tags.css** – definice tříd kaskádových stylů

/img

- obsahuje obrázky a grafiku použitou na stránce

/inc

- **.htaccess** – omezení přístupu zvenčí
- **/phpmailer** – třída pro posílání emailů z PHP [22]
- **/tiny_mce** – JavaScriptový WYSIWYG editor [23]
- **actions.php** – zachycuje akce z formulářů a volá příslušné funkce
- **functions.php** – obsahuje definice funkcí, viz. kapitola 9.3
- **settings.php** – nastavení PHP, SMTP, připojení k MySQL atd.
- **header.php** – hlavička stránky
- **menu.php** – menu
- **left_column.php** – levý sloupec stránky – zařizuje vkládání požadovaných stránek, případně chybových stránek; obsahuje seznam stránek, které je povoleno vložit – tzv. `whitelist`

- **right_column.php** – pravý sloupec stránky – zobrazuje přihlašovací / odhlašovací formulář a v případě, že je uživatel přihlášen, tak i seznam, týmů a turnajů, kde je uživatel registrovaný
- **footer.php** – patička stránky

/menu

- obsahuje soubory, potřebné pro funkčnost menu, které bylo vytvořeno programem Sothink DHTML Menu [24]

/pages

- **.htaccess** – omezení přístupu zvenčí
- **confirm_tournaments.php** – zobrazení seznamu turnajů založených administrátory a tyto je možno je schválit nebo zamítnout; přístupné pouze pro uživatele s oprávněním supervizor
- **confirm_user_registration.php** – potvrzení registrace uživatele
- **create_tournament.php** – založení nového turnaje
- **error1.php** – chybová stránka – uživatel není přihlášen
- **error2.php** – chybová stránka – nedostatečná práva pro přístup ke stránce
- **error3.php** – chybová stránka – pokus o prolomení zabezpečení formuláře
- **error4.php** – chybová stránka – nelze se připojit k MySQL serveru
- **error5.php** – chybová stránka – objekt neexistuje
- **error6.php** – chybová stránka – stránka neexistuje
- **error7.php** – chybová stránka – změnila se IP adresa klienta
- **error8.php** – chybová stránka – nežádoucí reload stránky
- **home.php** – úvodní stránka
- **league.php** – detaily ligy
- **league_admin.php** – administrace ligy; přístupné pouze pro administrátora dané ligy

- **league_match.php** – detaily ligového zápasu a zadání výsledku
- **login.php** – přihlašovací stránka
- **playoff.php** – detaily playoff
- **playoff_admin.php** – administrace playoff; přístupné pouze pro administrátora daného playoff
- **playoff_match.php** – detaily playoff zápasu a zadání výsledku
- **register_team.php** – registrace týmu
- **register_user.php** – registrace uživatele
- **search.php** – vyhledávání
- **team.php** – detaily týmu
- **team_admin.php** – administrace týmu; přístupné pouze pro administrátora daného týmu
- **tournaments.php** – seznam veřejných turnajů
- **user.php** – detaily uživatele
- **user_admin.php** – administrace uživatele; přístupné pouze pro daného uživatele
- **user_management.php** – správa uživatelů; přístupné pouze pro uživatele s oprávněním supervizor

9.3 Soubor functions.php

Nejdůležitější soubor celého systému je functions.php, jelikož obsahuje definice všech funkcí. V následující části bude vysvětlen význam a případně i princip každé funkce.

- **check_session()**
Ověřuje existenci session (tzn. zda je uživatel přihlášen) a také kontroluje IP adresu klienta, jestli se shoduje s IP adresou při přihlášení.
- **check_user(\$user)**
Testuje, zda je přihlášen uživatel \$user.

- **check_admin()**
Ověřuje, zda je uživatel administrátor.
- **check_supervisor()**
Ověřuje, zda je uživatel supervizor.
- **check_team_admin(\$team)**
Testuje, jestli je přihlášený uživatel správce týmu \$team.
- **check_tournament_admin(\$tournament)**
Testuje, je-li přihlášený uživatel administrátor turnaje \$tournament.
- **check_security(\$test,\$hash)**
Slouží pro ověření zabezpečení formulářů (viz. kapitola 11.2). Z proměnná \$test představující data, která je třeba ověřit, a z bezpečnostního hesla je vytvořen hash, který se porovná s proměnnou \$hash. Pokud jsou různé, vrací chybu.
- **check_reload()**
Porovná reload_protection hash z formuláře a ze session a pokud jsou různé, znamená to, že bylo v prohlížeči stisknuto tlačítko Reload nebo F5, čímž mohlo dojít k nežádoucímu vícenásobnému odeslání formuláře. Zjistí-li to, vrací chybu.
- **set_reload_protection()**
Vygeneruje sha1 reload_protection hash aktuálního času a bezpečnostního hesla, který je uložen do session a který je potom vložen do každého formuláře a slouží pro detekci vícenásobného odeslání formuláře.
- **check_date(\$date)**
Pomocí regulárního výrazu a PHP funkce checkdate() prověří, jestli je proměnná \$date datum ve formátu dd.mm.rrrr.
- **check_time(\$time)**
Pomocí regulárního výrazu a ověření, zda hodnoty spadají do povoleného rozsahu, otestuje, jestli je proměnná \$time odpovídá, času ve formátu hh:mm.
- **connect_db(\$mysql_server,\$mysql_username,\$mysql_password,\$mysql_db)**
Funkce pro připojení k databázi. Parametry připojení lze nastavit v souboru inc/settings.php. Pokud se nelze připojit, vrátí chybu.

- **seo(\$text)**

Využívá se pro tvorbu tzv. pěkných URL. Funkce byla převzata z [25]. Odstraní z textu diakritiku, převede písmena na malá a všechny znaky kromě písmen a číslic změní na pomlčku. Pokud je více pomlček za sebou, ponechá pouze jednu a nakonec odstraní pomlčky ze začátku a konce textu.
- **login(\$username,\$password)**

Ověří, zda existuje uživatel s daným jménem a heslem a pokud ano, vytvoří novou session, do které uloží uživatelské jméno, id uživatele, IP adresu, úroveň oprávnění, a pole týmů, kterých je členem.
- **logout()**

Zruší session proměnné a následně i danou session, čímž je uživatel odhlášen ze systému.
- **register_user(\$username,\$password,\$confirm_password,\$firstname,\$lastname,\$email)**

Slouží pro registraci nového uživatele. Nejdříve zkontroluje uživatelské jméno, které smí obsahovat pouze znaky A-z, 0-9, tečku a podtržítka, musí být 3-20 znaků dlouhé a následně ověří, jestli již uživatel se stejným jménem není registrován. Následuje kontrola hesla, pro které platí stejná pravidla, jako pro uživatelské jméno, jen délka musí být minimálně 6 znaků. Z hesla se vytvoří hash, pomocí funkce sha1(). Jméno a příjmení jsou nepovinné údaje, ale pokud jsou zadány, proběhne kontrola na povolené znaky, což jsou v tomto případě velká a malá písmena i s diakritikou a tečka. Nakonec se zkontroluje email, který musí být ve správném formátu a nesmí se již v systému vyskytovat. Pokud jsou všechny údaje správné, je uživatel vložen do databáze a na jeho emailovou adresu je odeslán email pro potvrzení registrace.
- **update_user(\$firstname,\$lastname,\$email)**

Umožňuje změnu jména, příjmení a emailu. Po kontrole, která je shodná jako při registraci, jsou tyto údaje změněny.
- **change_password(\$old_password,\$new_password,\$confirm_password)**

Slouží pro změnu hesla uživatele. Nejdříve se ověří, zda je správně zadané staré

heslo, potom se zkontroluje nové heslo, stejně jako při registraci, a pokud vyhovuje, vytvoří se hash, kterým se nahradí původní.

- **register_team(\$name)**

Funkce, jež vkládá nový tým do databáze. Před tím se ovšem ověří délka názvu, která musí mít 3-50 znaků, zda se v názvu nevyskytují nepovolené znaky a zda již neexistuje tým se stejným názvem. Uživatel, který tým zakládá, se automaticky stává jeho administrátorem.

- **create_tournament(\$name,\$description,\$kind,\$type,\$n_players,
\$registration_begin_date,\$registration_begin_time,\$registration_end_date,
\$registration_end_time,\$private)**

Funkce pro založení nového turnaje. Opět se ověřuje délka názvu a nepovolené znaky. Dále se ověřuje, zda je vyplněn druh a typ turnaje, počet hráčů, který smí být v rozsahu 3-64 a správnost data a času registrace do turnaje. Uživatel zakládající turnaj se stává jeho administrátorem.

- **switch_team(\$team)**

Přepíná uživatele mezi jeho týmy.

- **register_tournament(\$tournament)**

Zařizuje registraci do turnaje. Nejdříve ověří, zda již není registrován plný počet hráčů. Poté přihlásí daného uživatele nebo tým do turnaje.

- **unregister_tournament(\$tournament)**

Ruší turnajovou registraci uživatele nebo týmu.

- **unregister_tournament_admin(\$tournament,\$player)**

Slouží administrátorovi turnaje pro zrušení turnajové registrace hráčů.

- **send_confirm_mail(\$email)**

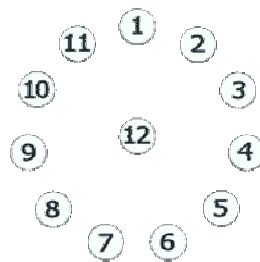
Tato funkce zasílá na danou emailovou adresu potvrzovací email, který slouží pro dokončení registrace uživatele. Využívá při tom freeware třídu PHPMailer [22]. Nastavení SMTP serveru se nachází v souboru inc/settings.php.

- **confirm_user_registration(\$hash)**

Slouží pro dokončení registrace uživatele. Parametrem je hash emailové adresy,

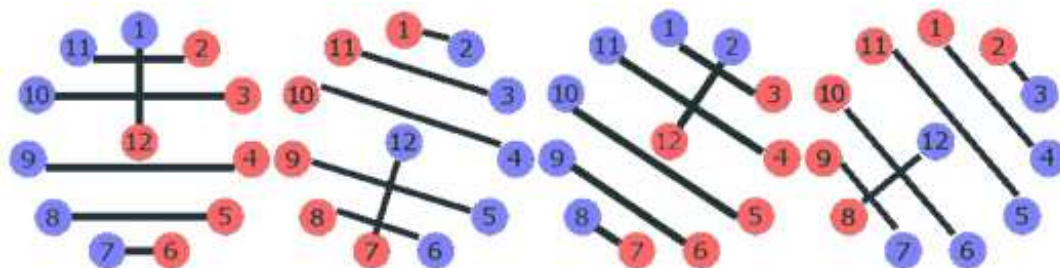
podle něhož se najde daný uživatel v databázi. Po ověření se již může uživatel přihlásit do systému.

- **request_team_membership(\$team)**
Odešle požadavek na přijetí uživatele do daného týmu. O přijetí musí poté rozhodnout administrátor týmu.
- **accept_team_membership(\$user)**
Přijme uživatele do týmu.
- **cancel_team_membership(\$user,\$team)**
Odebere uživatele z týmu.
- **refuse_team_membership(\$user)**
Zamítne žádost uživatele o členství v týmu.
- **change_team_admin(\$team,\$user)**
Předá administraci týmu jinému uživateli.
- **tournament_status(\$tournament,\$status)**
Změní status turnaje. Turnaj se může nacházet v jednom z následujících stavů: čeká na schválení, probíhá registrace, rozlosování zápasů, turnaj probíhá, turnaj je dokončen, turnaj byl zrušen a turnaj byl zamítnut.
- **generate_league_matches(\$tournament,\$n_mutual,\$first_match,\$default_play_time,\$default_days)**
Jedná se o jednu z nejdůležitějších funkcí, protože má na starost rozlosování ligových zápasů. Po kontrole správnosti vstupních parametrů dojde k promíchání hráčů. Vlastní generování zápasů funguje dle principu popsaného na [26] a [27]. Náhodně je vybrán hráč, který bude ve středu pomyslného kruhu pro losování, jak je znázorněno na následujícím obrázku:



Obrázek 5: Rozložení hráčů pro rozlosování zápasů

Je-li počet hráčů lichý, je doplněn jeden virtuální hráč. Následuje samotné generování. V prvním kole hraje hráč č. 12 doma s hráčem č. 1, který se tak v daném kole stává tzv. pivotem. Další zápasy v daném kole jsou 2 vs. 11, 3 vs. 10, 4 vs. 9, 5 vs. 8 a 6 vs. 7. Hráči napravo od pivota (2,3,4,5 a 6) jsou v daném kole vždy domácí. V následujícím kole je role pivota posunuta o 180°, čímž se dosáhne toho, že ti, co byli domácí, nyní hrají venku a naopak. Hráč ze středu kruhu je nyní hostující a pivot domácí. Je tak zajištěna maximální spravedlnost podmínek pro všechny hráče. Zápasy kola tedy vypadají následovně: 7 vs. 12, 8 vs. 6, 9 vs. 5, 10 vs. 4, 11 vs. 3 a 1 vs. 2. V dalším kole je pivot hráč č. 2, potom č. 8 a tak dále, dokud se v roli pivota nevystřídají všichni hráči. Graficky tento proces ukazuje Obrázek 3 (červený tým hraje doma, modrý venku).



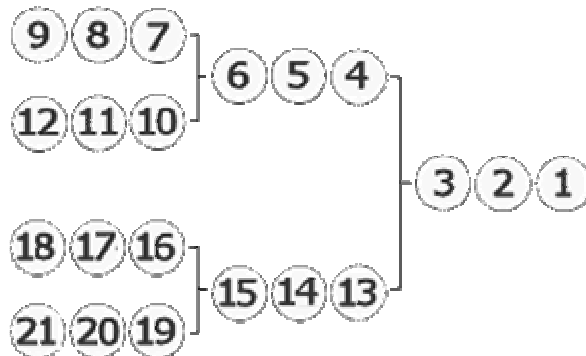
Obrázek 6: Losování ligových kol

Před uložením zápasu do databáze je ověřeno, zda v něm nefiguruje jako soupeř virtuální hráč, takový zápas není uložen a daný hráč má v daném kole volno.

- **generate_playoff_matches(\$tournament,\$n_winning,\$seed,\$first_match,\$default_play_time,\$default_days)**

Další stěžejní funkcí je generování zápasů playoff. Po kontrole vstupních parametrů je načteno pole hráčů (\$players) a seřazeno podle pole nasazení (\$seed). Poté je doplněn počet hráčů na nejbližší násobek 2^n virtuálními hráči a je zjištěn počet kol, potřebný pro odehrání playoff. Dále je zavolána funkce **generate_playoff**, která rekurzivně vygeneruje prázdného turnajového pavouka od finále po základní kolo, včetně návaznosti zápasů. Pokud je více vzájemných zápasů v jednom kole, jsou tyto seskupeny pomocí \$grouping, která má pro každou skupinu jinou hodnotu, a proměnná \$subround určuje pořadové číslo zápasu v daném kole. Pořadí

generování zápasů pro tři zápasy v kole (tzn. na dva vítězné) graficky ukazuje Obrázek 7:



Obrázek 7: Postup generování zápasů playoff

Dalším krokem je obsazení hráčů do prvního kola dle nasazení. Budeme se držet předchozího obrázku, který znázorňuje pavouka pro 8 hráčů. První nasazený se utká a osmým ve skupině zápasů 9, 8, 7. Druhý se sedmým jsou nasazení do zápasů 18, 17, 16. Třetí se šestým se utkají v zápasech 12, 11, 10 a nakonec čtvrtý s pátým obsadí zápasy 21, 20, 19. Potká-li se v prvním kole hráč s virtuálním hráčem, zápasy se uloží do databáze kvůli následnému vykreslování pavouka, ale těmto zápasům je nastaven příznak `virtual`, což znamená, že takové zápasy se nehrají a nejsou v systému ani vidět a hráč automaticky postupuje do druhého kola. Uvedený systém zaručuje to, že nejvýše nasazení hrají proti nenasazeným soupeřům a vzájemně tak na sebe narazí až v pozdějších fázích turnaje (budou-li postupovat favorité).

- **`matches_datetime($tournament,$play_date,$play_time,$result_date,$result_time)`**
Posledním krokem před spuštěním turnaje je nastavení termínů odehrání zápasů a mezního zadání výsledku zápasů. To zajistí tato funkce.
- **`league_match_result($match,$score1,$score2)`**
Slouží pro uložení výsledku ligového zápasu. Nejdříve se zkontroluje, zda je skóre v číselném formátu a v rozmezí 0-999. Pokud soupeř ještě nevložit výsledek, vytvoří v pomocné tabulce nový záznam a čeká, až zadá výsledek i soupeř. Zadal-li již protihráč výsledek, je tento záznam vyhledán v pomocné tabulce a oba výsledky jsou porovnány. Pokud se shodují, výsledek je platný, je aktualizován záznam v tabulce zápasů a záznam z pomocné tabulky je odstraněn. Jsou-li různé, do

záznamu v pomocné tabulce je uložen i druhý výsledek a nastává spor, který musí vyřešit administrátor turnaje.

- **league_match_result_admin(\$match,\$score1,\$score2)**

Administrátor turnaje má možnost zadat výsledek ligového zápasu přímo. To je užitečné, pokud například hráči zadají různé výsledky nebo jeden z nich výsledek nezadá, popřípadě je-li třeba opravit již zadaný výsledek. V těchto případech je volána tato funkce.

- **reset_league_match_result(\$match)**

Vynuluje výsledek ligového zápasu.

- **playoff_match_result(\$match,\$score1,\$score2)**

Slouží pro uložení výsledku playoff zápasu. Nejdříve se zkontroluje, zda je skóre v číselném formátu, v rozmezí 0-999 a zda není remíza, která ve vyřazovacích zápasech být nesmí. Další postup je stejný jako v případě ligového zápasu. Pokud soupeř ještě nevložil výsledek, vytvoří v pomocné tabulce nový záznam a čeká, až zadá výsledek i soupeř. Zadal-li již protihráč výsledek, je tento záznam vyhledán v pomocné tabulce a oba výsledky jsou porovnány. Pokud se shodují, výsledek je platný, je aktualizován záznam v tabulce zápasů a záznam z pomocné tabulky je odstraněn. Jsou-li výsledky zadané oběma hráči různé, do záznamu v pomocné tabulce je uložen i druhý výsledek a nastává spor, který musí vyřešit administrátor turnaje.

Dále je potřeba zjistit, zda již jeden z hráčů nedosáhl potřebný počet vítězství na postup do dalšího kola. Jestli je tomu tak, neodehraným zápasům daného kola je nastaven atribut „virtual“, čímž se zpřístupní, a vítěz je nasazen do zápasů následujícího kola.

- **playoff_match_result_admin(\$match,\$score1,\$score2)**

Podobně jako u ligových zápasů je i u zápasů playoff možnost zadat výsledek přímo administrátorem turnaje. Stejně jako u klasického zadání výsledku je i zde zajištěno, že vítěz automaticky postupuje do dalšího kola. Pokud je výsledek dodatečně upraven a změní-li se tím vítěz daného kola, jsou pak zpřístupněny neodehrané zápasy.

- **reset_playoff_match_result(\$match)**
Vynuluje výsledek playoff zápasu. Pokud by vynulování ovlivnilo postupujícího, jsou opět zpřístupněny neodehrané zápasy.
- **change_registration_end(\$tournament,\$registration_end_date,\$registration_end_time)**
Umožní administrátorovi turnaje změnit konec registrace do turnaje. Toho se dá využít například, pokud se do původního termínu nepřihlásí dostatek hráčů a je tak vhodné prodloužit registraci a počkat, až se přihlásí více hráčů.
- **change_tournament_description(\$tournament,\$description)**
Umožní administrátorovi turnaje změnit konec registrace do turnaje. Toho se dá využít například, pokud se do původního termínu nepřihlásí dostatek hráčů a je tak vhodné prodloužit registraci a počkat, až se přihlásí více hráčů.
- **draw_playoff_bracket(\$tournament)**
Tato funkce zařídí vykreslení turnajového pavouka. Zobrazení je vyřešeno pomocí tabulky, tak jak je ukázáno na následujícím obrázku, kde jsou červenými čarami znázorněny jednotlivé buňky tabulky:

Čtvrtfinále	Semifinále	Finále	Vítěz
adam			
	adam 2 3 1 1		
		eman	
eman 2 1 2 1 3			
	eman 1 0 4 2		
hugo 1 3 1 4 2			
beda			
	beda 2 1		
cyril 1 3 4 2			
	cyril 1 3		
david 2 1 1 0			

Obrázek 8: Vykreslení turnajového pavouka pomocí tabulky

Celý pavouk je nejprve vytvořen v PHP jako dvourozměrné pole a až poté je vykreslen jako tabulka, protože v HTML probíhá vykreslování tabulky shora dolů, ale pavouka je třeba zobrazit zleva doprava.

Nejdříve je zjištěna potřebná výška a šířka tabulky:

$$\text{výška: } 2 \times \text{pocet_hracu} - 1 \quad (1)$$

$$\text{šířka: } 2 \times \text{pocet_kol} \quad (2)$$

Poté následuje vlastní generování. V každém sudém cyklu (čísluje se od 0) je vytvořena buňka s hráčem, v lichých cyklech se tvoří spojnice mezi hráči. Každý cyklus se skládá z $(\text{pocet_hráčů_v_daném_kole} / 2)$ podcyklů, které jsou na Obrázku 7 zarámovány modře. V našem případě je v prvním kole 8 hráčů tzn. 4 podcykly, ve druhém kole 4 hráči tzn. 2 podcykly atd. V každém podcyklu jsou spočítány následující prvky:

1. počet prázdných polí nad prvním hráčem: $2^{aktua\ ln\ i_kolo-1} - 1$ (3)

2. první hráč

3. počet prázdných polí mezi prvním a druhým hráčem:

$$2 \times (2^{aktua\ ln\ i_kolo-1} - 1) + 1 \quad (4)$$

4. druhý hráč

5. počet prázdných polí pod druhým hráčem: $2^{aktua\ ln\ i_kolo-1} - 1$ (5)

6. prázdné pole

Obdobně jsou spočítány spojnice mezi hráči. Dále je zjištěno, kdo je případně vítěz turnaje a je umístěn do pole na pozici $[2 \times počet_kol][počet_hráčů - 1]$, což poté v tabulce odpovídá políčku vpravo uprostřed.

Zbývá jen vykreslit tabulku. To probíhá v podstatě transponováním daného pole v PHP, čehož je dosaženo prohozením indexů při generování tabulky.

- **draw_league_table(\$tournament)**

Vykreslení ligové tabulky probíhá pouze na základě výsledků zápasů. Pomocí následujícího dotazu se zjistí počet vítězství, remíz a proher, +/- skóre a počet bodů každého hráče.

```
select users.id,users.username as name,((select coalesce(sum(score1),0) from
league_match where player1=users.id and tournament='$tournament') + (select
coalesce(sum(score2),0) from league_match where player2=users.id and
tournament='{ $tournament}')) as score1, ((select coalesce(sum(score2),0) from
league_match where player1=users.id and tournament='{ $tournament}') + (select
coalesce(sum(score1),0) from league_match where player2=users.id and
tournament='{ $tournament}')) as score2, ((select coalesce(sum(score1),0)-
coalesce(sum(score2),0) from league_match where player1=users.id and
tournament='{ $tournament}')+(select coalesce(sum(score2),0)-
coalesce(sum(score1),0) from league_match where player2=users.id and
tournament='{ $tournament}')) as diff, ((select count(id) from league_match
where player1=users.id and tournament='{ $tournament}' and score1>score2) +
(select count(id) from league_match where player2=users.id and
tournament='{ $tournament}' and score2>score1)) as win, ((select count(id) from
league_match where player1=users.id and tournament='{ $tournament}' and
score1=score2) + (select count(id) from league_match where player2=users.id and
tournament='{ $tournament}' and score2=score1)) as draw, ((select count(id) from
league_match where player1=users.id and tournament='{ $tournament}' and
score1<score2) + (select count(id) from league_match where player2=users.id and
tournament='{ $tournament}' and score2<score1)) as lose, (((select count(id)
from league_match where player1=users.id and tournament='{ $tournament}' and
score1=score2) + (select count(id) from league_match where player2=users.id and
tournament='{ $tournament}' and score2=score1))+((select 3*count(id) from
league_match where player1=users.id and tournament='{ $tournament}' and
score1>score2) + (select 3*count(id) from league_match where player2=users.id
and tournament='{ $tournament}' and score2>score1))) as points from
users,tournament_register where users.id=tournament_register.player and
tournament_register.tournament='{ $tournament}' order by points desc, diff desc,
score1 desc, win desc";tournament='{ $tournament}' and score2<score1)) as lose,
(((select count(id) from league_match where player1=users.id and
tournament='{ $tournament}' and score1=score2) + (select count(id) from
league_match where player2=users.id and tournament='{ $tournament}' and
score2=score1))+((select 3*count(id) from league_match where player1=users.id
and tournament='{ $tournament}' and score1>score2) + (select 3*count(id) from
league_match where player2=users.id and tournament='{ $tournament}' and
score2>score1))) as points from users,tournament_register where
users.id=tournament_register.player and
tournament_register.tournament='{ $tournament}' order by points desc, diff desc,
score1 desc, win desc
```

Obrázek 9: SQL dotaz pro výpočet údajů do ligové tabulky

Celkový počet zápasů se spočítá až v PHP. Poté se pomocí cyklu vykreslí ligová tabulka.

- **print_league_rounds_matches(\$tournament)**

Vypíše seznam jednotlivých kol. Po kliknutí na kolo se rozbalí zápasy daného kola. Vedle kola se může zobrazit otazník nebo vykřičník ve žlutém kolečku. Otazník signalizuje, soupeř již zadal výsledek zápasu nebo že hráč nestihl zadat výsledek do termínu. Vykřičník znamená, že soupeř zadal odlišný výsledek než daný hráč.
- **print_playoff_rounds_matches(\$tournament)**

Dělá to samé jako `print_league_rounds_matches($tournament)`, ale kvůli odlišné struktuře ligových a playoff zápasů nelze použít jedna funkce.
- **search(\$search)**

Kontroluje, zda hledaný řetězec neobsahuje nepovolené znaky a zda je alespoň 3 znaky dlouhý.
- **reset_password(\$user)**

Nastaví heslo uživatele na „1234“.
- **ban_user(\$user)**

Zakáže uživateli přihlášení do systému.
- **unban_user(\$user)**

Povolí uživateli přihlášení do systému.

10 DATABÁZE

Databáze systému obsahuje celkem 10 tabulek. Detailní popis jednotlivých tabulek je uveden dále, nyní budou popsány vztahy mezi nimi. První tabulkou je tabulka **users**. Její sloupec `id` slouží jako cizí klíč v tabulkách `users_teams` (`user`), `teams` (`admin`), `team_membership_requests` (`user`), `tournament_register` (`player`), `league_match` (`player1`, `player2`) a `playoff_match` (`player1`, `player2`). Další tabulka je **teams**, jejíž sloupec `id` je cizím klíčem v tabulkách `users_teams` (`team`), `team_membership_requests` (`team`), `tournament_register` (`player`), `league_match` (`player1`, `player2`) a `playoff_match` (`player1`, `player2`). Sloupec `id` tabulky **tournament** je cizím klíčem u tabulek `tournament_register` (`tournament`), `league_match` (`tournament`) a `playoff_match` (`tournament`). Poslední relace je mezi tabulkami **league_match** resp. **playoff_match**, kde sloupec `id` plní roli cizího klíče u tabulky **league_match_result** resp. **playoff_match_result**. Vše, co bylo výše slovně popsáno, ukazuje graficky následující schéma:

- Tabulka **users** obsahuje data o registrovaných uživateli. Uživatelské jméno i email musí být unikátní.

Tabulka 1: Tabulka users

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
username	varchar(20)	ne	unique
password	varchar(40)	ne	
level	unsigned tinyint	ne	
firstname	varchar(50)	ano	
lastname	varchar(50)	ano	
email	varchar(50)	ne	unique
valid	unsigned tinyint	ne	
ban	unsigned tinyint	ne	

id – číslo uživatele, které slouží jako cizí klíč pro některé další tabulky

username – uživatelské jméno, pod kterým se uživatel přihlašuje do systému

password – heslo v podobě sha1 hashe

level – úroveň oprávnění uživatele (1 – supervisor, 2 – administrátor, 3 – běžný uživatel)

firstname – křestní jméno

lastname – příjmení

email – email uživatele

valid – určuje, zda je uživatel ověřený, tzn. že potvrdil registrační email (0 - nepotvrzený, 1 - potvrzený)

ban – určuje, zda má uživatel znemožněno přihlášení (0 – smí se přihlásit, 1 – nemůže se přihlásit)

- Tabulka **users_teams** uchovává informace o členství uživatelů v týmech. Na sloupce **user** a **team** je aplikován index **unique**, čímž se zamezí možné duplicitě (tzn. hráč může být členem jednoho týmu pouze jednou).

Tabulka 2: Tabulka *users_teams*

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
user	unsigned int	ne	unique
team	unsigned int	ne	

id – číslo záznamu

user – id uživatele z tabulky *users*

team – id týmu z tabulky *teams*

- V tabulce **teams** jsou uložena data o týmech. Název týmu se nesmí v tabulce vyskytovat vícekrát.

Tabulka 3: Tabulka *teams*

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
name	varchar(50)	ne	unique
admin	unsigned int	ne	

id - číslo týmu, které slouží jako cizí klíč pro další tabulky

name – název týmu

admin – id uživatele z tabulky *users*, který je administrátorem týmu

- Tabulka **team_membership_requests** obsahuje žádosti uživatelů na členství v týmech. Index unique na sloupcích team a user brání vícenásobnému požadavku na jeden tým.

Tabulka 4: Tabulka team_membership_requests

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
team	unsigned int	ne	unique
user	unsigned int	ne	

id – číslo záznamu

team – id týmu z tabulky teams

user – id uživatele z tabulky users

- V tabulce **tournament** se nachází údaje o turnajích.

Tabulka 5: Tabulka tournament

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
name	varchar(50)	ne	
kind	enum('league', 'playoff')	ne	
type	enum('single', 'team')	ne	
n_players	unsigned tinyint	ne	
registration_begin	datetime	ne	
registration_end	datetime	ne	
private	unsigned tinyint	ne	
admin	unsigned tinyint	ne	
status	enum('founded', 'registering', 'starting1', 'starting2', 'running', 'finished', 'cancelled', 'refused')	ne	

id - číslo turnaje, které slouží jako cizí klíč pro další tabulky

name – název turnaje

kind – druh turnaje (liga / playoff), dle toho sloupce se určuje, zda se pro ukládání zápasů použije tabulka `league_match` nebo `playoff_match`

type – typ turnaje (jednotlivci / týmový)

n_players – maximální počet účastníků turnaje

registration_begin – začátek registrace do turnaje

registration_end – konec registrace do turnaje

private – určuje, zda bude turnaj soukromý nebo veřejný

admin – id uživatele z tabulky `users`, který je administrátorem turnaje

status – fáze turnaje

- Tabulka **tournament_register** ukládá registrace hráčů (uživatelů nebo týmů dle typu turnaje) do turnajů. Na sloupce `tournament` a `player` je aplikován index `unique`, což dovolí jednomu hráči být registrovaný do každého turnaje jen jednou.

Tabulka 6: Tabulka `tournament_register`

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
tournament	unsigned int	ne	unique
player	unsigned int	ne	

id – číslo záznamu

tournament – id turnaje

player – id uživatele nebo týmu, podle toho, o jaký typ turnaje se jedná

- Do tabulky **league_match** se ukládají ligové zápasy.

Tabulka 7: Tabulka league_match

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
tournament	unsigned int	ne	
round	unsigned tinyint	ne	
player1	unsigned int	ano	
player2	unsigned int	ano	
score1	unsigned smallint	ano	
score2	unsigned smallint	ano	
play_datetime	datetime	ne	
result_datetime	datetime	ne	

id – číslo zápasu, které slouží jako cizí klíč u tabulky league_match_result

tournament – id turnaje z tabulky tournament, ke kterému daný zápas patří

round – číslo ligového kola

player1 – id prvního hráče (uživatele / týmu)

player2 – id druhého hráče (uživatele / týmu)

score1 – skóre prvního hráče

score2 – skóre druhého hráče

play_datetime – termín odehrání zápasu

result_datetime – mezní termín zadání výsledku zápasu

- Tabulka **playoff_match** uchovává zápasy playoff turnajů.

Tabulka 8: Tabulka playoff_match

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
tournament	unsigned int	ne	
round	unsigned tinyint	ne	
subround	unsigned tinyint	ne	
grouping	unsigned tinyint	ne	
player1	unsigned int	ano	
player2	unsigned int	ano	
score1	unsigned smallint	ano	
score2	unsigned smallint	ano	
play_datetime	datetime	ne	
result_datetime	datetime	ne	
next_match	unsigned int	ano	
virtual	unsigned tinyint	ne	

id – číslo zápasu, které slouží jako cizí klíč u tabulky playoff_match_result

tournament – id turnaje z tabulky tournament

round – číslo kola

subround – číslo zápasu v daném kole

grouping – slouží pro seskupení zápasů daných hráčů v daném kole

player1 – id prvního hráče (uživatele / týmu)

player2 – id druhého hráče (uživatele / týmu)

score1 – skóre prvního hráče

score2 – skóre druhého hráče

play_datetime – termín odehrání zápasu

result_datetime – termín zadání výsledku zápasu

next_match – určuje, do kterého zápasu v dalším kole, postoupí vítěz

virtual – označuje virtuální zápas (0 – řádný zápas, 1 – virtuální zápas)

- Tabulka **league_match_result** je pomocnou tabulkou a uchovává výsledky zadané hráči zápasu.

Tabulka 9: Tabulka league_match_result

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
match	unsigned int	ne	
score_A	varchar(10)	ano	
score_B	varchar(10)	ano	

id – číslo záznamu

match – id zápasu z tabulky league_match

score_A – výsledek zadaný prvním hráčem

score_B – výsledek zadaný druhým hráčem

- Tabulka **playoff_match_result** je pomocnou tabulkou a uchovává výsledky zadané hráči zápasu.

Tabulka 10: Tabulka playoff_match_result

Název sloupce	Datový typ	Nulový	Další vlastnosti
id	unsigned int	ne	primary key, auto_increment
match	unsigned int	ne	
score_A	varchar(10)	ano	
score_B	varchar(10)	ano	

id – číslo záznamu

match – id zápasu z tabulky playoff_match

score_A – výsledek zadaný prvním hráčem

score_B – výsledek zadaný druhým hráčem

11 ZABEZPEČENÍ

Bezpečnost je jedna ze stěžejních vlastností každé webové aplikace. Proto je třeba jí věnovat patřičnou pozornost a vzít v potaz všechna možná rizika. Následující kapitola popisuje všechny prvky zabezpečení, které byly implementovány do turnajového systému.

11.1 Uživatelská práva

V systému jsou definovány tři úrovně oprávnění – supervizor, administrátor a běžný uživatel. Supervizor je hlavní správce systému a má neomezený přístup ke všem administracím (turnajů, týmů, uživatelů). Druhou úrovní je administrátor, který může přistupovat do administrací všech turnajů, ne však týmů a uživatelů. Nejnižší oprávnění má běžný uživatel, který má přístup pouze ke své administraci a administracím turnajů a týmů, které založil.

Jakýkoli pokus o přístup na stránku bez patřičných práv vede k zobrazení chybové stránky. Může se jednat např. o pokus uživatele o vstup do administrace turnaje, jehož není správce.

11.2 Ochrana formulářů proti podvržení dat

Ve většině formulářů jsou předávána data, která nesmí být měněna – např. id turnaje, id zápasu, id týmu atd. Pokud by se tyto údaje změnily, mohlo by dojít k manipulaci s daty, ke kterým nemá uživatel přístup. Proto je důležité mít mechanismus, který ověří, zda jsme z formuláře dostali data, která jsme do něj vygenerovali. Toho je docíleno tím, že do každého formuláře je přidáno skryté pole nazvané „security“, do kterého je uložen hash dat, která potřebujeme chránit, a hesla, které je uloženo v souboru inc/settings.php. Při zpracování formulářů se vygeneruje hash přijatých dat a hesla a pokud se shoduje s hashem z formuláře, je vše v pořádku. V opačném případě se zobrazí chybová stránka.

11.3 Ošetření uživatelských vstupů z formulářů

Veškeré uživatelské vstupy z formulářů jsou v příslušných funkcích kontrolovány na správnost. Textová pole jsou ošetřena pomocí regulárních výrazů, kde se kontrolují povolené znaky i formát dat. U číselných vstupů se ověřuje, zda se jedná opravdu o číslo pomocí PHP funkce `is_numeric()` a zda náleží do povoleného rozsahu. Data a časy taktéž prochází kontrolou. Nejdříve pomocí regulárních výrazů a v případě data PHP funkcí

checkdate(), v případě času ověřením na povolený rozsah. Stránka tournaments.php obsahuje filtr turnajů podle statutu. Při zpracování nejde vstup z filtru přímo do SQL dotazu, ale je předtím kontrolován pomocí příkazu switch(). Pokud tedy filtr nepatří do povoleného rozsahu, není akceptován.

Výše zmíněná opatření by měla zabránit tzv. SQL injection útoku, což je technika napadení databáze vsunutím kódu přes neošetřený vstup a vykonání vlastního SQL dotazu. Taktéž znemožní tzv. Cross-Site Scripting (XSS), kdy útočník dokáže do stránek podstrčit svůj vlastní javascriptový kód, což může využít buď k poškození vzhledu stránky, jejímu znefunkčnění anebo dokonce k získávání citlivých údajů návštěvníků stránek, obcházení bezpečnostních prvků aplikace a phishingu. [28]

11.4 Ochrana proti PHP injection

PHP injection (někdy též nazývaná nechráněná inkluze) je technika, při které útočník vloží na stránku vlastní skript (většinou záškodnický). Typický příklad je adresa `www.example.com/index.php?page=contact`, kterou lze upravit např. na `www.example.com/index.php?page=http://www.saboteur.com/hack.php`, čímž se skript `hack.php` vloží do stránky `www.example.com/index.php`. Toto je může přihodit, pokud není vkládání stránek nijak ošetřené. Chránit se lze více způsoby, ovšem nejspolehlivější metoda je použití seznamu povolených stránek – tzv. whitelist. Ta byla využita i v tomto systému. V souboru `left_column.php`, kam se pomocí `include` vkládají stránky, je pole, které obsahuje seznam povolených stránek. Pokud se požadovaná stránka v seznamu nachází, je pomocí `include` vložena, jinak je zobrazena chybová stránka.

11.5 Ochrana proti Session steal

Session steal, neboli ukradnutí session, spočívá v získání identifikátoru session a následné zneužití s cílem přimět cílový systém, aby považoval útočníka za autorizovaného. Tomu se dá do velké míry zabránit kontrolou IP adresy, čehož je využito i zde. Při přihlášení uživatele se do session uloží jeho IP adresa a ta je při každém požadavku porovnávána s adresou klienta. Pokud se neshodují, je uživatel vyzván, aby se znovu přihlásil.

11.6 Ochrana PHP skriptů pomocí .htaccess

V adresářích `inc/` a `pages/`, kde se nachází soubory `.php`, jsou umístěny soubory `.htaccess`, které omezují přímý přístup do těchto složek. Toho je dosaženo příkazem „deny from all“, který všem zamezí přístup zvenčí, do složek se tím pádem dostanou pouze požadavky pocházející ze serveru, kde běží naše aplikace.

11.7 Obrana proti nežádoucímu vícenásobnému odeslání formuláře

Při procházení stránek za pomoci tlačítek Vpřed a Zpět se snadno může stát, že se uživatel vrátí na stránku, kde odesílal nějaký formulář. V tom případě prohlížeč zobrazí výzvu, že je nutné provést reload stránky. Tím se ovšem znovu odešle i onen formulář, což je v drtivé většině případů nežádoucí a proto je třeba tomu zamezit. Řešení na straně klienta jsou nedostatečná, proto bylo zvoleno řešení kontroly na serveru. Princip je v zásadě hodně podobný zabezpečení formulářů, který byl popsán výše. Při každém zobrazení stránky je po zachycení a zpracování formulářové akce vygenerován hash z aktuálního času (microtime) a bezpečnostního hesla, tím pádem je hash pokaždé jiný. Hash se uloží do session a vloží do každého formuláře na stránce. Při odeslání se kontroluje, zda se hash z formuláře shoduje s hashem uloženým v session. Je-li tomu tak, je formulář zpracován, v opačném případě je jasné, že formulář již byl jednou odeslán a je zobrazena chybová stránka.

12 KOMPATIBILITA

Pro každou webovou aplikaci, která chce být úspěšná a využívaná, je důležité, aby se správně zobrazila pokud možno všem uživatelům. Jelikož podpora webových standardů se v různých prohlížečích, dokonce i odlišných verzích téhož prohlížeče, liší, je mnohdy složité vytvořit stránku, která bude ve všech prohlížečích vypadat stejně. Jednou z možností je využít tzv. CSS hacky pro každý prohlížeč, nicméně toto řešení není úplně ideální. Stránky se obtížněji udržují a kód je méně přehledný. Mnohem lepší je vytvořit validní stránky, i když ani v tomto případě nemáme jistotu jednotného vzhledu, protože jak bylo řečeno výše, každý prohlížeč si kód interpretuje po svém. Nejvíce problémů je tomto ohledu s Internet Explorerem, u kterého si autoři nikdy příliš nedělali hlavu ze standardů. Nicméně validita webu do určité míry vypovídá o úrovni autora webu, kód je čistý a vyhneme se tak problémům do budoucna. Navíc v posledních verzích nejrozšířenějších prohlížečů se implementace standardů sjednocuje, což je o důvod víc dodržovat při tvorbě webu daná pravidla.

12.1 Kompatibilita v různých prohlížečích

Při tvorbě turnajového systému byl stanoven požadavek, že musí být validní dle standardu XHTML 1.0 Transitional. Toho se podařilo dosáhnout a stránky úspěšně prošly validátorem W3C. Navíc díky tomu, že nebyly použity žádné CSS hacky pro prohlížeče, jsou validní i kaskádové styly a to dle verze CSS 2.1. Vznikl tak dobrý předpoklad toho, že stránky budou vypadat stejně ve všech prohlížečích.

Testování proběhlo v Internet Exploreru 6 a 8, Firefoxu 3 a Opeře 9.6. Dle očekávání vypadají stránky na všech platformách stejně, jediná drobná odlišnost je u formulářového prvku <select>, který je obecně považován za nejproblémovější, protože prohlížeče různě počítají jeho šířku. Dalo by se to obejít pomocí CSS hacků, nicméně kaskádové styly by již nebyly validní, proto od toho bylo upuštěno.

12.2 Kompatibilita na straně serveru

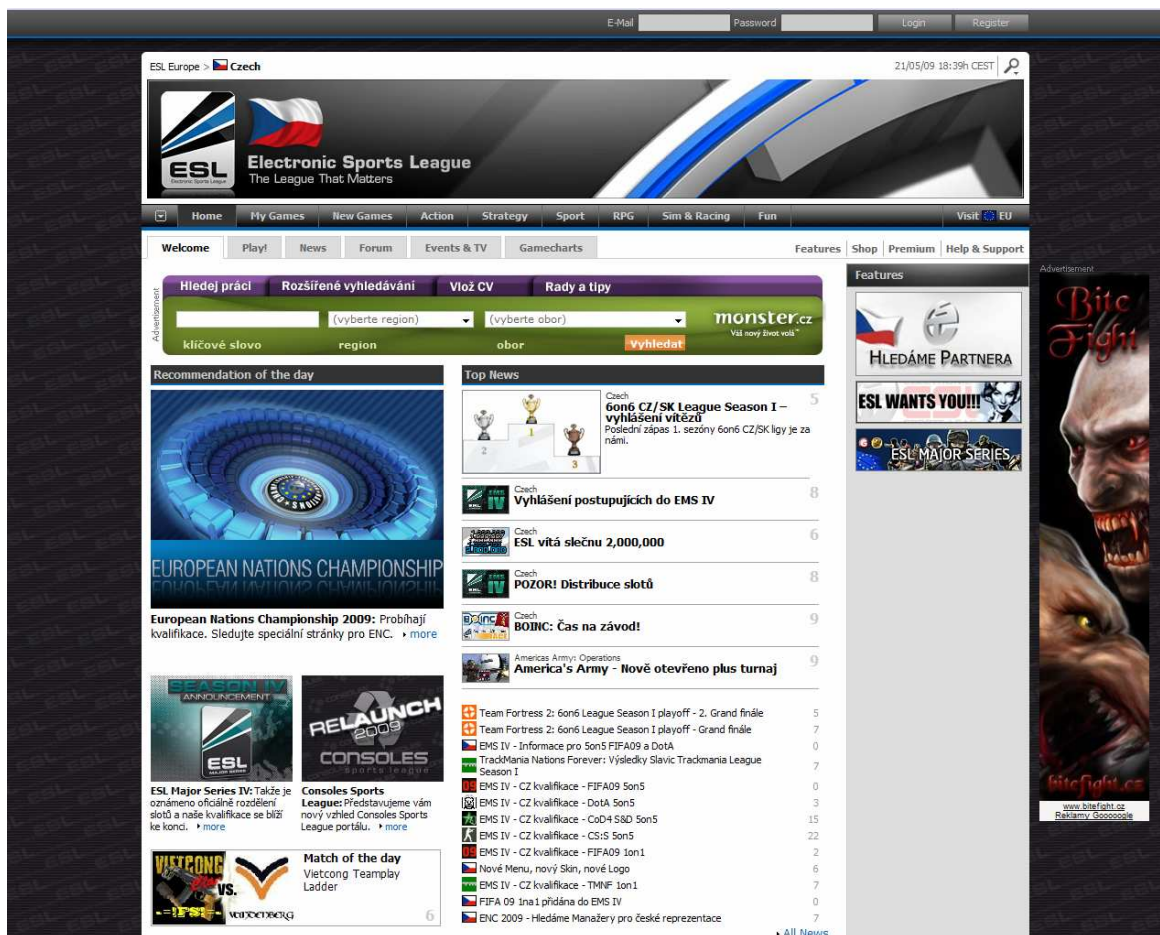
Vývoj a ladění systému probíhalo za pomoci tzv. all-in-one balíku (obsahuje HTTP server, PHP a MySQL) Vertrigo. Použitá verze obsahovala server Apache 2.0.63, PHP 5.2.6 a MySQL 5.0.51b. Systém nevyžaduje žádná speciální rozšíření na straně serveru, takže by

neměl být problém s provozem i na jiných verzích, nicméně je třeba mít správně nastavené jisté direktivy. V první řadě je třeba nastavit v konfiguraci PHP parametr `register_global` na `off`. Jelikož systém používá pro předávání session cookies a ne URL, je nutné nastavit i toto s pomocí `session.use_only_cookies` `on` a `session.use_trans_sid` `off`. O správné nastavení těchto direktiv se stará soubor `inc/settings.php`. Dále je nutné, aby server podporoval `mod_rewrite` a editaci souboru `.htaccess`, což je nutné pro SEO Cool URL. Žádné další požadavky nejsou kladeny.

13 POROVNÁNÍ S KONKURENČNÍMI SYSTÉMY

Ve světě i u nás již samozřejmě existují zaběhlé online turnajové systémy. Ze světových patří mezi nejznámější například ClanBase, GGL nebo ESL, u nás je praktický jediný server tohoto druhu, a to GamePark. Společným rysem všech výše zmíněných je to, že jsou určeny a optimalizovány pouze pro online počítačové (popř. konzolové) hry. Existuje i několik open source projektů (např. Open Tournament System), nicméně většinou mají velmi omezené funkce a jsou obtížně konfigurovatelné (výše zmíněný OTS kupříkladu vyžaduje pro svou funkci nainstalované phpBB fórum).

13.1 Electronic Sports League

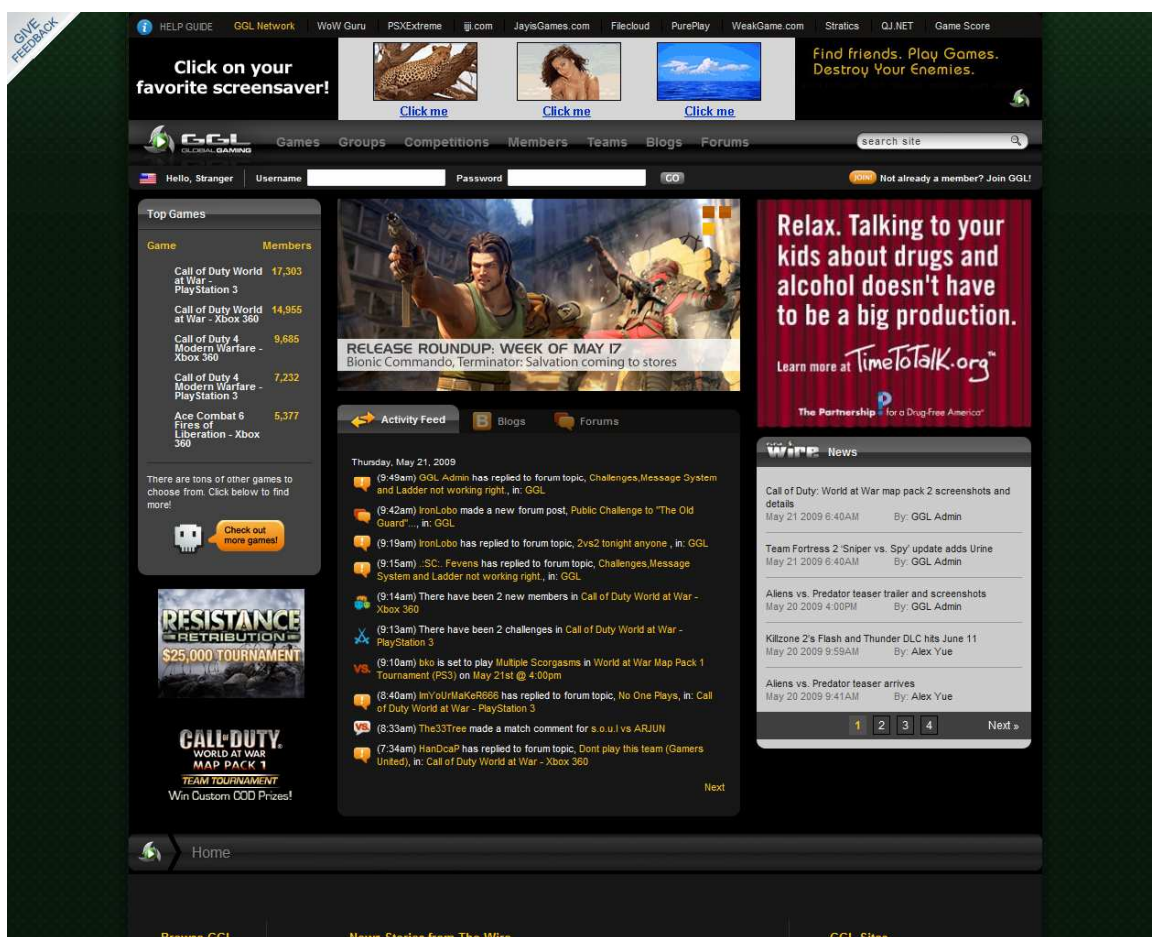


Obrázek 11: Hlavní strana ESL.eu

Electronic Sports League (www.esl.eu) byl založen v roce 1997 německou společností Turtle Entertainment. Nyní má více jak 2000000 zaregistrovaných uživatelů a z toho více jak 960000 aktivních hráčů a více jak 7000 zápasů za jeden den. Je považována za největší

evropskou online počítačovou herní ligu. Ligy jsou dělené do tří základních vrstev. Profesionální soutěže EPS - ESL Pro Series, dále EAS – ESL Amateur Series ligy a do žebříčkového systému. Za poslední roky se podařilo ESL expandovat úspěšně do USA i Číny, pro tuzemské hráče je výhodou, že existuje i česká mutace. Portál nabízí propracovaný systém odměn a motivací pro hráče, navíc s Premium Membership hráč získá spoustu výhod oproti běžným hráčům, např. aktivní podporu ligy, účast ve speciálních ligách/turnajích, např. ESL Amateur Series, plánovač, žádné reklamní bannery, rozšířené statistiky, systém krátkých zpráv, oznámení o zápasech přes ICQ a spoustu dalších.

13.2 ClanBase a Global Gaming League



Obrázek 12: Hlavní strana GGL.com

ClanBase (www.clanbase.com), původně holandský portál založený v roce 1998, si postupem času získal značnou prestiž a v letech 2000-2002 se jeho EuroCupů účastnili

nejlepší evropské hráči a týmy. V roce 2004 začal server ClanBase spolupracovat s americkým portálem Global Gaming League (www.ggl.com) a postupem času se s ním spojil. Nabízí podobné služby jako výše popsany portál ESL a za zmínku stojí, že je vyvíjen samotnými hráči zadarmo.

13.3 GamePark

Obrázek 13: Hlavní strana GamePark.cz

Herní server GamePark (www.gamepark.cz) běží od února 2006. Jedná o portál o počítačových hrách, takže zde najdeme novinky z tohoto odvětví, recenze, články o hardware atd. Nabízí turnaje a ligy všech aktuálních online her, od strategií přes střílečky až po závodní hry. Pro hraní je nutná registrace a stažení GP klienta, přes kterého se přihlašuje do her. Kromě obligátních sponzorovaných turnajů a lig nabízí tzv. Gbody, které uživatel získává za aktivitu a lze za ně získat různé věci v aukcích nebo je jen sbírat a umístit se tak ve statistikách. Dále nabízí Premium účet, se kterým získá uživatel možnost

různých více či méně užitečných funkcí, jako například avatary, skinny, medaile, seznam kamarádů atd.

13.4 Srovnání s naším systémem

Náš systém se samozřejmě co se propracovanosti a rozsahu týče nemůže výše zmíněným portálům rovnat – vždyť ty jsou většinou vyvíjeny týmy lidí po mnoho let. Cílem spíše bylo zaměřit se na jejich slabé stránky a nedokonalosti a zaplnit tak pomyslné místo na trhu. Jak již bylo zmíněno, většina portálů je zaměřena pouze na online hry. Proto byla snaha vytvořit maximálně univerzální turnajový systém použitelný i pro jiné sporty a hlavně nabídnout jej všem. Z tohoto důvodu má každý uživatel možnost založit si svůj vlastní turnaj, ať už se jedná o pingpongovou ligu nebo turnaj v kuličkách. Kromě toho existují samozřejmě i hlavní turnaje, které zakládají administrátoři a jsou tak vlastně obdobou turnajů na zmíněných serverech.

Systém byl otestován několika uživateli a co do funkčnosti nebyly zaznamenány žádné problémy. Hlavním nedostatkem je tak především propracovanost uživatelského prostředí a věcí okolo, což logicky nabízí úvahu na téma možného rozšíření do budoucna.

13.4.1 Návrhy rozšíření

Co se týká formátu turnajů, bylo by vhodné přidat tzv. double-elimination playoff, kdy poražený nevypadává hned z turnaje, ale zařadí se do spodní větve turnajového pavouka a i nadále má šanci bojovat o vítězství. Další možností je turnaj se základními skupinami a následnou vyřazovací částí, jak to známe například z MS v hokeji.

Pro veřejný provoz by bylo nutné zapracovat na uživatelském prostředí a různých „vychytávkách“. Jednou ze zásadních věcí je fórum, diskuse k turnajům a zápasům a systém pro zasílání zpráv mezi uživateli. Dále by bylo vhodné rozšířit profil uživatele o další údaje, dát možnost vložení loga, chybí také zaslání hesla v případě ztráty. Další věcí, která by neměla chybět, je systém bodů za aktivitu a získávání medailí za účast a umístění v turnajích, psaní do fóra atd., od čehož by se odvíjela úroveň uživatele (rank).

ZÁVĚR

Hlavním cílem diplomové práce bylo vytvořit turnajový systém dle zadaných požadavků. Jedním z hlavních byla univerzálnost – systém je primárně určen pro počítačové hry, ale lze jej využít i pro jakýkoli jiný sport nebo soutěž, která má charakter ligy nebo vyřazovacího turnaje. Další vlastností je dostupnost pro každého. Registrovaní uživatelé mohou vytvářet své vlastní turnaje – stačí pár kliknutí a můžete pořádat například vlastní školní ligu v kopané nebo třeba turnaj v šipkách.

Text diplomové práce je rozdělen do dvou částí – teoretické a praktické. V teoretické části jsou zmapovány hlavní technologie, které slouží pro tvorbu a provoz webu. Jedná se o WWW a HTTP, které tvoří hlavní provozní pilíře celého webu. Serverovou část obstarává Apache s PHP ve spojení s databází MySQL a ty zajišťují výstup pro klienta ve formě (X)HTML dokumentu naformátovaného pomocí CSS. Poslední kapitola pojednává o optimalizaci pro vyhledávače – tzv. SEO.

Praktická část se zabývá popisem vytvořeného systému. První kapitola je věnována ovládní a rozboru funkcionality. Následuje popis adresářové struktury a významu jednotlivých souborů. Samostatná kapitola je určena souboru functions.php, který obsahuje definice všech funkcí, které v podstatě tvoří jádro systému. Další část popisuje návrh a strukturu databáze, vazby mezi tabulkami a samozřejmě také jednotlivé tabulky. Důležitým aspektem je zabezpečení aplikace, kterému je věnována samostatná kapitola a kde jsou rozebrány bezpečnostní mechanismy, které jsou v systému obsaženy. V závěru praktické části je srovnání systému s možnými konkurenty, se kterými se zatím naprogramovaný systém nemůže rovnat, především co se týče uživatelských záležitostí.

Všechny body zadání se podařilo splnit, čímž byl dosažen cíl diplomové práce.

CONCLUSION

The main objective of thesis was to create a tournament system according to the set requirements. One of the main was universality - the system is primarily designed for computer games, but it can be used for any other sport or competition, which has the character or league or elimination tournament. Another feature is availability to everyone. Registered users can create their own tournaments - just a few clicks and you can organize your own school football league or tournament in darts.

Text thesis is divided into two parts - theoretical and practical. In the theoretical part there are mapped main technologies used for development and maintenance of the web. This is the WWW and HTTP, which form the main pillars of the Web. Server part provides Apache with PHP in conjunction with the MySQL database and prepares output for the client in the form of (X)HTML document formatted using CSS. The last chapter deals with the Search Engine Optimization - the SEO.

The practical part deals with the description of the system. The first chapter is devoted to analysis of functionality and control. There is a description of directory structure and the importance of individual files. A separate chapter is intended to functions.php file, which contains definitions of all functions, which essentially forms the core of the system. Next section describes the design and structure of the database, the links between tables as well as individual tables. An important aspect is the security, which is devoted a separate chapter to, and where they are dismembered security mechanisms, which are contained in the system. In conclusion of the practical part the system is compared with potential competitors, which currently programmed system can not equal to, especially in terms of user issues.

All items have managed to enter the meet and target of the thesis was achieved.

SEZNAM POUŽITÉ LITERATURY

- [1] Michael Kotlet, Bernd Öggl, PHP 5 a MySQL 5 – Průvodce webového programátora, Computer Press, říjen 2007, ISBN: 978-80-251-1813-9
- [2] Elizabeth Astro, HTML, XHTML a CSS – Názorný průvodce tvorbou WWW stránek, Computer Press, duben 2007, ISBN: 978-80-251-1531-2
- [3] Luboslav Lacko, Ajax – Hotová řešení, Computer Press, květen 2008, ISBN: 978-80-251-2108-5
- [4] Optimalizace pro vyhledávače - SEO [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://seo.jasminka.cz>>.
- [5] World Wide Web - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/World_Wide_Web>.
- [6] World Wide Web Consortium - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/World_Wide_Web_Consortium>.
- [7] Hypertext Transfer Protocol - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol>.
- [8] HTTPS - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/HTTPS>>.
- [9] PHP - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/PHP>>.
- [10] Historie a současnost PHP – Knihovna PHP – Interval.cz [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <<http://php.interval.cz/clanky/historie-a-soucasnost-php/>>.
- [11] Apache HTTP Server - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Apache_HTTP_Server>.

- [12] MySQL - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-15].
Dostupný z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [13] HyperText Markup Language - Wikipedie, otevřená encyklopedie [online]. 2009
[cit. 2009-05-14]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/HyperText_Markup_Language>.
- [14] Extensible HyperText Markup Language - Wikipedie, otevřená encyklopedie
[online]. 2009 [cit. 2009-05-14]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language>.
- [15] Cascading Style Sheets - Wikipedie, otevřená encyklopedie [online]. 2009
[cit. 2009-05-14]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Cascading_Style_Sheets>.
- [16] SEO – optimalizace pro vyhledávače Ataxo [online]. 2009 [cit. 2009-05-11].
Dostupný z WWW: <<http://www.ataxo.cz/sluzby/seo-optimalizace/>>.
- [17] Tvorba www stránek - tvorba webových stránek - profesionální webdesign
[online]. 2009 [cit. 2009-05-13]. Dostupný z WWW:
<<http://www.prezentacni.info/tvorba-www-SEO-optimalizace-vyhledatelnost.html>>.
- [18] Search Engine Optimization - Wikipedie, otevřená encyklopedie [online]. 2009
[cit. 2009-05-12]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Search_Engine_Optimization>.
- [19] Symbio Internetová agentura [online]. 2009 [cit. 2009-05-13]. Dostupný z WWW:
<<http://www.symbio.cz/clanky/budovani-zpetnych-odkazu.html>>.
- [20] GRAVEGAME - programování grafika. PHP a HTML scripty a návody [online].
2009 [cit. 2009-05-14]. Dostupný z WWW:
<<http://sluzby.gravegame.net/seozak.php>>.
- [21] Pěkné SEO - seo.pekne.cz [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW:
<<http://seo.pekne.cz/seo/zakazane-seo-techniky/>>.
- [22] PHPMailer [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW:
<<http://phpmailer.codeworxtech.com>>.

- [23] TinyMCE [online]. 2009 [cit. 2009-05-12]. Dostupný z WWW: <<http://tinymce.moxiecode.com/index.php>>.
- [24] Sothink DHTML Menu [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.sothink.com/product/dhtmlmenu/index.htm>>.
- [25] PHP triky – Weblog o elegantním programování v PHP pro mírně pokročilé [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://php.vrana.cz>>.
- [26] Generování rozpisu zápasů v systému každý s každým [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.tomkot.cz/clanky/generovani-rozpisu-zapasu-v-systemu-kazdy-s-kazdym/>>.
- [27] OHF – Pohled pod led [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.ohf.cz/pohled/losovani.html>>.
- [28] Cross-site scripting - Wikipedie, otevřená encyklopedie [online]. 2009 [cit. 2009-05-14]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Cross-site_scripting>.
- [29] PHP: Hypertext Preprocessor [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.php.net>>.
- [30] MySQL: The world's most popular open source database [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.mysql.com>>.
- [31] The Apache HTTP Server Project [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://httpd.apache.org>>.
- [32] Interval.cz [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://interval.cz>>.
- [33] Jak psát web, návod na HTML stránky [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.jakpsatweb.cz>>.
- [34] World Wide Web Consortium – Web Standards [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.w3c.org>>.
- [35] SEO nástroje - analýza www stránek a vyhledávačů - SEO Servis [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://seo-servis.cz>>.

[36]ESL – Electronic Sports League [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.esl.eu/cz/>>.

[37]ClanBase [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://clanbase.ggl.com>>.

[38]GamePark [online]. 2009 [cit. 2009-05-08]. Dostupný z WWW: <<http://www.gamepark.cz>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheet
DTD	Document Type Definition
IP	Internet Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
PHP	PHP: HyperText Preprocessor
SEO	Search Engine Optimization
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WYSIWYG	What You See Is What You Get
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language
XSS	Cross-Site Scripting

SEZNAM OBRÁZKŮ

Obrázek 1: Příklad selektoru.....	29
Obrázek 2: Příklad třídy.....	29
Obrázek 3: Příklad identifikátoru	30
Obrázek 4: Hlavní stránka systému	37
Obrázek 5: Rozložení hráčů pro rozlosování zápasů.....	47
Obrázek 6: Losování ligových kol.....	48
Obrázek 7: Postup generování zápasů playoff.....	49
Obrázek 8: Vykreslení turnajového pavouka pomocí tabulky.....	52
Obrázek 9: SQL dotaz pro výpočet údajů do ligové tabulky	54
Obrázek 10: Schéma databáze	57
Obrázek 12: Hlavní strana ESL.eu.....	70
Obrázek 11: Hlavní strana GGL.com	71
Obrázek 13: Hlavní strana GamePark.cz	72

SEZNAM TABULEK

Tabulka 1: Tabulka users	58
Tabulka 2: Tabulka users_teams.....	59
Tabulka 3: Tabulka teams.....	59
Tabulka 4: Tabulka team_membership_requests.....	60
Tabulka 5: Tabulka tournament	60
Tabulka 6: Tabulka tournament_register	61
Tabulka 7: Tabulka league_match	62
Tabulka 8: Tabulka playoff_match	63
Tabulka 9: Tabulka league_match_result	64
Tabulka 10: Tabulka plyoff_match_result.....	64