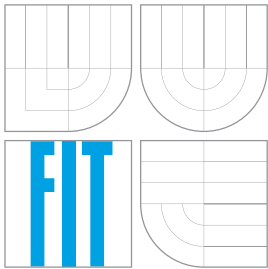


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM SPORTOVNÍHO KLUBU

INFORMATION SYSTEM OF SPORT CLUB

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN PAVELKA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAEL KUNC

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Pavelka Jan**

Obor: Informační technologie

Téma: **Informační systém sportovního klubu**

Kategorie: Databáze

Pokyny:

1. Seznamte se s technologiemi pro vývoj webových aplikací a informačních systémů.
2. Zjistěte požadavky na informační systém Nohejbalového oddílu TJ Spartak MSEV Přerov a dalších nohejbalových oddílů.
3. Na základě zjištěných požadavků z bodu 2 navrhnete univerzální redakční systém použitelný pro nohejbalové sportovní kluby s různými úrovněmi oprávnění (administrátor, člen klubu, návštěvník webu).
4. Informační systém realizujte. Otestujte ho na nohejbalovém oddílu uvedeném v bodě 2 a na alespoň jednom dalším oddílu.
5. Zhodnoťte dosažené výsledky a diskutujte další možné pokračování tohoto projektu.

Literatura:

- Stejskal, J.: Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Computer Press, 2004.
- Gilmore, W. J.: Velká kniha PHP5 & MySQL. Zoner Press, 2005.
- Ullman, L.: PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. Computer Press, 2004.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

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

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

Vedoucí: **Kunc Michael, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

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

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jan Pavelka**
Id studenta: 84212
Bytem: Chotucká 240, 289 33 Křinec
Narozen: 28. 12. 1984, Přerov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

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

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

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

Název VŠKP: Informační systém sportovního klubu
Vedoucí/školitel VŠKP: Kunc Michael, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

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

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

Článek 2 Udělení licenčního oprávnění

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

Článek 3 Závěrečná ustanovení

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

Brně dne:

.....

Nabyvatel


.....
Autor

Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat webový informační systém nohejbalového klubu. Práce zahrnuje seznámení se s technologiemi pro vývoj webových aplikací a informačních systémů, popis problematiky vývoje, analýzu a specifikaci požadavků a popis implementace informačního systému. Systém je implementován za použití technologií PHP, JavaScript, XHTML, CSS a MySQL.

Klíčová slova

PHP, MySQL, XHTML, CSS, informační systém, web, databáze, nohejbal

Abstract

The goal of this bachelor's thesis is to create and to implement web information system of footballtennis club. Thesis includes introduction with overview of technologies for development of web applications and information systems, description of problems in development, analysis and specifications of requirements and description of implementation of information system. System is implemented with use of technologies PHP, JavaScript, XHTML, CSS and MySQL.

Keywords

PHP, MySQL, XHTML, CSS, information system, web, database, footballtennis

Citace

Jan Pavelka: Informační systém sportovního klubu, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém sportovního klubu

Prohlášení

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

.....

Jan Pavelka

14. dubna 2007

Poděkování

Děkuji tímto vedoucímu své bakalářské práce, panu Ing. Michaelu Kuncovi, za vedení mého projektu, užitečné rady a podnětné připomínky.

© Jan Pavelka, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	5
2	Analýza požadavků	6
2.1	Definice pojmů	6
2.1.1	Informace	6
2.1.2	System	7
2.1.3	Informační systém	7
2.1.4	Webový informační systém	8
2.1.5	Databáze	8
2.2	Vymezení cílů, obecné požadavky	9
2.2.1	Použitelnost	9
2.2.2	Využitelnost	9
2.2.3	Přístupnost	10
2.2.4	Vzhled	10
2.2.5	Zabezpečení	10
2.2.6	Rozšířitelnost	10
3	Specifikace požadavků	11
3.1	Specifikace obecných požadavků	11
3.2	Specifikace požadavků podle typu uživatele	12
4	Návrh řešení	15
4.1	Architektura	15
4.2	Výběr technologií	16
4.2.1	Technologie současného webdesignu	16
4.2.2	Technologie pro vývoj dynamických webových stránek	19
4.2.3	Databáze	22
4.2.4	Webový server	24
4.2.5	Kódování	24
4.3	Návrh vzhledu, layout	26
4.4	Způsob zabezpečení	27
4.5	Diagram případů užití	28

4.6 ER Diagram	29
5 Implementace	32
5.1 Adresářová struktura	32
5.2 Popis hlavních skriptů a souborů	33
5.3 Popis implementace některých částí	35
6 Testování	38
7 Závěr	39
Seznam použitých zdrojů	40
Seznam použitých zkratk a symbolů	41
Seznam příloh	42
A Ukázka vzhledu informačního systému	43
B Návod na instalaci systému	44
C Obsah příloženého CD	45

Seznam obrázků

4.1	Architektura klient-server	15
4.2	Grafické znázornění technologie DHTML	19
4.3	Layout stránek informačního systému, rozmístění komponent	26
4.4	Diagram případů užití	30
4.5	ER Diagram	31
A.1	Ukázka vzhledu informačního systému	43

Seznam tabulek

3.1	Přehled typů uživatelů informačního systému a jejich práv, 1. část	13
3.2	Přehled typů uživatelů informačního systému a jejich práv, 2. část	14

Kapitola 1

Úvod

Rozmach internetu udivuje lidstvo den co den. Tato revoluce v informatice nabývá enormních rozměrů. Meze internetového rozvoje jsou zcela v nedohlednu. Jsme svědky vzniku neuvěřitelně silného nástroje pro komunikaci, zábavu, obchod a informace. Internet je bezpochyby jedním z největších vynálezů 20. století. Také obrovský nárůst uživatelů internetu po celém světě vede lidi, i vše co společně vytváří, k potřebě se prezentovat a zviditelnit, jednoduše být součástí tohoto obrovského informačního zdroje.

Tato potřeba se zviditelnit snad ještě více platí v oblasti sportu. Od pradávna byl sport pro lidi formou zábavy, a to nejen pro zúčastněné, ale zejména pro přihlížející. Každý sport pro svůj vývoj potřebuje popularitu, reklamu, sledovanost a také podávat lidem pokud možno co největší množství informací. Nejinak je tomu u nohejbalu.

Tento poměrně mladý, ryze český resp. československý sport zažívá v posledních letech veliký rozmach. Stále více lidí začíná hrát nohejbal na amatérské úrovni, každým rokem přibývají týmy profesionální, zvyšuje se počet dlouhodobých i krátkodobých profesionálních soutěží i mezinárodních akcí. Rovněž ve světě má nohejbal velké zastoupení. Z prvotní České republiky se rozšířil téměř do celé Evropy i za její hranice, do zemí jako např. USA nebo Brazílie. K tomuto rozmachu patří i rozmach mediální. Nohejbal je čím dál častěji vysílán v televizi, rádiu, pravidelné výsledky a informace vychází v tisku a rozvíjí se i ve světě internetu.

Účelem této bakalářské práce je napomoci rozvoji nohejbalu v oblasti informační vytvořením aplikace, která bude, co se týče oblasti působnosti, jedna z prvních svého druhu. Předmětem této bakalářské práce je vytvoření webového informačního systému obecného nohejbalového sportovního klubu. Pro testovací účely je použit nohejbalový klub TJ Spartak MSEM Přerov.

Jednotlivé kapitoly provedou čtenáře celým vývojem webového informačního systému, počínaje analýzou a specifikací požadavků, přes návrh jeho řešení, až po popis samotné implementace a testování vytvořené aplikace.

Kapitola 2

Analýza požadavků

2.1 Definice pojmů

2.1.1 Informace

Na pojem *informace* se lze dívat z řady různých pohledů. Definice informace podle České terminologické databáze knihovnictví a informační vědy [1] je následující:

V nejobecnějším slova smyslu se *informace* chápe jako údaj o reálném prostředí, o jeho stavu a procesech v něm probíhajících. Informace snižuje nebo odstraňuje neurčitost systému (např. příjemce informace); množství informace je dáno rozdílem mezi stavem neurčitosti systému (entropie), kterou měl systém před přijetím informace a stavem neurčitosti, která se přijetím informace odstranila. V tomto smyslu může být informace považována jak za vlastnost organizované hmoty vyjadřující její hloubkovou strukturu (varietu), tak za produkt poznání fixovaný ve znakové podobě v informačních nosičích. V informační vědě a knihovnictví se informací rozumí především sdělení, komunikovatelný poznatek, který má význam pro příjemce nebo údaj usnadňující volbu mezi alternativními rozhodovacími možnostmi. Významné pro informační vědu je také pojetí informace jako psychofyzilogického jevu a procesu, tedy jako součásti lidského vědomí (např. N. Wiener definuje informaci jako „obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním“). V exaktní vědě se např. za informaci považuje sdělení, které vyhovuje přísným kritériím logiky či příslušné vědy. V ekonomické vědě se informací rozumí sdělení, jehož výsledkem může být zisk nebo užitek. V oblasti výpočetní techniky se za informaci považuje kvantitativní vyjádření obsahu zprávy. Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami (0, 1) a vyjadřuje se jednotkou nazvanou bit.

2.1.2 Systém

Obecný *systém* je (podle [8]) účelově definovaný soubor komponent, mezi kterými existují určité vztahy, a které splňují nějaký cíl.

Systém se skládá z *atributů* (veličiny jež charakterizují určitý prvek systému), *událostí* (změna atributu nebo změna konfigurace systému – například komponenty) a *časových množin* (hodnoty vztažené k určitému okamžiku).

Hranice a okolí systému: Hranice systému vymezuje samotný systém nebo odděluje více systémů.

Logická hranice je pomyslnou hranicí a vymezuje podsystémy v rámci systému, ovšem *okolí systému* je již „viditelnou“ hranicí. Prvky vně hranice pak ovlivňují chování systému.

Rozdělení systémů: Systémy dělíme takto:

- Uzavřené × otevřené – podle toho, zda nastává interakce s okolím,
- deterministické × stochastické – tzn. jednoznačné nebo statistické chování,
- statické × dynamické – tzn. lineární nebo diferenciální (systém si pamatuje vnitřní stav),
- spojité × diskrétní – podle časových událostí (případně také kombinované).

Systémy obecně dělíme na *tvrdé* a *měkké*. Tvrdé systémy jsou spojovány s jedním specifickým (strukturovaným) problémem (většinou technické systémy), naopak v měkkých systémech vystupuje celá řada faktorů, jsou obecnější (člověk je aktivním prvkem systému = individualita ⇒ problém, protože každý má jiné znalosti a jinak uvažuje).

Řízení a zpětná vazba: V systémech může nastat zpětná vazba, kdy výstupní veličina opětovně ovlivňuje vstupní veličinu, a tedy i samotný systém. Každý takový systém má tedy tendence být nestabilní.

2.1.3 Informační systém

Informační systém (IS) je systém pro shromažďování, zpracování, transformaci a poskytování informací a dat nezávisle na jejich časovém a prostorovém rozptylu. Příkladem informačního systému může být např. kartotéka, telefonní seznam, kniha došlé pošty nebo účetnictví. V této práci jde o IS automatizovaný pomocí počítačů, avšak obecně tyto systémy mohou být i v papírové nebo jiné podobě.

Informací zde míníme sdělení, které odstraňuje nejistotu nebo nevědomost nebo ji lze také chápat jako data s nějakým přidaným významem (data + význam). *Daty* míníme jakékoli zaznamenané poznatky či fakta. Data tedy samy o sobě nenesou žádný význam. Z hlediska počítačů jsou to pouze hodnoty.

Zvláštním typem informace je *znalost*, jež představuje poznání informací nebo informací a dat založené na jejich pochopení a ověření. Se znalostmi lze pracovat jako s daty, jejich interpretace však může být o něco složitější, neboť téměř vždy představují více informací najednou.

2.1.4 Webový informační systém

Internet poskytuje pro informační systém skvělou platformu. Slovo „webový“ je odvozeno od zkratky WWW neboli *World Wide Web*, což je jedna ze služeb právě sítě internet. Webový informační systém je tedy IS fungující nad službou World Wide Web, která se dá definovat jako soustava propojených hypertextových dokumentů.

Hlavními prvky tvořící World Wide Web jsou:

- Soustava hypertextových případně jiných dokumentů a souborů,
- jazyk pro zpracování těchto dokumentů (HTML, XML, aj.),
- přenos v síti zajišťovaný internetovými protokoly (HTTP, FTP, aj.).

Pro zobrazování webových dokumentů (stránek) a pohyb v informačním prostoru WWW slouží klientský program obecně zvaný prohlížeč.

2.1.5 Databáze

Nedílnou součástí IS jsou i *databáze*. Databáze je určitá uspořádaná množina dat uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento systém se v české odborné literatuře nazývá *systém řízení báze dat* (SŘBD). Běžně se tedy označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i SŘBD.

Z hlediska způsobu ukládání dat a vazeb mezi nimi se databáze rozdělují na:

- Hierarchické
- Síťové
- Relační
- Objektové

Z nichž v současnosti nejpoužívanější je databáze *relační*.

Relační databáze

je založená na relačním modelu, v němž jsou data logicky uspořádána do relací, tj. výsledků kartézského součinu nad doménami neboli množinami údajů. Fyzickou reprezentací relace je dvojrozměrná tabulka, jejíž řádky obsahují kolekce údajů o sledovaném objektu (tzv. entity, n-tice, záznamy, věty) a ve sloupcích jsou vždy obsaženy údaje stejného typu (tzv. atributy, pole, položky). Tabulky a sloupce jsou identifikovány svými jmény, řádky jsou identifikovány primárním klíčem. Veškeré vztahy mezi daty jsou vyjádřeny rovněž pomocí relací; vztahy mezi souvisejícími záznamy se definují prostřednictvím stejných údajů v dvojici položek (cizí klíč). Množinový princip relačních

databází umožňuje realizovat při manipulaci s daty neprocedurální operace založené na relační algebře a relačním kalkulu – projekce (výběr sloupců), selekce (výběr řádků), spojení (spojení řádků z různých tabulek na základě společné položky). Tuto definice relační databáze uvádí ve své knize [11] J. Pokorný.

2.2 Vymezení cílů, obecné požadavky

Informační systém by měl zajišťovat dvě hlavní funkce.

Funkce první První funkcí je prezentace klubu okolnímu světu. Systém by měl podávat informace o dění v klubu, jeho členech, o výsledcích soutěží, nastávajících akcích, změnách, novinkách, o sponzorech i o klubu samotném. Měla by být také zajištěna zpětná vazba od návštěvníků formou možnosti vyjádření jejich názoru.

Funkce druhá Druhá funkce by měla plnit funkci redakčního systému. Týká se především vnitřního chodu klubu. Zahrnuje komunikaci jednotlivých členů klubu mezi sebou, zobrazování nařízení a postojů vedení klubu, provádění úkonů týkajících se správy klubu a chodu klubu, správu událostí a dokumentů. Tato funkce má také na starosti správu a fungování funkce první.

Obecné požadavky a cíle systému zahrnují několik hledisek:

- Použitelnost
- Využitelnost
- Přístupnost
- Vzhled
- Zabezpečení
- Rozšířitelnost

2.2.1 Použitelnost

Informační systém a jeho ovládání musí být přehledný a srozumitelný, nenutí uživatele přemýšlet nad věcmi, které by měl dělat automaticky, neboli intuitivně. Pokud navštíví uživatel systém poprvé, měl by snadno a rychle pochopit jeho používání. Naopak pokud je již se systémem obeznámen, měl by rychle a bez problémů dosahovat svých plánovaných cílů. Uspořádání a ovládání IS by mělo být lehce zapamatovatelné a obecně jeho používání by nemělo přivodit jakýkoliv negativní zážitek, pokud jde o samotný systém.

2.2.2 Využitelnost

Systém musí být univerzální, použitelný obecně pro různé nohejbalové kluby. Měl by také umožňovat zobrazení různých cizojazyčných znaků ve správném nezkresleném tvaru. Vhodné je také, aby fungoval na většině standardních webových serverech bez změny funkčnosti.

2.2.3 Přístupnost

Na toto hledisko by měl být kladen velký důraz. Systém by měl být přístupný co největšímu spektru webových uživatelů. Důvody, které mohou zapříčinit problémy ze strany uživatele jsou zejména:

- **technické** – Uživatelé by neměli být znevýhodněni jednak z hlediska softwaru, který k zobrazení stránek informačního systému používají (Internet Explorer, FireFox, Opera apod.) a jednak platformy, na které si stránky prohlížíjí. Při přístupu k informačnímu obsahu by návštěvníci neměli být znevýhodněni ani rychlostí svého připojení k internetu.
- **tělesné** – Zde se jedná především o problémy, které ovlivňují vizuální kontakt se systémem. Zde patří zrakové postižení, jako například slabozrakost nebo barvoslepost.

Na uvedené problémy by měl být brán zřetel a kladena snaha jich co nejvíce eliminovat. Čím širšímu okruhu lidí bude IS přístupný a bude jim poskytovat při prohlížení jeho obsahu určitý komfort, tím více lidí se bude na tento IS vracet.

2.2.4 Vzhled

Vzhled (nebo také design) vytváří první dojem uživatele, proto je důležitý pro prezentaci klubu veřejnosti v podobě návštěvníků. Vzhled IS by měl působit příjemně a decentně.

Systém by měl být koncipován tak, aby bylo možné vzhled IS měnit, aniž by byl ovlivněn obsah. Změna vzhledu by měla být snadno proveditelná. Jelikož jde o univerzální systém použitelný pro více klubů, možnost snadné modifikace zde nabývá významu ještě většího.

2.2.5 Zabezpečení

Celý IS by měl být dobře zabezpečen proti útokům různého typu z vnějšku a také pokud možno zevnitř systému. Zabezpečovací mechanismy by však měly co nejméně ovlivňovat práci běžného uživatele.

2.2.6 Rozšířitelnost

Vzhledem k univerzálnosti, obecnosti a s ohledem na využitelnost systému v různých klubech by měl být IS navržen tak, aby mohl dle požadavků a záměrů jednotlivých klubů být případně doplněn o další funkce a chování, které v něm nejsou obsaženy.

K rozšířitelnosti patří i nutnost možnosti aktualizace systému vzhledem k času, případným novým technologiím, změnám a zvyklostem.

Kapitola 3

Specifikace požadavků

3.1 Specifikace obecných požadavků

Rychlost načítání

Jak se můžeme dozvědět např. na internetovém portálu Jakpsatweb [10], zabývající se problematikou webových stránek, kritéria, která ovlivňují délku doby čekání na načtení stránek informačního systému a obecně každé internetové stránky jsou:

- Datová velikost stránek včetně obrázků
- Ukládání stránek na klientovi (tzv. cache)
- Struktura stránek (tabulky a pod.)
- Rychlost serveru
- Propustnost linky

Z těchto kritérií může samotná webová aplikace přímo ovlivnit pouze první tři.

Datovou velikost stránky ovlivňují zejména objekty jako obrázky, rámy, externí stylopisy (tj. CSS soubory), klientské skripty (externí JavaScript), applety a samozřejmě samotný text stránky. Je tedy žádoucí, aby celkový součet datových objemů byl co nejmenší.

Zrychlení stránek pomocí jejich **ukládání na klientovi** (nebo ukládání některých jejich objektů) se děje v případě opakovaného načtení stejného objektu. V takovém případě se objekt nestahuje ze serveru, nýbrž se použije objekt uložený na klientovi, což může v některých případech výrazně ovlivnit rychlost.

Pokud je stránka nevhodně **strukturovaná**, její text může během načítání různě poskakovat nebo se v horším případě nezobrazí vůbec, dokud se nenačte celý obsah. Tento problém bývá nejčastěji způsoben používáním velkých tabulek. Důvodem je zobrazení tabulky až poté, co se celá načte (v dnešní době se to týká pouze prohlížeče Internet Explorer, ovšem ten je stále nejpožívanějším prohlížečem). Také při neuvedení rozměrů obrázku musí prohlížeč přepočítat zobrazení, což má za následek pomalejší načtení stránky. Na tyto problémy je třeba brát ohled.

Validita stránek IS

Validitou webových stránek se rozumí jejich soulad s technickými pravidly pro psaní zvoleného značkovacího jazyka, v němž jsou vytvořeny. Tímto jazykem bývá nejčastěji HTML a jeho různé verze. Validita webových stránek hraje významnou roli v přístupnosti a použitelnosti webu.

Hlavní výhody validních stránek:

- mají výrazně lepší předpoklady zobrazit se správně většině uživatelů, včetně těch hendikepovaných či využívajících nestandardní prohlížeče či zařízení (mobilní telefony, PDA, méně rozšířené prohlížeče, alternativní operační systémy).
- jsou čitelné pro roboty vyhledávačů (Google, Seznam, ...), kteří porozumí jejich obsahu a zařadí je správně do své databáze. Budou tak snáze naležitelné pro větší počet potenciálních návštěvníků.
- jsou dobrou vizitkou nejen pro tvůrce, ale i vlastníka a provozovatele stránek. Svědčí o jeho smyslu pro kvalitu a zodpovědnost.
- jsou srozumitelné i návštěvníkům, kteří si vypnou obrázky či kaskádové styly, zobrazují se návštěvníkům rychleji, zaručují kompatibilitu i s budoucími verzemi prohlížečů, atd.

Je zřejmé, že validita přináší řadu výhod, a proto je vhodné, aby stránky IS byly validní.

Optimalizace pro internetové vyhledávače

Optimalizace pro vyhledávače, neboli *SEO* (Search Engine Optimization), je metodologie vytváření a upravování webových stránek takovým způsobem, aby byly ve výsledcích hledání v internetových vyhledávačích (např. Google, Seznam) zobrazeny na co nejlepších místech.

V knize P. Druska [9] je uvedeno 9 základních pravidel pro SEO. Jsou to například: použití vhodného titulu stránky, používání nadpisů v textu, pojmenování obrázků a odkazů, hustota klíčových slov, registrace stránky do vyhledávačů, používání zpětných odkazů, atd. Při vývoji IS by se mělo dbát na tyto pravidla.

Uživatelský vstup

Uživatelské vkládání a formátování textu musí zohledňovat uživatele bez hlubší znalosti jazyka, ve kterém jsou stránky napsány (standardně HTML).

Příhodné je použití tzv. WYSIWYG editoru (z anglického „what you see is what you get“, česky „co vidíš, to dostaneš“). Je to způsob editace textů, při kterém je verze textu zobrazená na obrazovce vzhledově totožná s verzí výslednou.

3.2 Specifikace požadavků podle typu uživatele

Informační systém rozlišuje celkem 7 typů uživatele. Každý typ má patřičná práva podle svého pole působení, přičemž obecně platí, že typ uživatele s vyššími právy přebírá práva typů uživatelů

s nižšími právy. Dále platí, že jednomu uživateli může být přiřazeno více typů najednou. Jednotlivé typy uživatelů (seřazeno vzestupně podle práv) jsou následující:

- **Návštěvník** – Nejfrekventovanější typ. Je to návštěvník IS, který není členem klubu. Vidí pouze ty části systému, které mají sloužit k prezentaci klubu okolnímu světu. Jeho aktivní zásah do systému může být pouze formou příspěvku v návštěvní knize, komentáře k aktualitě nebo hlasování v anketě. Nijak výrazně do systému nezasahuje.
- **Člen klubu s nižšími právy** – Uživatel tohoto typu je členem klubu, může se do IS přihlásit a mít přístup k redakčnímu systému, který tvoří vnitřní část IS. Má stejná práva jako návštěvník rozšířená o přístup k redakčnímu systému a základními úkony v něm, jako jsou přidávání příspěvků do fóra, nahrávání fotografií atd. viz tabulka 3.2.
- **Finančník** – Má stejná práva jako člen klubu s nižšími právy, avšak navíc spravuje záležitosti týkající se financí klubu. K tomu náleží i přístup do patřičných sekcí týkajících se financí.
- **Člen klubu s vyššími právy** – Narozdíl od člena klubu s nižšími právy se může více zapojovat do správy IS. Má právo zadávat aktuality, výsledky zápasů, zakládat ankety atd. viz tabulka 3.2. Nepřebírá ovšem práva finančníka, tudíž nemá přístup ke správě financí.
- **Trenér** – Od člena klubu s vyššími právy se liší možností přidávání, editace a mazání členů s nižšími právy. Také může zadávat speciální trenérské komentáře k zápasům.
- **Vedení** – Přejímá práva od člena klubu s vyššími právy a navíc může přidávat, editovat nebo mazat všechny typy členů
- **Administrátor** – Plní funkci správce celého systému. Ma přístup ke všem sekcím a přebírá všechna práva.

Typ uživatele	Práva	Přístupné sekce
Návštěvník	<ul style="list-style-type: none"> - prohlížení jednotlivých sekcí menu - přehled o nastávajících akcích - přidávání příspěvků do návštěvní knihy - přidávání komentářů k jednotl. aktualitám - hlasovat v anketě - možnost vidět, kdo je právě online 	aktuality, návštěvní kniha, fotogalerie, kontakt, historie klubu, využití kurtů, sponzoři, odkazy, výsledky, členská základna, události, anketa

Tabulka 3.1: Přehled typů uživatelů informačního systému a jejich práv, 1. část

Typ uživatele	Práva	Přístupné sekce
Člen klubu s nižšími právy	<ul style="list-style-type: none"> - vše co návštěvník - přidávání příspěvků do fóra - přidávání akcí do kalendáře - přidávání komentářů k jednotliv. aktualitám - přidávání komentářů k jednotliv. akcím - upload souborů - mazání svých příspěvků - zadávání nepřítomnosti - editace osobních údajů, změna hesla - zasílání, přijímání osobních zpráv - vytváření fotogalerií 	<p>jako návštěvník +</p> <p>fórum, upload souborů, osobní zprávy, kalendář</p>
Finančník	<ul style="list-style-type: none"> - vše co člen klubu s nižšími právy - zadávání, editace členských příspěvků - zadávání, editace cestovních dokladů - zadávání, editace stravného - zadávání, editace jiných výdajů - správa rozpočtu 	<p>jako člen klubu s nižšími právy +</p> <p>členské příspěvky, cestovní doklady, stravné, jiné výdaje, rozpočet</p>
Člen klubu s vyššími právy	<ul style="list-style-type: none"> - vše co člen klubu s nižšími právy - přidávání novinek - zadávání, editace výsledků zápasů - zadávání nových anket - mazání příspěvků 	<p>jako člen klubu s nižšími právy</p>
Trenér	<ul style="list-style-type: none"> - vše co člen klubu s vyššími právy - přidávání, editace, mazání členů s nižšími právy - zadávání, editace výsledků zápasů - zadávání trenérských komentářů 	<p>jako člen klubu s vyššími právy</p>
Vedení	<ul style="list-style-type: none"> - vše co člen klubu s vyššími právy - přidávání, editace, mazání finančníka, trenérů 	<p>jako člen klubu s vyššími právy</p>
Administrátor	<ul style="list-style-type: none"> - vše co všechny typy členů - přidávání, editace, mazání vedení - změna hesel 	<p>všechny sekce</p>

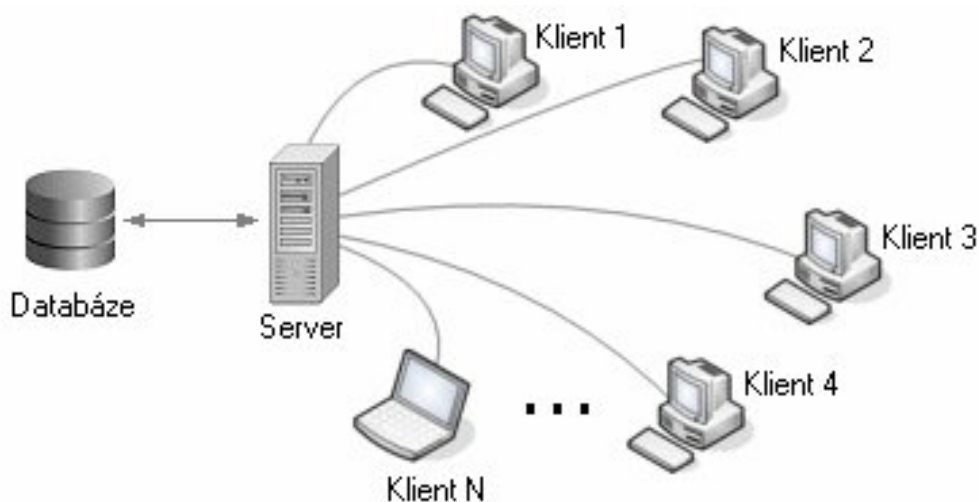
Tabulka 3.2: Přehled typů uživatelů informačního systému a jejich práv, 2. část

Kapitola 4

Návrh řešení

4.1 Architektura

Nad architekturou není třeba příliš přemýšlet. Stručně řečeno, hlavní funkce aplikace spočívá v uchovávání dat a v poskytování prostředí pro jejich vkládání, zobrazování a editaci. Vzhledem k tomu, že je systém určen pro širokou veřejnost, je jasné, že do něj bude přistupovat velké množství uživatelů a navíc z různých míst. Základem aplikace bude databáze, která musí být přístupná více uživatelům a navíc přístupná přes aplikaci samotnou. Z toho vyplývá, že jak databáze, tak i aplikace musejí být na místě, které je přístupné pro velké množství uživatelů. Tento požadavek skvěle zajišťuje internet. K aplikaci tak bude mít přístup každý, kdo disponuje připojením k internetu a webovým prohlížečem. Této síťové architektuře, kdy je klient (v tomto případě webový prohlížeč) oddělen od serveru, který je obvykle na vzdáleném počítači (v tomto případě aplikace s databází na internetovém serveru) a vzájemně spolu komunikují, se nazývá klient-server architektura. Zachycuje ji obrázek 4.1



Obrázek 4.1: Architektura klient-server

4.2 Výběr technologií

4.2.1 Technologie současného webdesignu

Představme si situaci, kdy by na webu neexistovaly žádné standardy. Každý výrobce prohlížeče by mohl používat vlastní jazyk pro návrh webových stránek, vlastní protokol pro komunikaci mezi serverem a prohlížečem apod. Tvůrce stránek by v takovém případě musel implementovat všechna taková řešení, což by bylo velmi nákladné a náročné, nebo ještě hůře, tvůrce by podporoval jen vybrané prohlížeče. Takové prostředí by potom velmi nahrávalo vzniku neprůhledných monopolů, což by byla zajisté neúnosná situace.

Naštěstí ale existuje relativně nezávislá, nezisková standardizační organizace s názvem W3C (World Wide Web Consortium). Produkuje veřejné normy a stará se o jejich propagaci. Tyto normy jsou vyvíjeny odborníky v příslušných oblastech a navíc jsou většinou přístupné komentářům okolí, a proto obsahují obvykle nejlepší možná řešení. Pro všechny subjekty pohybující se v dané oblasti je potom nejjednodušší tyto normy používat, protože si mohou být jisté, že s takovými produkty nebude mít problém žádný jiný subjekt z dané oblasti, a navíc nemusí utrácet prostředky na vývoj vlastních řešení.

W3C založil v roce 1994 jeden z duchovních otců webu, Tim Berners-Lee, v zájmu jeho standardizace. Dnes je v této organizaci přímo angažováno přes 60 lidí a mnoho spolupracovníků. Každá W3C norma prochází vývojem popisovaným takzvanými „*Recommendation track technical reports*“. Jsou to zprávy, které procházejí přijímacím procesem, na jehož konci se stávají platnou specifikací. Tyto zprávy procházejí následujícími pěti stádii, v každém získávají konkrétnější a propracovanější podobu až do konce procesu, kdy se stávají specifikací (v přesném překladu „doporučením“):

1. **Working draft** (pracovní návrh) – tato první veřejná zpráva je čistě pracovní, může v ní dojít k zásadním změnám a není nijak uznána W3C.
2. **Last call working draft** (poslední pracovní návrh) – tato zpráva je již pokročilejším pracovním návrhem, je určena především ke komentování ostatními skupinami W3C, členy W3C a veřejností.
3. **Candidate Recommendation** (kandidát na specifikaci) – tato zpráva je upravena podle komentářů veřejnosti i ostatních skupin a členů W3C. Zároveň v sobě zahrnuje ohled na ostatní dokumenty W3C. Je výzvou ke zkušebnímu zavedení dané technologie, přičemž W3C skupina, která zprávu vydala, je otevřena komentářům a zkušenostem se zavedením technologie.
4. **Proposed Recommendation** (navržená specifikace) – tato zpráva čeká na schválení W3C.
5. **Recommendation** (specifikace, v terminologii W3C doporučení) – tato zpráva je uznána W3C jako platná specifikace určitého standardu a je určena k zavedení a propagaci.

Nyní zde uvedu některé W3C normy, které se týkají této práce resp. webového informačního systému, o kterém práce pojednává.

XHTML

XHTML (zkratka anglického „eXtensible Hypertext Markup Language“ - „rozšiřitelný značkovací jazyk pro hypertext“) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý konsorciem W3C. Jedná se o nástupce jazyka HTML (Hypertext Markup Language), který prodělal poněkud živelný vývoj završený verzí HTML 4 (poprvé publikována v prosinci 1997, poslední revize pod označením HTML 4.01 je z prosince 1999). Dnes již spíše historický význam má verze HTML 3.2 z ledna 1997.

Jazyk XHTML verze 1.0 získal od W3C status doporučení (tj. nejvyšší status) v lednu 2000. Účelem i významovým rozsahem se jedná o HTML 4.01, řídí se však syntaktickými pravidly XML. Modularizovaná verze XHTML vstoupila v platnost v květnu 2001 pod označením XHTML 1.1. *Modularizace* přináší možnost vývoje různých XHTML profilů uzpůsobených pro různé typy zařízení a skupiny uživatelů. Příkladem XHTML profilu, vyvinutého pro přenosná zařízení (mobilní telefony, PDA, pagery, atd.), je XHTML Basic z prosince 2000.

Stávající verze XHTML jsou:

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.

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

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 Mobile Profile podporuje rozdíl od WAP 2.0 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. Dokumenty tohoto typu mají standardně koncovku .xhtml.

XHTML-Print – Zaměřený na tiskový výstup. Tato verze je ve třetím vývojovém stádiu, tzn. Candidate Recommendation.

XHTML 2.0 – Není zamýšleno tak, aby bylo zpětně kompatibilní se svými předchůdci. Je teprve v prvním vývojovém stádiu, které má označení Working Draft.

Jako nejvhodnější pro IS, o kterém pojednává tato práce, jsem shledal použití verze XHTML 1.0 Strict spolu s technologií CSS popsanou níže.

CSS

CSS je zkratka pro anglický název „Cascading Style Sheets“, česky „kaskádové styly“. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Technologie CSS byla navržena konsorciem W3C. Doposud byly vydány zatím dvě verze specifikace CSS1 a CSS2 (plus CSS 2.1), a pracuje se na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu.

Původně měl toto umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se nevyvinul jak bylo zamýšleno. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, což je dnes hlavní myšlenkou vývoje, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takovýto vývoj žádoucí. Dalo by se říci, že CSS je jakýmsi rozšířením možností zobrazení obsahu (X)HTML.

DOM

DOM (Document Object Model) je objektově orientovaná reprezentace webových dokumentů umožňující modifikaci obsahu, struktury nebo stylu dokumentu a jeho částí. DOM představuje standardizované programátorské rozhraní pro přístup ke struktuře, stylu a událostem dokumentu. Využívá koncepcí OOP (objektově orientované programování), není však vázán na konkrétní programovací jazyk. Tím umožňuje dalším softwarovým aplikacím pohodlně manipulovat s HTML a XML dokumenty a vytvářet tak dynamické webové stránky.

V mém případě informačního systému bude technologie DOM využívána jazykem JavaScript viz. podkapitola 4.2.2 téma JavaScript.

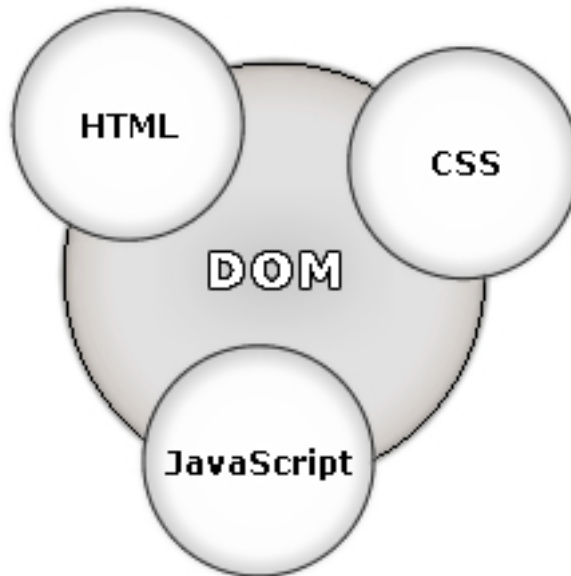
DHTML

Technologie DHTML (Dynamic Hypertext Markup Language) vychází ze stejné myšlenky jako skriptovací jazyk JavaScript, popsáný níže. Jejím cílem je dokument reagující na události bez spolupráce se serverovou stranou webové aplikace (např. změna obsahu stránky, validace formulářů). V DHTML je přístupný každý objekt webové stránky a s každým objektem je možné manipulovat – měnit jeho obsah, způsob zobrazení apod. Jako objekty jsou přístupné i kaskádové styly připojené ke stránce. DHTML je skloubením následujících technologií:

- DOM
- Skriptovací jazyk na klientské straně - JavaScript
- Kaskádové styly CSS

Grafické znázornění technologie DHTML můžeme vidět na obrázku 4.2.

Informace o technologiích webdesignu byly čerpány ze stránek konsorcia W3C [7], portálu Interval.cz [2], zabývajím se přímo problematikou webdesignu a otevřené encyklopedie Wikipedie [8].



Obrázek 4.2: Grafické znázornění technologie DHTML

4.2.2 Technologie pro vývoj dynamických webových stránek

Nejprve je třeba vysvětlit pojmy *skriptovací technologie* a *objektové technologie*.

skriptovací technologie – z hlediska webové aplikace je to technologie, kdy jsou určité části kódu (nazývány skripty) vloženy do původního kódu značkovacího jazyka (např. HTML). Tyto skriptové části se nejdříve nějakým způsobem zpracují a interpretují a až poté je výsledný celistvý kód značkovacího jazyka interpretován (standardně na straně klienta).

objektové technologie – tyto technologie podporují vývoj webových aplikací objektovým způsobem. Od skriptovacích technologií se liší používáním vlastní pokročilé infrastruktury převzaté z technologie, která má širší záběr než jen podporu vývoje webových aplikací. Často to může práci ulehčit – k dispozici jsou totiž automaticky některé pokročilé nástroje.

Současné nejpoužívanější jazyky užívané při vytváření webových aplikací jsou:

- **ASP** (Active Server Pages)
- **JSP** (JavaServer Pages)
- **PHP** (PHP: Hypertext Preprocessor)

ASP

První verze ASP (Active Server Pages) byla vydána na konci roku 1996. Tato technologie, vyvinutá firmou Microsoft, patří mezi technologie skriptovací. Soubory ASP jsou HTML stránky obohacené

o skripty. Je nutno upozornit, že ASP není jazyk, ale serverová technologie, jejímž základem je skriptovací jazyk. Používat se dá téměř kterýkoliv skriptovací jazyk, přičemž nejvíce používané jsou jazyky VBScript (Visual Basic Script) a JScript (implementace jazyka JavaScript od firmy Microsoft).

Koncem devadesátých let dvacátého století začala firma Microsoft pracovat na zcela nové platformě, která byla pojmenována *.NET*. Podle Microsoftu je to „sada softwarových technologií společnosti Microsoft pro propojování světa informací, lidí, systémů a zařízení“ [3]. Se vznikem *.NET Frameworku* vzniká i nová objektová technologie **ASP.NET** odvozená od původní skriptovací ASP. Není to ovšem jen další verze ASP, nýbrž se jedná o zcela novou technologii. Není to ani samostatná technologie pro tvorbu webových stránek, ale integrální součást nové obsáhlé platformy *.NET*. Značí to i fakt, že ASP.NET nelze nainstalovat samostatně, ale pouze s celým *.NET Frameworkem* jako jeho součástí. Ten je zdarma dostupný pro vlastníky operačního systému Windows.

Základní stavební kameny ASP.NET jsou *Web Forms*, *Web Services* a *Mobile Internet Toolkit*. *Web Forms* slouží k vytváření webových stránek pomocí komponent, tak, jako je tomu zvykem při vytváření „oknových“ aplikací ve Windows. Například i zde je vidět společné zástřeší platformy *.NET*. *Web services* jsou služby pro komunikaci aplikací přes internet a *Mobile Internet Toolkit* je sada nástrojů pro generování výstupu pro mobilní zařízení.

JSP

Technologie JSP byla vyvinutá společností Sun Microsystems, za účelem vývoje aplikací na straně serveru. Samotné JSP není nikterak složité, jsou to pouhé HTML stránky, do kterých jsou vloženy úseky kódů v jazyce Java, které jsou ohraničeny speciálními uvozujícími a ukončujícími posloupnostmi znaků zvanými *tagy*. Tento kód v jazyce Java potom vytváří onen požadovaný dynamický obsah stránky. Použit JSP je vhodné hlavně tehdy, když větší část stránky tvoří statický obsah. V opačném případě je vhodné rovnou použít na vytváření stránek technologii *servletů*. Servlet je totiž generovaný z každé zpracovávané JSP stránky, ať už stránka obsahuje více kódu v Javě nebo v HTML.

Servlet je neoddělitelnou součástí platformy J2EE od firmy Sun Microsystems. Jsou to programové komponenty napsané v jazyce Java běžící na straně serveru. Tak, jako všechny aplikace v Javě, pro svůj chod potřebují běžící *JVM* (Java Virtual Machine). Funkce JVM je v podstatě interpretování generovaného kódu od překladače. Servlety jsou tedy načítány a spouštěny pomocí JVM na příslušném serveru s aplikací. Podle specifikace jsou servlety nevizuální komponenty, tzn. že se nezobrazují pomocí žádného grafického uživatelského rozhraní (GUI). To však neznamená, že nemůžou „generovat“ uživatelské rozhraní. Většinou se totiž používají servlety pracující nad internetovým protokolem HTTP. Jejich odpovědí je potom v převážné části HTML kód. Servlety se mohou použít na různé úlohy jako např. čtení nebo zápis do databáze, posílání e-mailů, zpracovávání formulářů, posílání cookies atd. Jednoduše jsou určeny pro tvorbu dynamických internet/intranet řešení, podobně jako JSP. (Zdrojem informací o technologii JSP mi byl internetový portál Interval.cz [2])

PHP

PHP (původně *Personal Home Page*, nyní *PHP: Hypertext Preprocessor*) je podle jeho autorů „široce používaný v hlavním smyslu skriptovací jazyk, který je vhodný zejména pro vývoj webových aplikací a může být vložený do HTML“ [5].

Vznik jazyka PHP je datován rokem 1994, kdy jeho tvůrce, dánsko-kanadský programátor Rasmus Lerdorf, ve svém volném čase vytvořil jednoduchý systém pro evidování přístupu ke svým stránkám v jazyce Perl. Později autor přepsal systém do jazyka C. Po zdokonalení systému jej uvolnil pod názvem Personal Home Page Tools, který se později změnil na Personal Home Page Construction Kit. Postupem času Rasmus Lerdorf vytvořil i nástroj pro používání databáze SQL. Tento program pro zpřístupnění databází na webu nesl název Form Interpreter (FI). Spojením těchto dvou programů vznikl systém PHP/FI 2.0, který si získal celosvětovou proslulost. Obsah zkratky PHP dnes ztratil svůj původní význam a nynější označení celého systému je „PHP: Hypertext Preprocessor“.

PHP je tedy skriptovací jazyk, jehož příkazy jsou do webových stránek (přesněji do HTML kódu) umístěny mezi speciální oddělovače – tagy (<? ?> nebo <?php ?>). Vlastní skriptovací jazyk kombinuje několik programovacích jazyků – C, Perl, Java, obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k různým databázovým serverům, podporu řady internetových protokolů (HTTP, FTP, POP3, SMTP, ...) a lze ho také jednoduchým způsobem doplnit o funkce nové, které standardně nejsou jeho součástí. Stránky obsahující PHP vsuvky jsou pro odlišení od běžných webových stránek ukládány s příponou .php (.php3, .phtml). Vývoj technologie PHP stále pokračuje. V současné době je k dispozici verze 5.2.1. S pátou verzí se výrazně zlepšil přístup k objektově orientované programování, který je podobný jazyku Java.

Velkou výhodou technologie PHP je fakt, že je šířena jako freewareový produkt včetně zdrojových kódů. PHP není závislé ani na platformě ani na webovém serveru na kterém je provozováno. Lze jej tedy užívat jak např. pod Windows tak i pod jinými operačními systémy s tím, že není svázáno s žádným konkrétním serverem a proto může běžet na libovolném. Dokazuje to velké procento serverů na kterém je PHP v současnosti používáno a jeho velká podpora ze strany hostingových služeb.

PHP jazyk je velice oblíbený. Jeden z důvodů je jednoduchost jeho použití. Tím, že PHP kombinuje vlastnosti více programovacích jazyků, ponechává vývojáři částečnou svobodu v syntaxi.

Při výběru technologie jsem se rozhodl pro jazyk PHP. Důvodů je několik:

- PHP je v současnosti nejrozšířenější a nejpoužívanější jazyk pro vývoj webových aplikací
- vzhledem k veliké rozšířenosti PHP lze o tomto jazyku najít mnoho informací, rad a návodů
- JSP i ASP jsou vhodnější spíše pro rozsáhlé projekty, na kterých pracuje více lidí. Jelikož jde v této práci o projekt menšího rozsahu, navíc vytvářený jednou osobou, je PHP optimální volbou.

- JSP i ASP mají jednu společnou nevýhodu – jsou to komerční produkty, které nejsou nikterak levné a jejich použití je navíc svázáno s použitím WWW serveru dané firmy. ASP navíc běží pouze na platformě operačním systému Windows – pokud by bylo v budoucnu potřeba používat IS na jiné platformě, nebylo by to možné.
- technologie PHP je podporována téměř na všech webhostingových serverech, což se o ASP ani JSP říci nedá.

Spolu s PHP budu používat i jazyk *JavaScript*.

JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape.

Zpravidla se používá jako interpretovaný programovací jazyk pro WWW stránky, vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Narozdíl od ostatních zmíněných jazyků, patří mezi skriptovací jazyky, jejichž kód je zpracováván a interpretován na straně klienta, což vede k celkovému zrychlení aplikace a snížení zátěže serveru. Nevýhodou je ovšem fakt, že ne každý klient má povoleno tento jazyk na svém prohlížeči spouštět. K tomuto problému ovšem bude při vývoji IS přihlíženo.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. I když se to na první pohled může zdát, s jazykem Java nemá, až na podobný název a podobnou syntaxi, nic společného. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Standards Organization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript (některé informace o JavaScriptu byly čerpány z online encyklopedie Wikipedie [8]).

4.2.3 Databáze

Databázových systémů dnes existuje mnoho, mezi nejpoužívanější patří:

- Microsoft SQL Server
- Oracle
- PostgreSQL
- MySQL

Microsoft SQL Server

Microsoft SQL Server je robustní a spolehlivý databázový systém společnosti Microsoft, který tvoří databázovou vrstvu (data management tier, zadní vrstvu nebo back-end) aplikací podnikových informačních systémů strategického významu. Výhodou, že MS SQL obslouží bez problému provozní

informační systém firmy i její interaktivní webovou prezentaci, lze spatřovat v jednoduchém propojení těchto aplikací, a z toho plynoucí např. výhodu pro elektronické obchodování.

Omezením tohoto databázového systému je závislost na platformě Microsoft Windows a ačkoliv se jedná o kvalitní produkt, jeho cena se pohybuje nad možnostmi běžného uživatele.

Oracle

Databázový systém Oracle je vyvíjen firmou Oracle Corporation. Aktuální verzí je Oracle Database 10g. Tento systém podporuje nejen standardní relační dotazovací jazyk SQL podle normy SQL92, ale také proprietární firemní rozšíření Oracle (např. pro hierarchické dotazy), imperativní programovací jazyk PL/SQL rozšiřující možnosti vlastního SQL (v tomto jazyce je možné tvořit uložené procedury, uživatelské funkce, programové balíky a triggerly), dále podporuje objektové databáze a databáze uložené v hierarchickém modelu dat (XML databáze, jazyk XSQL). V současnosti je jeden z nejpropracovanějších databázových systémů.

Jeho podstatná nevýhoda je, že neexistuje bezplatná verze a ceny komerčních licencí jsou vysoké. Patrně i z tohoto důvodu je provozován pouze úzkým okruhem hostingových služeb.

PostgreSQL

PostgreSQL je plnohodnotným relačním databázovým systémem s otevřeným zdrojovým kódem. Je spolehlivý a bezpečný, běží na všech rozšířených operačních systémech (jako Linux, různé typy UNIXů, Windows) a je šířen pod BSD licencí (Berkeley Software Distribution), která je nejliberálnější ze všech open source licencí. Systém PostgreSQL může být dle licence neomezeně používán, modifikován i distribuován, z čehož plyne velká přednost, kterou je rozšířitelnost.

MySQL

MySQL je relační databázový systém, vytvořený švédskou firmou MySQL AB. Je k dispozici jak pod bezplatnou licenci, tak pod komerční placenou licenci, jejíž cena je však pouze zlomkem ceny konkurenčních produktů. Databázový systém MySQL je vysoce výkonný a stabilní relační databázový systém, který dokáže pojmout velké množství dat, aniž by při tom ztratil mnoho ze svého výkonu. Komunikace probíhá pomocí jazyka SQL s některými rozšířeními. MySQL není vázán na konkrétní platformu a může být využit na všech dnes používaných operačních systémech.

Pro svůj výkon a díky bezplatné šířitelnosti zaujímá systém MySQL vysoký podíl používanosti v současných databázových aplikacích. Velmi oblíbená a jako základní kombinace webového serveru nasazovaná je kombinace MySQL, PHP a serveru Apache, popsaného níže. Užitečnou a používanou pomůckou pro správu databáze MySQL prostřednictvím webového rozhraní je aplikace phpMyAdmin napsaná v jazyce PHP.

Pro vysokou cenu a malou podporou ze strany hostingových služeb jsem z výběru použitelného databázového systému vyloučil produkty Oracle i Microsoft SQL Server. Ze zbylých dvou databází

PostgreSQL a MySQL jsem zvolil systém MySQL, pro jeho rozsáhlou podporu ze strany hostingových služeb a vysoký výkon.

4.2.4 Webový server

Nejpoužívanějšími webovými servery v současné době jsou servery *Apache* a *IIS* od firmy Microsoft.

IIS

IIS (Internet Information Services) označuje sadu internetových služeb pro servery běžící pod operačním systémem Microsoft Windows. Tento server používá již zmiňovaná technologie ASP.

Apache

Apache HTTP Server je softwarový webový server s otevřeným kódem pro téměř všechny operační systémy. V současné době dodává prohlížečům na celém světě většinu internetových stránek. Jeho vývoj začal v roce 1993 v NCSA (National Center for Supercomputing Applications) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. Avšak v dalším roce, kvůli odchodu hlavního programátora, došlo ke zpomalení vývoje a poté, v roce 1998, k úplnému zastavení. NCSA HTTPd však mezitím již používali správci webových serverů a dodávali k němu vlastní úpravy – tzv. *patche* (patch v češtině znamená „záplata“). Odtud pochází název webového serveru Apache „A patchy server“ („záplatovaný server“). Hlavní úlohu v dalším vývoji sehráli Brian Behlendorf a Cliff Skolnick, kteří založili e-mailovou 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 (čerpáno z Wikipedie [8]).

Dle společnosti Netcraft [4], která se zabývá monitorováním internetu se zaměřením na analýzu používaných webových serverů a operačních systémů na kterých servery běží, je Apache nejpoužívanějším webovým serverem, který je používán na nadpoloviční většině webových serverů (ke květnu 2007 je to 56% zastoupení). Druhým nejpoužívanějším je server IIS pohybujícím se v rozmezí 20-35 procent zastoupení.

Výběr webového serveru, na kterém má být vytvořený systém provozován, souvisí s výběrem jazyka pro vývoj dynamických webových stránek. Vzhledem k faktu, že webová aplikace informačního systému bude vyvíjena za pomoci jazyka PHP, je zde jasnou volbou server Apache.

4.2.5 Kódování

Aby počítač mohl pracovat s textem, musí mít nadefinováno, jaké znaky se v textu mohou vyskytovat. Množina znaků, kterou je možné v textu používat se nazývá *znaková sada*. Znaková sada je právě taková množina znaků, kde každý znak má přiřazený číselný kód. Reprezentace těchto

jednotlivých znaků číselným kódem se nazývá *kódování* (obecně je kódování reprezentace nějaké konečné množiny – zpravidla znaků – pomocí nějaké jiné konečné množiny – zpravidla čísel).

Asi nejznámější ze znakových sad je *ASCII* (American Standard Code for Information Interchange), která definuje 128 znaků, jež zahrnují písmena anglické abecedy, číslice, interpunkční znaménka a pár dalších speciálních znaků. Těchto 128 znaků je ale dostačujících pouze pro angličtinu. Ostatní jazyky včetně češtiny zde postrádají některé své specifické znaky, jako jsou písmena s diakritickými znaménky a přehláskami nebo uvozovky dole. Vznikly tedy další sady, které jsou rozšířením ASCII – definují celkem 256 znaků. Prvních 128 znaků je stejných jako v ASCII a zbylých 128 je určeno pro použití pro národní sady. Nejpoužívanější znakové sady pro češtinu v současnosti jsou *ISO Latin 2* (ISO 8859-2) a *Windows 1250*.

ISO 8859-2 je kódování standardnější, používané v operačních systémech Unix a Linux, ale je podporováno i operačním systémem Windows. Někdy se označuje jako Latin 2, ISO Latin 2.

Windows-1250 je preferováno v operačním systému Windows, ale je podporováno i na systémech Unix a Linux.

Pro obě zmiňovaná česká kódování platí, že jeden byte (byte je tvořen z 8-mi bitů, z nichž každý může mít pouze 2 hodnoty – 0 a 1, tzn. celkem 2^8 možností tzn. 256 znaků) reprezentuje 1 znak.

Avšak tyto 8bitové sady nejsou tím pravým řešením, protože v nich není dost místa pro některé jazyky a navíc je problém, pokud je třeba použít např. ruské a české znaky najednou. Proto vznikly znakové sady, v kterých je dost místa pro všechny znaky všech používaných jazyků. Jsou to 32bitová (4bytová) *ISO 10646* (nazvaná Universal Character Set – *UCS*) a 16bitová (2bytová) *Unicode*. První může obsahovat 2^{32} (4 294 967 296) a druhá 2^{16} (65 536) znaků, což nabízí mnohem větší místo pro znaky všech možných jazyků najednou.

Jak již bylo zmíněno, kódování přímo určuje, jak je znak (respektive jeho kód ve znakové sadě) převeden na sekvenci bitů. Mezi široce užívaná mezinárodní kódování patří *UTF-8* a *UTF-16*, pro češtinu je možnost použití kódování ISO 8859-2 a Windows-1250.

- **ISO 8859-2 a Windows-1250** – u těchto 8-bitových znakových sad není kódování nikterak složité, kódují všechny své znaky do 8 bitů (1 bytu).
- **UTF-8** – kóduje znaky z ASCII do sedmi bitů a ostatní znaky do 2 až 6 bytů (tj. 16 až 48 bitů). České znaky jsou kódovány do 2 bytů. Číslo „8“ v názvu zde znamená, že nejmenší možná kódová jednotka má délku 8 bitů.
- **UTF-16** – kóduje všechny znaky do dvou bytů (16 bitů) a je defacto totožné se znakovou sadou Unicode, pouze obsahuje navíc několik speciálních znaků. Kódová jednotka zde vždy má velikost 16 bitů.

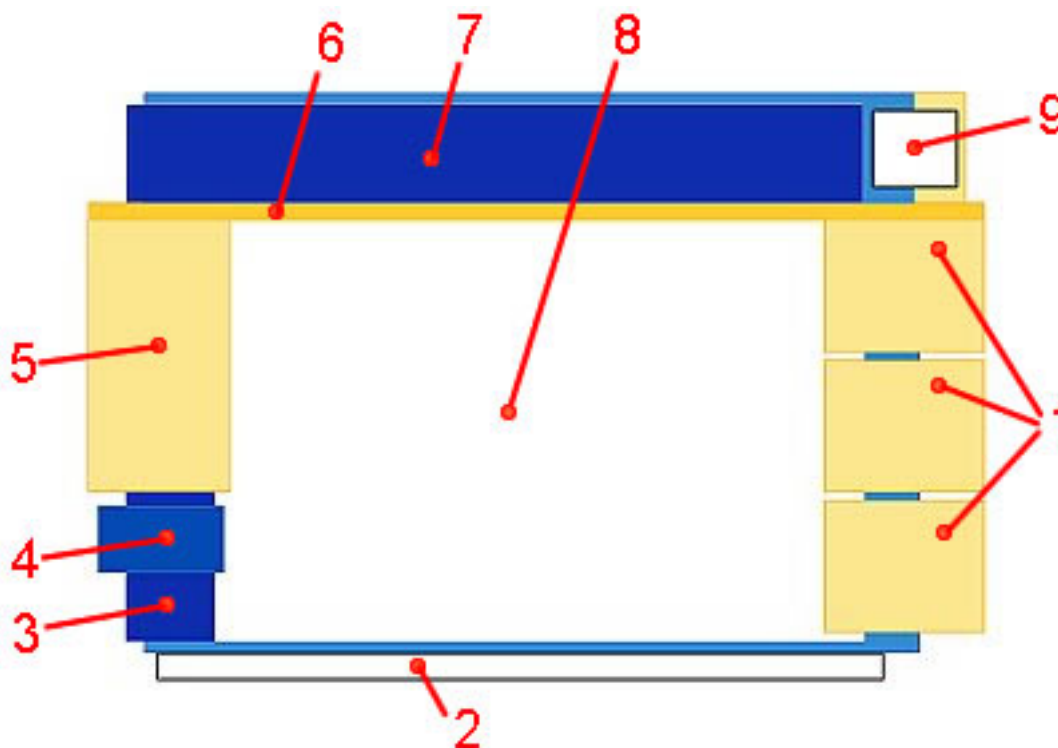
Některé informace o kódování byly čerpány z portálu Interval.cz [2].

Z hlediska velikosti souboru (což je pro web důležité), se nejlépe jeví kódování ISO 8859-2 a Windows-1250, hned za nimi potom kódování UTF-8. Avšak pro svou rozšířenost a především mezinárodnost, je jistě pro IS nejlepším řešením kódování UTF-8.

4.3 Návrh vzhledu, layout

Layout znamená základní, zpravidla kreslený koncept obsahového a grafického rozvržení dokumentu (velikost, umístění, barvy pozadí a textu atd.). Při návrhu layoutu by měl být brán ohled na přehlednost a čitelnost dokumentu.

Návrh layoutu ukazuje obrázek 4.3:



Obrázek 4.3: Layout stránek informačního systému, rozmístění komponent

Popis jednotlivých částí z obrázku 4.3:

- 1** – Tyto oblasti slouží jako informační panely. Budou zde zobrazeny prvky jako anketa, nastávající a uplynulé události a sponzoři. Po přihlášení uživatele (člena klubu) zde přibude i kalendář.
- 2** – V tomto místě bude tzv. *pata* stránky. Pata obsahuje informace jako kdo je právě online, čas vygenerování stránky, autora stránek, datum vzniku, dále pro jaké prohlížeče a rozlišení je stránka optimalizovaná atd. Bude zde i kontakt na autora stránek.
- 3** – Místo označené číslem 3 je vyhrazeno pro počítadlo přístupů.
- 4** – Tato oblast bude obsahovat formulář pro přihlášení a odhlášení členů klubu.
- 5** – Oblast 5 je určená pro hlavní navigaci na stránkách – menu. Pomocí menu se bude měnit obsah oblasti 8.

- 6** – Na tomto pruhu budou v levé části zobrazeny krátké aktuální informace a v pravé části čas a datum. Uprostřed tohoto pásu se bude nacházet interní navigace – interní menu. Toto menu bude zobrazeno pouze po přihlášení uživatele.
- 7** – Oblast s číslem 7 bude obsahovat název klubu, logo nohejbalu popř. nějaké další grafické prvky.
- 8** – Tělo stránky. V této části budou zobrazovány obsahy různých kategorií menu, sekcí, hlášení a editací.
- 9** – Místo pro logo klubu nebo města.

Obsahové a grafické rozvržení informačního systému bude zpracováno bez použití HTML tabulek (z důvodu zvýšení rychlosti načítání, viz. kapitola 3.1), pouze pomocí blokových a řádkových elementů XHTML a za pomoci kaskádových stylů CSS.

4.4 Způsob zabezpečení

Správné zabezpečení aplikace obnáší především:

- Ověření identity uživatele (tzv. *autentizace*).
- Přiřazení přístupových práv uživateli (tzv. *autorizace*).
- Potvrzení, že požadavek byl doručen beze změny (tzv. *integrita*).
- Zaručení, že nikdo nemůže během přenosu zjistit obsah zprávy (tzv. *utajení*).

autentizace

Autentizace je proces, při kterém se ověřuje, zda je uživatel nebo entita opravdu ten, za koho se vydává.

Autentizace v IS bude řešena nutností zadat uživatelské jméno a heslo do HTML formuláře. Po jeho odeslání se data ověří a v případě jejich správnosti systém uživatele přihlásí.

autorizace

Při autorizaci se ověřuje, zda má uživatel dostatečná oprávnění pro přístup k určitému souboru či pro provedení určité akce. Tato kontrola se provádí na základě členství uživatele v různých uživatelských skupinách, přístupových seznamech apod. Předpokládá se předchozí úspěšná autentizace, na jejíž spolehlivosti je autorizace závislá.

Autorizace v IS bude zajišťována pomocí údajů o právech uživatele uložených v databázi.

integrita

Integrita je stav, kdy přečtená data jsou totožná s daty uloženými. Tzn. během uložení (přenosu) dat nedošlo k jejich neočekávaným změnám. Dalo by se také říci, že integrita je platnost dat.

Nejnebezpečnějším porušením integrity v IS je porušení záměrné např. nějakým útočníkem. Integrita citlivých dat, tj. hesel, bude v IS zajišťována *hašovací funkcí*. Je to jakýsi předpis pro výpočet kontrolního součtu (haše) ze zprávy či většího množství dat, zde konkrétně hesla. Pro zajištění integrity bude použita hašovací funkce s názvem SHA-1.

utajení

Utajení lze zaručit pomocí šifrování. Dominantní šifrovací technologií používanou na internetu je protokol SSL. Vzhledem k tomu, že se v této práci jedná o aplikaci menšího rozsahu, rozhodl jsem se šifrování SSL (Secure Sockets Layer) nepoužít. Ponechávám však možnost pro případné doplnění aplikace o toto šifrování.

4.5 Diagram případů užití

Diagram případů užití (Use Case diagram) se vytváří s pomocí uživatelů systému pro zjištění funkčních požadavků na systém. Diagram představuje soubor případů užití, aktérů a jejich vzájemných vztahů. Každý případ užití popisuje posloupnost událostí. Každá posloupnost je inicializovaná jistou entitou, která se nazývá *aktér*.

Diagram případů užití webového informačního systému nohejbalového klubu je znázorněn na obrázku 4.4.

Popis diagramu

Diagram je tvořen jednotlivými případy užití, aktéry a vztahy mezi nimi. Aktéři zde představují jednotlivé typy uživatelů a jsou reprezentováni ikonou lidské postavy. Jméno aktéra se nachází přímo pod ikonou. Případy užití jsou reprezentovány elipsou a jejich názvy se nachází přímo pod elipsou.

Popis diagramu z hlediska aktérů:

- **Návštěvník** – Tento aktér představuje každého návštěvníka informačního systému, který není členem klubu. Má možnost vidět, kdo z návštěvníků je právě online, pomocí kliknutí na jednotlivé sekce v menu může prohlížet tyto sekce, má možnost přidávat komentáře, příspěvky do guestbooku a hlasovat v anketě.
- **Člen klubu s nižšími právy** – Je to aktér vycházející z návštěvníka, který od něho přejímá všechny případy užití. Tento jev se nazývá *generalizace*. Generalizace se značí speciální šipkou s trojúhelníkem na konci. Ke členovi klubu s nižšími právy jsou přiřazeny případy užití jako přidávání akcí do kalendáře, editace osobních údajů atd., jak lze vidět na obrázku 4.4.

- **Člen klubu s vyššími právy a finančník** – Tito aktéři přejímají případy užití od člena klubu s nižšími právy. Liší se od sebe různými případy užití, jak je znázorněno na obrázku 4.4.
- **Vedení a Trenér** – Ze člena klubu s vyššími právy vycházejí aktéři „Vedení“ a „Trenér“. Můžeme vidět, že vedení zahrnuje o dosti více případů užití nežli trenér.
- **Administrátor** – Tento aktér přejímá případy užití od všech zmiňovaných aktérů. Navíc má možnost měnit hesla jednotlivým uživatelům.

4.6 ER Diagram

Datové modelování typicky ukazuje perzistentní data ukládaná v databázi. Má také zásadní význam pro návrh databází. Základním modelem je diagram entit a vztahů s názvem *ER Diagram* (Entity Relationship Diagram).

ER Diagram webového informačního systému nohejbalového klubu je znázorněn na obrázku 4.5.

Popis diagramu

ER Diagram modeluje data, která potřebujeme v systému uchovávat a vztahy mezi těmito daty. Je tvořen těmito prvky:

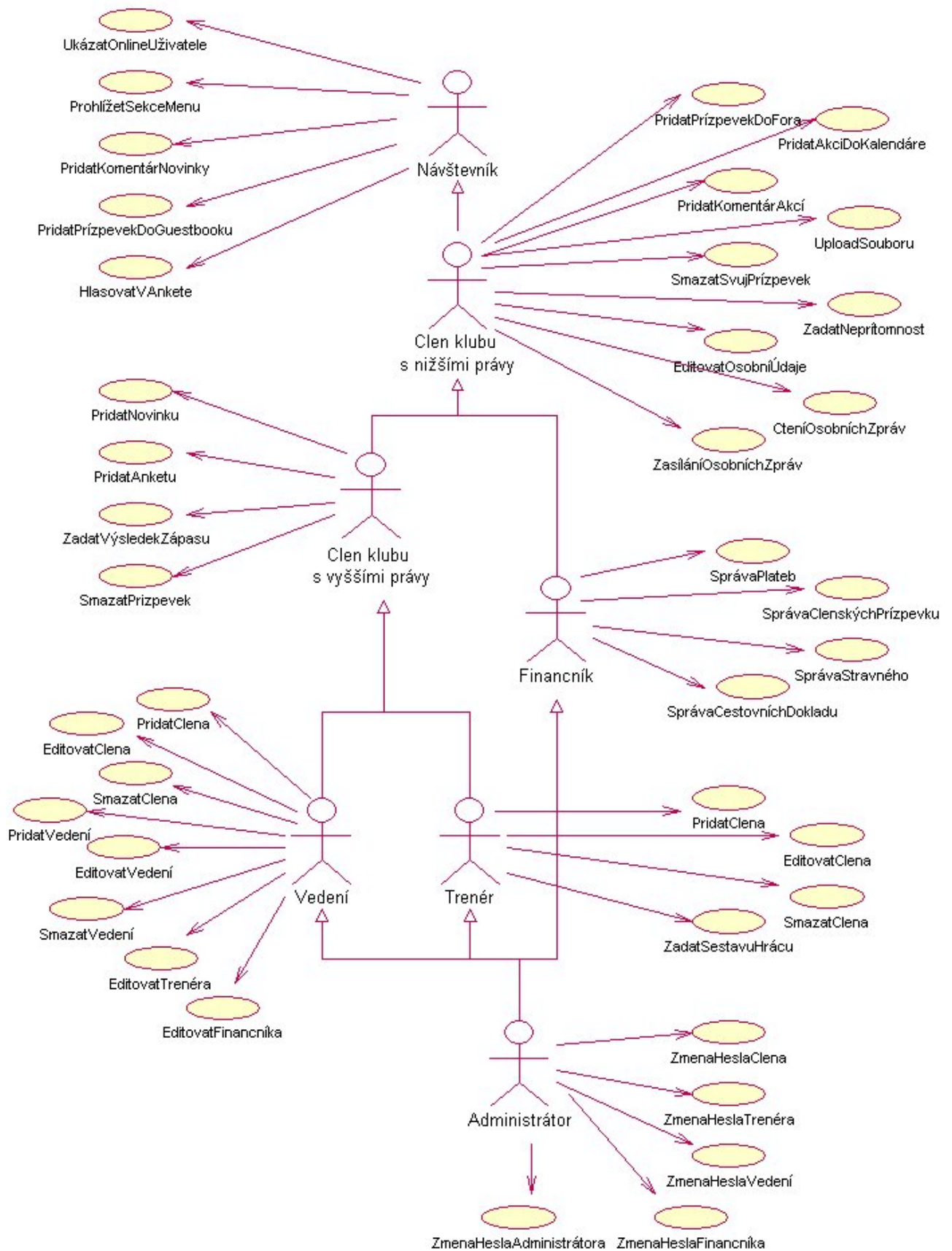
Entitní množina je množina entit téhož typu, které sdílí tytéž vlastnosti (atributy). Entitní množina je v ER Diagramu reprezentována tabulkou s názvem entitní množiny a názvy atributů.

Vztahová množina je množina vztahů téhož typu, které sdílí tytéž vlastnosti. Vztahová množina se vyskytuje obecně mezi jednou až několika množinami entitními a je reprezentována úsečkou mezi těmito množinami (většinou je u úsečky uveden i název vztahové množiny). Například v ER Diagramu na obrázku 4.5 existují entitní množiny „ČlenKlubu“ a „KalendářUdálost“, mezi nimiž existuje vztahová množina vyjadřující možnost člena klubu přidat událost do kalendáře.

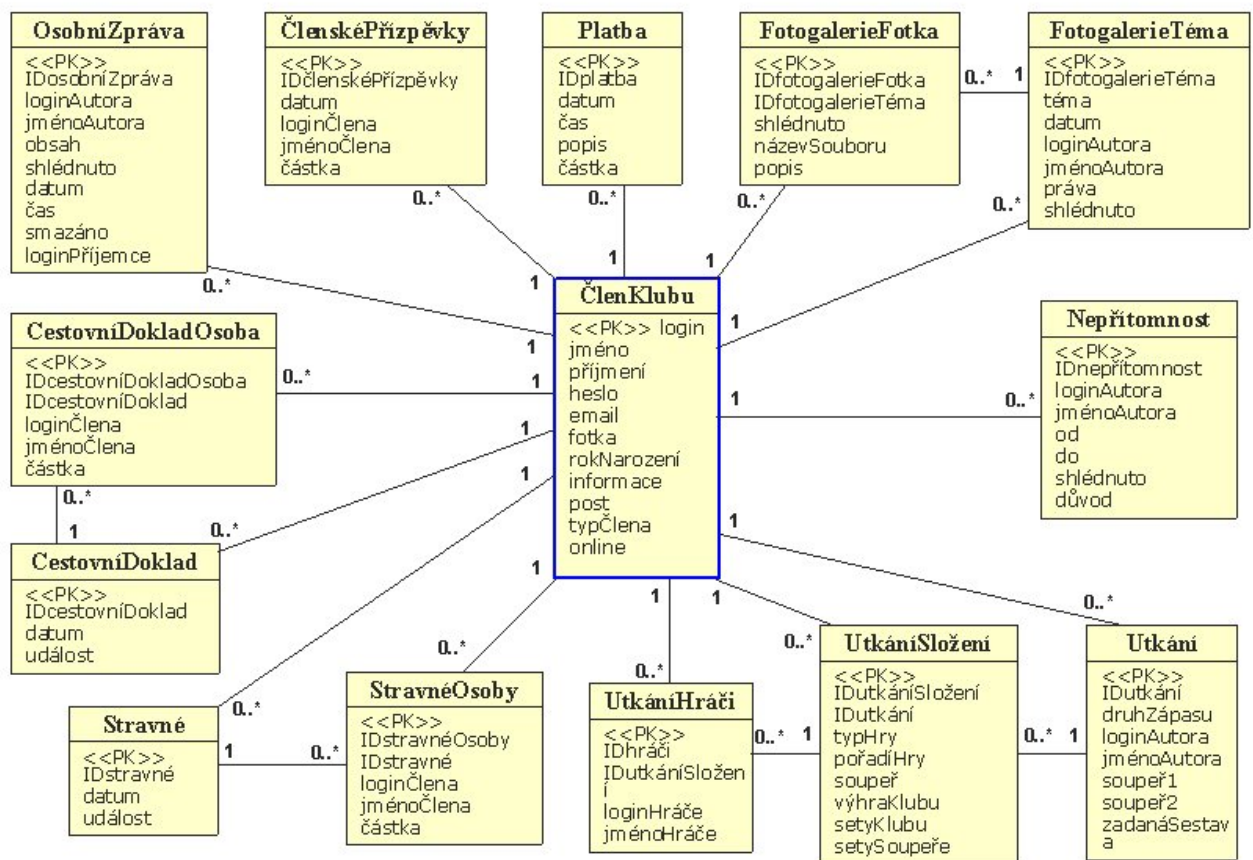
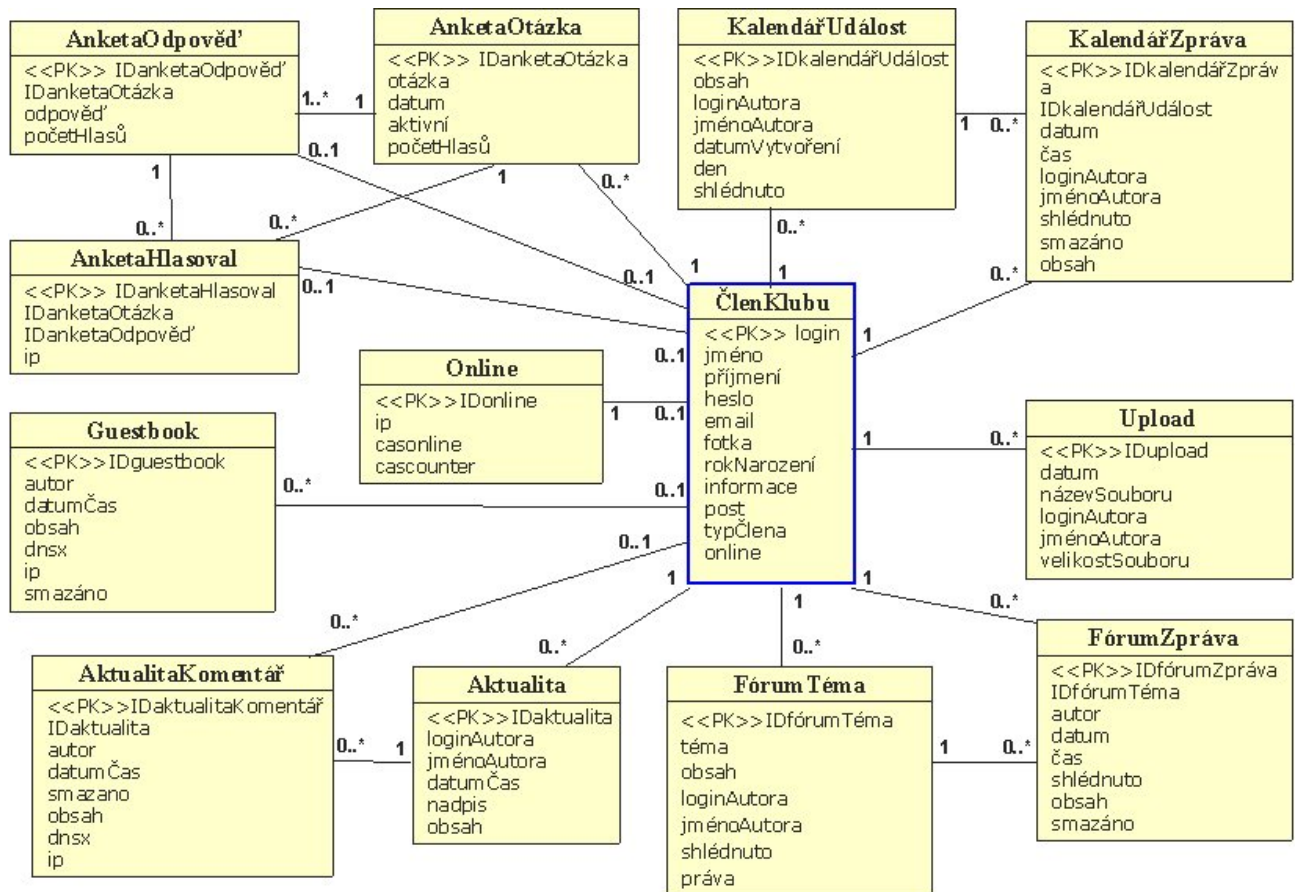
Kardinalita se značí u vztahové množiny a udává maximální počet vztahů vztahové množiny, ve kterých může participovat jedna entita. Např. v ER Diagramu na obrázku 4.5 entita typu „ČlenKlubu“ může zadávat 0 až neomezený počet událostí do kalendáře. Naproti tomu entita typu „KalendářUdálost“ má jednoho autora (člena klubu). Typické hodnoty, které se u kardinality uvádějí jsou: „0“, „1“ a „*“ (příp. „M“ místo „*“).

Diagram na obrázku 4.5 je rozdělen na dvě části z důvodu lepší přehlednosti obrázku. Entitní množina „ČlenKlubu“ se vyskytuje v obou částech obrázku, avšak je to jedna a tatáž množina.

Jednotlivé entitní množiny představují strukturu, v jaké budou data uloženy v databázi. Obecně platí, že jedna entitní množina představuje jednu tabulku v databázi a jeden atribut představuje jeden sloupec tabulky. Podle stávajícího ER Diagramu bude tedy databáze obsahovat celkem 26 tabulek. S ohledem na další možný vývoj aplikace tento počet nemusí být konečný.



Obrázek 4.4: Diagram případů užití



Obrázek 4.5: ER Diagram

Kapitola 5

Implementace

Tato kapitola již bude věnována samotné implementaci informačního systému. Cílem ovšem není popsat veškeré funkce a části informačního systému, ale vyzdvihnout ty, které jsou důležité nebo zajímavé. V případě potřeby detailnějších informací je možné nahlédnout do zdrojových kódů.

5.1 Adresářová struktura

Adresářová struktura aplikace je následující:

```
..
|-admincreate
|-css
|   |-sections
|-images
|   |-sponzori
|-include
|   |-connect
|   |-index
|   |-scripts
|   |-sections
|-memberphotos
|-photogalery
|-js
|-sql
|-tinymce
|-upload
```

.. V kořenovém adresáři se nacházejí dva důležité soubory – `index.php` a `!nastaveni.php`, které jsou popsány níže

`admincreate` V tomto adresáři se nachází inicializační script aplikace, pomocí něhož se vytváří přístupové heslo administrátora. Za chodu aplikace tento adresář nijak nevyužívá.

`css` Adresář obsahující soubory kaskádových stylů

— sections	Adresář obsahující soubory specifických kaskádových stylů podle aktuální zobrazené sekce.
include	V tomto adresáři se vyskytují soubory, které jsou do kódu vkládány PHP funkcí <code>include</code> (vyjma adresáře <code>scripts</code>).
— connect	Obsahuje skript pro připojení k databázi.
— index	Zde se vyskytují soubory, které jsou vkládány pomocí funkce <code>include</code> do hlavního souboru <code>index.php</code> .
— scripts	Tento adresář obsahuje samostatné skripty, které nejsou nikam „includovány“.
— sections	Adresář obsahující všechny sekce (každá sekce je reprezentována jedním souborem). Tyto sekce se zobrazují v oblasti ve středu aplikace.
memberphotos	Adresář <code>memberphotos</code> je určen pro ukládání profilových fotografií každého ze členů.
photogallery	Jak název napovídá, toto místo je vyhrazeno pro ukládání fotografií z fotogalerie.
js	Obsahuje skripty v jazyce JavaScript.
sql	Tento adresář je určen pro inicializační SQL skript, který vytvoří data v databázi a dále pro soubory obsahujícími zálohu databáze
tinymce	Je adresář WYSIWYG editoru (viz 3.1) s názvem TinyMCE.
upload	Do tohoto adresáře budou ukládány soubory nahráté na server uživatelem přes informační systém.

5.2 Popis hlavních skriptů a souborů

`index.php`

Hlavním souborem, který je načítán při každém zobrazení aplikace, je `index.php`.

Tento skript začíná vložením (includováním) skriptu `zacatek.php`, který spouští čas načtení stránky, aktivuje *session* proměnné (pomocí kterých systém odlišuje jednotlivé uživatele, kteří s aplikací pracují), pokud není zadána sekce, nastavuje implicitní sekci na „Aktuality“, vkládá soubor `connect.php` pro připojení k databázi, kontroluje čas pro automatické odhlášení, apod.

Následuje hlavička dokumentu, ve které jsou do dokumentu vloženy příslušné soubory kaskádových stylů CSS (pro právě aktivní sekci je dynamicky vložen příslušný CSS soubor pomocí PHP skriptu) a soubory s JavaScriptem, k nimž patří i soubor pro použití WYSIWYG editoru TinyMCE.

Další část kódu obsahuje elementy pro zobrazení horní části aplikace, tj. např. nadpis s názvem oddílu, logo, lišta pro interní menu atd.

Následuje část pro zobrazení navigačního menu aplikace. Za pomoci jednoduchého úseku php kódu je zvýrazněna aktuální sekce.

Po menu v kódu následuje vložení skriptu `login.php`, který má na starosti zobrazení formulářů pro přihlášení se do aplikace.

Další částí je část nazvaná „tělo“. To je složeno z několika do sebe zanořených blokových elementů, pro zajištění „natahování“ grafickým prvků na stranách těla. Do těla jsou vkládány skripty pro zobrazení aktuální sekce a zobrazení interního menu, pokud je uživatel přihlášen.

Dále je v souboru kód, který zobrazuje informační panely po pravé straně aplikace. Těmi jsou kalendář, události, anketa, a sponzoři. Každý panel je reprezentován patřičným vloženým skriptem.

Poslední částí hlavního skriptu `index.php` je zobrazení části s názvem „pata“. Ta vypisuje kolik návštěvníků je právě online, čas vygenerování stránky atd.

```
!nastaveni.php
```

Tímto skriptem se nastavují různé parametry aplikace, jako jsou doba automatického odhlášení, počet příspěvků zobrazených na stranu a mnohé další. Nezbytné je zde při inicializaci aplikace nastavit údaje potřebné pro připojení k databázi.

```
connect.php
```

Soubor `connect.php` má na starosti připojení k databázi. Tento soubor je ovšem také zásobárnou uživatelsky definovaných funkcí. Důvodem je, že soubor musí být přítomen v každém skriptu, protože je téměř vždy třeba komunikovat s databází. Tím je zajištěna přístupnost těchto uživatelsky definovaných funkcí všem skriptům.

Ze zajímavých funkcí bych zmínil například funkci `ArrayToURL`, jejíž vstupním parametrem je pole, a která vrací název všech proměnných pole a jejich hodnot v poli ve tvaru `?prom1=hodnota1&prom2=hodnota2&prom3=hodnota3...`, což se využívá pro potřebu zachování hodnot v adrese URL (tj. v globálním poli `$_GET`)

Další potřebnou funkcí je funkce s názvem `s1()`, která zajišťuje jednu z oblastí bezpečnosti aplikace. Řeší problém tzv. *SQL injection*, kdy útočník záměrně do pole URL nebo formulářů zadá hodnotu, která může způsobit nežádoucí příkaz v databázi. Takový problém se dá vyřešit pomocí vložení znaku zpětného lomítka do zadané hodnoty od uživatele před kritické znaky. Pro představu uvádím kód funkce.


```

/* funkce přidá k řetězci lomítka (obrana proti SQL injection)
   @param value ...řetězec, který má být upraven */
function sl( $value ) {
    // direktiva magic_quotes je na serveru zapnuta, netřeba dělat nic
    if( get_magic_quotes_gpc() ){
        // pokud není, vyzkouší se, jestli existuje funkce mysql_real_escape_string
        else if( function_exists( 'mysql_real_escape_string' ) ) {
            $value = mysql_real_escape_string( $value );
        }
        // pokud ne (tj. pro PHP verze < 4.3.0), použije se funkce addslashes
        else{
            $value = addslashes( $value );
        }
        return $value;
    }
}

```

styles.css

Hlavní soubor kaskádových stylů CSS. Definuje grafické výstupy hlavního souboru index.php. Je rozdělen do kategorií, jejichž názvy sami o sobě vypovídají o své působnosti, proto je není třeba nijak více komentovat. Jsou to „top“, „lista“, „grafika menu telo info“, „menu“, „pata“, „login“, „loginmenu“, „counter“, „kalendar“, „anketa“, „editacni odkazy“ a „sponzori“.

javascript.js

Tento soubor obsahuje funkce v jazyce JavaScript, jako např.

- funkce pro zobrazení a skrytí blokového elementu (použito u přihlašovacího formuláře),
- funkci, která zajistí průhlednost obrázků PNG i v prohlížeči Internet Explorer (standardně v tomto prohlížeči průhlednost PNG nefunguje),
- funkce, které implementují technologii AJAX (popsanou níže viz 5.3), použitou při hlasování v anketě.

5.3 Popis implementace některých částí

AJAX

AJAX (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí. Podobně jako např. DHTML, AJAX ve skutečnosti není konkrétní jednotlivá technologie, ale pojem označující použití několika technologií dohromady s určitým cílem.

V případě tohoto informačního systému je technologie AJAX použita při hlasování v anketě. Je implementována pomocí několika kroků:

1. AJAX začíná kliknutím na odkaz některé z odpovědí. V tom okamžiku je volána JavaScriptová funkce `anketa_hlasovat()` s parametrem udávajícím číslo odpovědi.
2. Funkce `anketa_hlasovat()` není nijak složitá, obsahuje volání funkce pro odeslání požadavku serveru. Pokud tato proběhne úspěšně, technologie AJAX je vykonána, v opačném případě se vykonávání JavaScriptu přeručí a přechází se na hlasování bez použití AJAXu.
3. Funkce pro odeslání požadavku serveru má název `send_xmlhttprequest()`. Je navržena tak, aby mohla být použita obecně všemi prvky, které by chtěly využívat AJAXu. Může být volána s celkem 5-ti parametry, z nichž při hlasování v anketě se používají pouze 3. Jsou to:
 - Název funkce zajišťující obsluhu při změně stavu požadavku. Tato funkce dostane parametr s výsledným objektem (XMLHttpRequest), který pošle server.
 - Název metody (GET, POST, ...) kterou se předávají parametry. V našem případě je to parametr udávající číslo odpovědi předávaný metodou GET.
 - URL požadavku, které má server zpracovat. Toto URL obsahuje již zmiňovanou hodnotu udávající číslo odpovědi.

Po zpracování serverem je volána zadaná funkce pro obsluhu odpovědi serveru.

4. Funkce pro obsluhu má název `anketa_obsluha()` a její činnost spočívá v ověření, že vrácená hodnota serveru znamená správné provedení požadavku a následné vložení odpovědi serveru do vnitřku XHTML kódu do oblasti, kde se nachází anketa.

To vše má za následek efektivní dynamickou změnu vzhledu ankety bez načtení (chcete-li „probliknutí“) celé stránky.

Nutno podotknout, že tato technologie je závislá na JavaScriptu, pokud tedy uživatel nemá povolen JavaScript spouštět, hlasování se provede běžným způsobem přes standardní odkaz a znovunačtení celé stránky.

TinyMCE

TinyMCE je název volně dostupného WYSIWYG editoru (viz kapitola 3.1) přístupného na internetu na adrese [6]. V IS je používán uživateli při vytváření příspěvků, zadávání aktualit atd.

TinyMCE je šířen jako tzv. open source, což znamená, že jsou dostupné jeho zdrojové kódy a může být modifikován. Pro použití v informačním systému byla jeho modifikace nezbytná. V originální podobě totiž produkuje kód, který není validní podle normy XHTML 1.0 Strict, podle které se mají stránky informačního systému řídit. Vzhledem k rozsáhlosti kódů, které TinyMCE tvoří, a z ní vyplývající jejich pochopení, zabrala modifikace poměrně dost času.

Kromě produkování validního kódu byl mírně upraven i vzhled editoru.

Identifikace uživatele

Každý nepřihlášený uživatel je identifikován na základě jeho IP adresy. Problém nastává, pokud mají 2 nebo více uživatelů resp. jejich počítače stejnou IP adresu. To může být způsobeno pokud pro přístup k internetu používají *proxy server*, což je server počítačové sítě, který umožňuje obecně 2 a více klientům nepřímé připojení k internetu. Funguje jako prostředník mezi klientem a cílovým serverem na internetu, překládá klientské požadavky a vůči cílovému serveru vystupuje jako klient. Přijatou odpověď následně odesílá zpět na klienta. To znamená, že všechny počítače používající proxy server vystupují na internetu jako počítač jediný.

Snaha identifikovat i tyto uživatele za proxy serverem spočívá ve zjištění jejich původní IP adresy, pod kterou vystupovali před vstupem na proxy server. Tato adresa je zpravidla ukládána do hlavičky zprávy požadavku klienta.

Pokud taková IP adresa za proxy serverem existuje, je v jazyce PHP dostupná v globálním poli `$_SERVER`, které obsahuje několik názvů indexů, kde může být proxy IP adresa uložena.

Identifikace nepřihlášeného uživatele v IS spočívá ve vytvoření a uložení textového řetězce, který se skládá z aktuální IP adresy uživatele doplněného o jeho IP adresu za proxy serverem, pokud existuje. Kód vytvoření řetězce je následující:

```
$ipadresa = $_SERVER['REMOTE_ADDR'];
if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){
    $ipadresa .= @"\.$_SERVER['HTTP_X_FORWARDED_FOR'];
}
if(isset($_SERVER['HTTP_FORWARDED'])){
    $ipadresa .= @"\.$_SERVER['HTTP_FORWARDED'];
}
if(isset($_SERVER['HTTP_CLIENT_IP'])){
    $ipadresa .= @"\.$_SERVER['HTTP_CLIENT_IP'];
}
if(isset($_SERVER['X_HTTP_FORWARDED_FOR'])){
    $ipadresa .= @"\.$_SERVER['X_HTTP_FORWARDED_FOR'];
}
```

Jak je vidět, jednotlivé ip adresy jsou v řetězci odděleny znakem @.

Kapitola 6

Testování

Testování probíhalo převážně v průběhu vytváření aplikace a obnášelo ověřování funkčnosti každé nově přidané funkcionality systému. Během tohoto procesu jsem proto testoval správné chování nových prvků ve všech stavech, do kterých se mohou dostat. Největší zaměření bylo na extrémní (nejkritičtější) stavy. Nově přidané funkce systému mohly ovlivnit chování těch, které byly již dříve v této fázi testovány. Z tohoto důvodu jsem s každou nově přidanou funkcionalitou, která aspoň z části ovlivňovala již ověřené funkce, tyto znovu testoval.

Systém byl také testován z hlediska bezpečnosti. Vyzkoušel jsem řadu na internetu popsaných metod SQL injection pro proniknutí do databáze a neoprávněnému přístupu k systému. Ani jedna z vyzkoušených metod nevedla k úspěšnému proniknutí do databáze. Navíc mou velkou výhodou oproti možným útočníkům byla znalost struktury databáze, jejích tabulek a použitých datových typů. V praxi útočník takové informace nemá a průnik do databáze je pro něj mnohem složitější. Systém prokázal odolnost proti SQL injection útokům.

Dalším možným bezpečnostním rizikem je předávání parametrů mezi skripty v rámci URL. Hodnoty parametrů jsou snadno upravitelné a jejich vhodná úprava u nezabezpečených stránek může vést k nepředpokládanému chování systému. Všechny skripty, kterým jsou data předávána v rámci URL, obsahují kontrolu správného tvaru parametrů. Systém na zadání chybných hodnot reagoval korektně – nedostal se do nepředpokládaného stavu.

Při testování nebyly objeveny žádné chyby nebo nepředpokládané stavy systému.

Kapitola 7

Závěr

Výsledkem této práce je funkční webový informační systém, který slouží k prezentaci a správě vnitřního chodu obecného nohejbalového klubu. Doposud jsem nenašel žádný jiný informační systém, který by obsahoval podobné uživatelské role a funkcionality. Také díky tomuto faktu je tento systém, co se týče oblasti působnosti, jeden z prvních svého druhu.

Systém do jisté míry splňuje všechny předem stanovené požadavky. Mezi jeho silné části patří úhledný grafický vzhled a z něho plynoucí přehlednost a srozumitelnost ovládání. Dalším výrazným pozitivem je důraz na přístupnost. Systém se správně zobrazuje na všech běžně používaných prohlížečích, umožňuje zvětšení textu pro slabozraké bez narušení designu, veškerá jeho funkčnost je zachována i při vypnutém JavaScriptu a celý systém je validní podle normy XHTML 1.0 Strict, díky čemuž má větší předpoklady pro správné zobrazení i uživatelům s nestandardními prohlížeči.

Mezi průměrné části informačního systému patří například jeho optimalizace pro internetové vyhledávače. Při vývoji každého jednotlivého prvku systému na ni byl brán ohled, avšak doposud nebylo přistoupeno k optimalizaci pro vyhledávače jako k jednomu celku za účelem vykonat patřičné změny systému. Tato optimalizace je dalším možným krokem vývoje projektu.

Méně silnou částí jsou některé prvky zabezpečení systému. Jedná se především o šifrování a z části o hašování (viz kapitola 4.4), které zajišťují utajení a integritu systému. Jak jsem již zmínil, vzhledem k tomu, že se v této práci jedná o aplikaci menšího rozsahu, rozhodl jsem se šifrování nepoužít. Ponechávám však možnost implementace šifrování v rámci dalšího rozvoje této aplikace. Za pomoci druhé zmíněné technologie – hašování – by se dala zdokonalit integrita systému použitím hašování na více citlivých údajů v databázi. Aplikace používá pouze hašování hesla.

Jelikož byl IS vyvíjen jednou osobou, nebylo třeba klást přehnaný důraz na oddělení aplikační a prezenční vrstvy. Z hlediska dalšího vývoje projektu by bylo vhodné do aplikace začlenit některé ze šablonovacích systémů, jako např. SMARTY.

Celá práce pro mne byla velkým přínosem, rozšířil jsem si své znalosti nejen v oblasti technologií pro vývoj informačních systémů, ale i ve všech oblastech jiných, které projekt zahrnuje. Má práce na vývoji aplikace bude pokračovat dále, po nasazení informačního systému do provozu budu zpětnou vazbou zjišťovat možná rozšíření nebo upravení jeho funkcionality.

Seznam použitých zdrojů

- [1] *KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV)* [online]. Praha: Národní knihovna České republiky, 2003. [cit. 18. 4. 2007]. Dostupné z: <http://sigma.nkp.cz/cze/ktd>.
- [2] *Interval.cz – webdesign a e-komerce denně* [online]. 2007. [cit. 1. 5. 2007]. Dostupné z: <http://interval.cz/>.
- [3] *Microsoft .NET. Definice základních součástí .NET* [online]. 2007. [cit. 30. 4. 2007]. Dostupné z: <http://www.microsoft.com/cze/net/basics/whatis.asp>.
- [4] *Netcraft* [online]. 2007. [cit. 3. 5. 2007]. Dostupné z: <http://netcraft.com/>.
- [5] *PHP: Hypertext Preprocessor* [online]. 2007. [cit. 1. 5. 2007]. Dostupné z: <http://php.net/>.
- [6] *TinyMCE Javascript Content Editor by Moxiecode Systems AB* [online]. 2007. [cit. 4. 5. 2007]. Dostupné z: <http://tinymce.moxiecode.com/>.
- [7] *World Wide Web Consortium* [online]. 2007. [cit. 2. 5. 2007]. Dostupné z: <http://www.w3.org/>.
- [8] *Wikipedie: Otevřená encyklopedie* [online]. 2007. [cit. 20. 4. 2007]. Dostupné z: <http://cs.wikipedia.org>.
- [9] Druska, P. *CSS a XHTML: tvorba dokonalých webových stránek krok za krokem*. 1. vyd. Praha: Grada Publishing, a.s., 2006. 200 s. ISBN 80-247-1382-9.
- [10] Janovský, D. *Jak psát web* [online]. 2007. [cit. 27. 4. 2007]. Dostupné z: <http://www.jakpsatweb.cz/>.
- [11] Pokorný, J. *Databázové systémy a jejich použití v informačních systémech*. 1. vyd. Praha: Academia, 1992. 313 s. ISBN 80-200-0177-8.

Seznam použitých zkratek a symbolů

AJAX Asynchronous JavaScript and XML	POP3 Post Office Protocol version 3
ASCII American Standard Code for Information Interchange	SEO Search Engine Optimization
ASP Active Server Pages	SHA Secure Hash Algorithm 1
BSD Berkeley Software Distribution	SQL Structured Query Language
CSS Cascading Style Sheets	SMTP Simple Mail Transfer Protocol
DHTML Dynamic Hypertext Markup Language	SŘBD Systém Řízení Báze Dat
DOM Document Object Model	SSL Secure Sockets Layer
ECMA European Computer Manufacturers Association	UCS Universal Character Set
ER Diagram Entity Relationship Diagram	URL Uniform Resource Locator
FTP File Transfer Protocol	UTF UCS/Unicode Transformation Format
GIF Graphics Interchange Format	VBScript Visual Basic Script
GUI Grafické Uživatelské Rozhraní	W3C World Wide Web Consortium
HTML Hypertext Markup Language	WAP Wireless Application Protocol
HTTP Hypertext Transfer Protocol	WYSIWYG What You See Is What You Get
IIS Internet Information Services	WWW World Wide Web
IP address Internet Protocol address	XHTML Extensible Hypertext Markup Language
ISO International Standards Organization	XML Extensible Markup Language
IS Informační Systém	
J2EE Java 2 Enterprise Edition	
JPEG Joint Photographic Experts Group	
JSP JavaServer Pages	
JVM Java Virtual Machine	
MySQL My Structured Query Language	
NCSA National Center for Supercomputing Applications	
OOP Objektivě Orientované Programování	
PDA Personal Digital Asistent	
PHP PHP: Hypertext Preprocessor	
PL/SQL Procedural Language/Structured Query Language	
PNG Portable Network Graphics	

Seznam příloh

- A** Ukázka vzhledu informačního systému
- B** Návod na instalaci systému
- C** Obsah přiloženého CD

Příloha A

Ukázka vzhledu informačního systému

Na obrázku A.1 je možno vidět vzhled aplikace z pohledu administrátora.

The screenshot displays the administrator interface of the website for the Prerovský nohejbalový klub TJ Spartak MSEM. The interface is organized into several sections:

- Header:** Features the club's logo, name, and a navigation menu with items like 'forum', 'udalosti', 'vzkazy', 'upload', 'platby', 'členské příspěvky', 'cestovní doklady', and 'stravné'. The date and time are shown as '10. května 2007, 12:59:06'.
- Left Sidebar (Menu):** Contains sections for 'Aktuality', 'Návštěvní kniha', 'Fotogalerie', 'Kontakt', 'Historie klubu', 'Využití kurtů', 'Sponzoři', 'Odkazy', 'Výsledky' (with sub-items for Men's A, Men's B, and Dorost), and 'Členská základna' (with sub-items for Men's A, Men's B, Dorost, Žáci, and Vedení). It also includes a 'Přihlášení' section showing the user is logged in as 'Administrátor' and a 'počet přístupů: 130' counter.
- Main Content Area:** Displays a list of news articles with titles, dates, and edit/delete buttons. The first article is 'Je jaro' (30. dubna 2007) with the text 'Pokud se podíváte ven, zjistíte, že je jaro.' The second is 'seznamy' (11. dubna 2007) with a list of items. The third is 'Zkouška češtiny' (1. dubna 2007) with a paragraph of text. The fourth is 'zkouška aktuality' (15. března 2007) with the text 'zkouška'.
- Right Sidebar:** Contains a 'KALENDAŘ' for May 2007, 'UDÁLOSTI' (7.5.07 - Státní svátek, 15.5.07 - Odevzdání bakalářské práce), and an 'ANKETA' (poll) asking 'Jak se umístí Prerovský tým v letošní 1. lize?' with a bar chart showing results for various positions.

At the bottom of the page, there is a footer with technical information: '1 online | Vygenerováno za 0.10705 sekund | © 2007, Nohejbal Prerov | autor: Jan Pavelka | Optimalizováno pro IE Explorer 6.0, Mozilla Firefox 1.5, Opera 9.1 a rozlišení 1024x768/24 | Validní XHTML Strict kód | no tables, clear css layout.'

Obrázek A.1: Ukázka vzhledu informačního systému

Příloha B

Návod na instalaci systému

Požadavky na chod informačního systému

Aplikace ke svému fungování vyžaduje webový server, který podporuje scriptovací jazyk PHP, standardně Apache, a dále databázový server MySQL.

Popis instalace v několika krocích

1. Prvním krokem je zvolení vhodného webového serveru spolu s databází.
2. Pokud máme server vybraný, nejprve je vhodné vytvořit databázi a naplnit ji příslušnými tabulkami a hodnotami. K databázi MySQL se nejčastěji přistupuje pomocí webové aplikace phpMyAdmin. V ní je potřeba vybrat příslušnou databázi (pokud ještě neexistuje, je třeba jí vytvořit) a v odkazu „SQL“ spustit SQL dotaz z textového souboru. Textový soubor pro vytvoření tabulek IS má název `sql.txt` a najdeme ho v adresáři `sql`. Před potvrzením provedení dotazu je vhodné zkontrolovat, že pole „znaková sada souboru“ je nastaveno na „utf-8“. Po provedení dotazu uloženého v souboru se v databázi vytvoří vše potřebné.
3. Dalším krokem je nastavení parametrů aplikace, jako jsou přihlašovací údaje k serveru databáze, název databáze, popřípadě proměnné programu, jako čas automatického odhlášení, počet zobrazovaných příspěvků na stranu a další. Toto nastavení se provádí v souboru `!nastaveni.php`, který se nachází v kořenové složce aplikace.
4. Po správném zadání hodnot v souboru `!nastaveni.php` zbývá nahrát obsah aplikace na webový server. Pro správné fungování systému je nezbytné zachovat adresářovou strukturu a soubory v ní. Nahrajeme tedy veškerý obsah kořenového adresáře aplikace beze změny na webový server.
5. Posledním krokem je vytvoření hesla administrátora. Skript pro vytvoření se nachází v adresáři `admincreate`. Spustí se zadáním příslušné adresy URL do prohlížeče s příponou adresáře (např. „`www.adresa.cz/admincreate`“).
6. Tímto je instalace IS dokončena. Jeho správné fungování můžeme vyzkoušet zadáním příslušné adresy serveru do prohlížeče, kam byla aplikace nahrána.

Příloha C

Obsah příloženého CD

MANUAL	- adresář, který obsahuje uživatelskou příručku pro používání informačního systému
SOURCE	- adresář, který obsahuje zdrojové kódy a soubory potřebné pro běh informačního systému
TEXT	- adresář, který obsahuje technickou zprávu bakalářské práce ve formátu .PDF (tj. tento soubor)
TEXT-SOURCE	- adresář, který obsahuje zdrojové kódy a soubory technické zprávy bakalářské práce potřebné pro vytvoření .PDF souboru
install.txt	- návod na instalaci informačního systému
readme.txt	- obsah CD