

UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

Informační systém pro svaz karate  
Ondřej Charvát

Bakalářská práce  
2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej CHARVÁT**  
Osobní číslo: **I07633**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Informační systém pro svaz Karate**  
Zadávací katedra: **Katedra informačních technologií**

### Z á s a d y p r o v y p r a c o v á n í :

V teoretické části bakalářské práce budou představeny technologie sloužící k tvorbě webových aplikací. Dále bude provedena analýza současných informačních systémů pro sportovní svazy a jejich porovnání se systémem pro svaz karate.

Cílem je realizace informačního systému pro svaz karate. Práce bude zaměřena na návrh a implementaci webové aplikace pro svaz karate s využitím relační databáze. Aplikace bude mít dvě části - veřejnou (přístupnou bez hesla, přehled soutěží a výsledků) a interní (přístupnou přes heslo, určenou pro ředitele soutěží a členy sportovní komise). Aplikace bude obsahovat databázi závodníků, soutěží, klubů, výsledků, soutěžních kategorií, vyhodnocení bodů, nominační systém na MČR a kalendář soutěží.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**Základní:**

**DRUSKA, P.** CSS a XHTML - tvorba dokonalých webových stránek krok za krokem. Grada 2006.

**PÍSEK, S.** HTML a XHTML - začínáme programovat. Grada 2003.

**ŠIMŮNEK, M.** SQL kompletní kapesní průvodce. Praha: Grada Publishing, 1999.

<http://www.w3.org/Style/CSS/>.

<http://www.w3.org/TR/html4/>.

<http://www.w3.org/TR/xhtml1/>.

**Doporučená:**

**SATRAPA, P.** Web Design. Neokortex s.r.o. 1997.

**VÁCLAVEK, P.** Javascript - Hotová řešení. Brno: Computer Press, 2003.

**LACKO, L.** Oracle, správa, programování a použití databázového systému. Praha: Computer Press, 2002.

**LONEY, K., THERIAULT, M.** Mistrovství v Oracle. Praha, Computer Press, 2002.

<http://interval.cz/>.

<http://www.jakpsatweb.cz/>

<http://www.linuxsoft.cz/php>

Vedoucí bakalářské práce:

**prof. Ing. Karel Šotek, CSc.**  
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **Prohlášení autora**

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10.5.2010

Ondřej Charvát

## **Poděkování**

Tímto bych rád poděkoval prof. Ing. Karlovi Šotkovi, CSc. za poskytnutí cenných rad a připomínek v průběhu vypracování bakalářské práce.

## **Anotace**

Tato práce se zabývá problematikou tvorby webových aplikací a možností jejich uplatnění pro potřeby sportovních svazů – uložení databáze závodníků, výsledků, soutěží, sestavení žebříčků, atd.

V praktické části práce je vytvořen informační systém pro svaz karate, jsou popsány jeho hlavní funkce a způsob implementace.

## **Klíčová slova**

webové aplikace, informační systémy, karate, sportovní svazy

## **Title**

Information system for Karate association

## **Annotation**

This paper studies the creating of the web applications and their importance for the sports associations' needs - storing the database of competitors, results, competitions and building charts.

The practical part presents programming of the information system for Karate Association and describes its main functions and implementations.

## **Keywords**

Web applications, information systems, karate, sport associations

# Obsah

<b>SEZNAM ZKRATEK .....</b>	<b>9</b>
<b>SEZNAM OBRÁZKŮ.....</b>	<b>10</b>
<b>SEZNAM TABULEK .....</b>	<b>10</b>
<b>1 ÚVOD.....</b>	<b>11</b>
<b>2 TVORBA WEBOVÝCH APLIKACÍ.....</b>	<b>12</b>
2.1 WEBOVÁ APLIKACE.....	12
2.2 WEB SERVER .....	12
2.2.1 <i>Vlastnosti web serveru .....</i>	<i>13</i>
2.2.2 <i>Software zabezpečující službu web serveru .....</i>	<i>13</i>
2.3 ZNAČKOVACÍ JAZYKY .....	13
2.3.1 <i>SGML.....</i>	<i>13</i>
2.3.2 <i>HTML.....</i>	<i>13</i>
2.3.3 <i>XML.....</i>	<i>14</i>
2.3.4 <i>XHTML.....</i>	<i>14</i>
2.4 PROGRAMOVACÍ JAZYKY .....	14
2.4.1 <i>PHP.....</i>	<i>14</i>
2.4.2 <i>ASP.....</i>	<i>15</i>
2.4.3 <i>JAVASCRIPT.....</i>	<i>15</i>
2.5 DATABÁZOVÉ SYSTÉMY.....	15
2.5.1 <i>Oracle database.....</i>	<i>15</i>
2.5.2 <i>MySQL .....</i>	<i>16</i>
2.5.3 <i>Firebird .....</i>	<i>16</i>
2.5.4 <i>PostgreSQL .....</i>	<i>16</i>
2.6 DESIGN WEBOVÉ APLIKACE A CSS .....	16
2.6.1 <i>Předchůdci kaskádových stylů .....</i>	<i>17</i>
2.6.2 <i>Kaskádové styly.....</i>	<i>17</i>
2.7 SEO OPTIMALIZACE .....	17
2.7.1 <i>Validita zdrojových kódů .....</i>	<i>18</i>
2.7.2 <i>Přehled některých metod SEO optimalizace .....</i>	<i>18</i>
2.8 BEZPEČNOST WEBOVÝCH APLIKACÍ.....	19
2.8.1 <i>Autentizace a autorizace uživatelů .....</i>	<i>19</i>
2.8.2 <i>Šifrování .....</i>	<i>19</i>
2.8.3 <i>Bezpečnostní mezery.....</i>	<i>19</i>
2.8.4 <i>SQL injection .....</i>	<i>19</i>
2.9 VOLBA VHODNÝCH TECHNOLOGIÍ .....	20
<b>3 INFORMAČNÍ SYSTÉMY PRO SPORTOVNÍ SVAZY .....</b>	<b>21</b>
3.1 VÝZNAM IS PRO ČINNOST SPORTOVNÍCH SVAZŮ .....	21
3.2 POŽADAVKY NA IS.....	21
3.3 SOUČASNÝ STAV .....	22
3.3.1 <i>Svazy karate.....</i>	<i>22</i>
3.3.2 <i>Ostatní sportovní svazy .....</i>	<i>22</i>

<b>4</b>	<b>INFORMAČNÍ SYSTÉM PRO SVAZ KARATE .....</b>	<b>23</b>
4.1	POŽADAVKY .....	23
4.1.1	<i>Vyhodnocení bodování pomocí koeficientů .....</i>	23
4.1.2	<i>Skupiny soutěží .....</i>	23
4.1.3	<i>Role uživatelů.....</i>	23
4.1.4	<i>Další požadavky na systém.....</i>	24
4.2	VOLBA TECHNOLOGIÍ .....	24
4.3	ROZVRSTVENÍ INFORMAČNÍHO SYSTÉMU .....	25
4.4	NÁVRH DATABÁZE .....	25
4.4.1	<i>E-R diagram.....</i>	26
4.4.2	<i>Popis tabulek a jejich atributů .....</i>	26
4.4.3	<i>Popis a syntaxe použitých pohledů .....</i>	32
4.4.4	<i>Indexy.....</i>	33
4.4.5	<i>Sekvence.....</i>	34
4.4.6	<i>Funkce .....</i>	35
4.4.7	<i>Triggery a referenční integrita dat.....</i>	36
4.5	NÁVRH WEBOVÉ APLIKACE .....	37
4.5.1	<i>Architektura aplikace .....</i>	37
4.5.2	<i>Zabezpečení.....</i>	38
4.5.3	<i>Integrita dat.....</i>	40
4.5.4	<i>Syntaxe přístupu k databázi.....</i>	42
4.5.5	<i>Administrační rozhraní.....</i>	43
4.5.6	<i>Rozhraní běžného uživatele .....</i>	44
4.5.7	<i>Rozvržení a vzhled.....</i>	44
4.6	NÁSTROJE POUŽITÉ PŘI REALIZACI PROJEKTU .....	45
<b>5</b>	<b>ZÁVĚR .....</b>	<b>46</b>
	<b>POUŽITÁ LITERATURA .....</b>	<b>47</b>
	<b>PŘÍLOHA A – UKÁZKA – BĚŽNÝ UŽIVATEL – VYHLEDÁNÍ ZÁVODNÍKA .....</b>	<b>49</b>
	<b>PŘÍLOHA B – UKÁZKA – BĚŽNÝ UŽIVATEL – SESTAVENÍ ŽEBŘÍČKU ZÁVODNÍKŮ .....</b>	<b>50</b>
	<b>PŘÍLOHA C – UKÁZKA – ADMINISTRAČNÍ ROZHRAŇÍ .....</b>	<b>51</b>



## Seznam zkratek

ASP	Active Server Pages
BSD	Berkeley Software Distribution (též Berkeley Unix)
CSS	Cascading Style Sheets
DDL	Data Definition Language
DML	Data Modification Language
DTD	Document Type Definition
ERD	Entity-Relationship Diagram
GNU	GNU's Not Unix (projekt zaměřený na svobodný software, inspirovaný operačními systémy unixového typu)
GPL	General Public License
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IS	Information System
ISO	International Organization for Standardization
Mac	Macintosh
OS	Operating System
PHP	Hypertext Preprocessor
PL/SQL	Procedural Language / Structured Query Language
RDBMS	Relation DataBase Management System
SEO	Search Engine Optimization
SGML	Standard Generalized Markup Language
SSL	Secure Sockets Layer
STK	Sportovně Technická Komise
SQL	Structured Query Language
UDF	User Defined Function
WML	Wireless Markup Language
WYSIWYG	What You See, Is What You Get
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

## Seznam obrázků

Obrázek 2.1 - Web server.....	12
Obrázek 4.1 - Use case diagram .....	24
Obrázek 4.2 - Rozvrstvení informačního systému .....	25
Obrázek 4.3 - E-R diagram .....	26

## Seznam tabulek

Tabulka 4.1- Bodové hodnocení.....	23
Tabulka 4.2 - Popis tabulky závodník.....	27
Tabulka 4.3 - Popis tabulky klub .....	27
Tabulka 4.4 - Popis tabulky soutěž.....	28
Tabulka 4.5 - popis tabulky kategorie.....	28
Tabulka 4.6 - Popis tabulky adresa.....	29
Tabulka 4.7 - Popis tabulky skupina.....	29
Tabulka 4.8 - Popis tabulky město .....	29
Tabulka 4.9 - Popis tabulky soutěží.....	30
Tabulka 4.10 - Popis tabulky reprezentace .....	30
Tabulka 4.11 - Popis tabulky umístění .....	30
Tabulka 4.12 - Popis tabulky koeficienty .....	31
Tabulka 4.13 - Popis tabulky body .....	31
Tabulka 4.14 - Popis tabulky SoutezSkupina.....	31
Tabulka 4.15 - Popis tabulky uživatel.....	32
Tabulka 4.16 - Popis tabulky informace .....	32
Tabulka 4.17 - Přehled použitých pohledů.....	33
Tabulka 4.18 - Přehled použitých indexů .....	34
Tabulka 4.19 - Přehled použitých sekvencí .....	35
Tabulka 4.20 - Přehled použitých triggerů.....	37
Tabulka 4.21 - Nástroje použité při realizaci projektu .....	45

## 1 Úvod

Tato bakalářská práce se zabývá problematikou tvorby webových aplikací. Cílem práce je vytvoření informačního systému pro sportovní svaz, konkrétně pro svaz Karate.

V kapitole tvorba webových aplikací je vypracován přehled některých technologií používaných při tvorbě webových aplikací.

V kapitole informační systémy pro sportovní svazy je zdůrazněn význam informačních systémů pro sportovní svazy. Jsou zde shrnuty některé základní požadavky na tyto systémy. Je provedeno ohodnocení současného stavu.

V kapitole informační systém pro svaz karate je popsána implementace a základní funkce vytvořeného informačního systému.

## 2 Tvorba webových aplikací

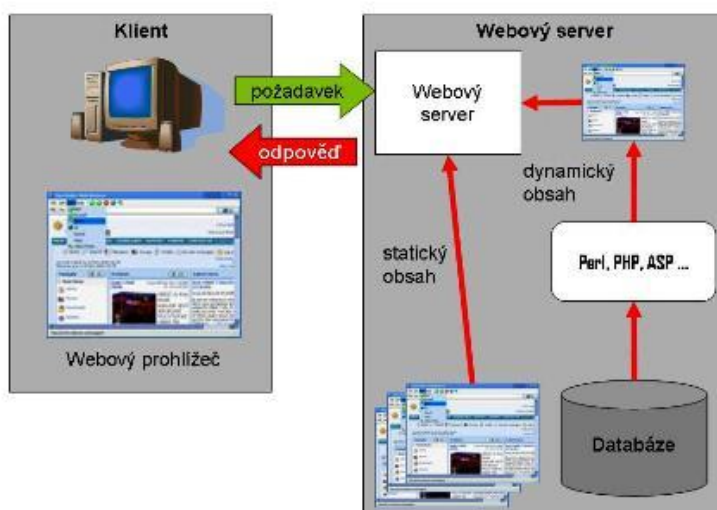
### 2.1 Webová aplikace

V dnešní době se zrychlujícím se životním tempem jsou kladeny vysoké požadavky na rychlost výměny informací a možnost jejich předávání mezi velmi vzdálenými místy. Vhodným řešením jsou stále populárnější webové aplikace, které mohou být spuštěny na velkém množství zařízení, jako jsou osobní počítače, notebooky, mobilní telefony, kapesní počítače a mnoho dalších. Další výhodou webových aplikací je při dodržení celosvětově uznávaných norem a standardů možnost jejich spuštění na libovolném operačním systému jen za pomoci některého webového prohlížeče.

Webové aplikace můžeme rozdělit podle několika různých kritérií. Jedním z možných dělení je například na statické a dynamické. Statické webové aplikace se ve většině případů vyznačují svojí jednoduchostí a výrazně menšími možnostmi. Typickým příkladem je jednoduchá webová prezentace firmy na internetu bez administračního rozhraní. U této aplikace web server pouze předá statický obsah. Dynamické aplikace vyžadují mnohem větší znalost problematiky tvorby webových aplikací. Zpracování obsahu je provedeno na web serveru a uživateli je zaslán výsledek. Ve většině případů spolupracují s nějakým databázovým serverem. Typickým příkladem je internetový obchod, redakční, blogovací nebo sportovní informační systém [1].

### 2.2 Web server

Jedná se o počítač vyřizující HTTP požadavky od klientů, kterými jsou programy zvané webové prohlížeče, a také se jedná o software, který na tomto počítači běží [2]. Způsob práce web serveru je zobrazen na obr. 2.1



Obrázek 2.1 - Web server

### **2.2.1 Vlastnosti web serveru**

Webové servery se mohou v jednotlivých parametrech značně lišit, mají však několik společných vlastností. Každý webový server je připojen k počítačové síti a přijímá požadavky klientů ve tvaru HTTP. Na tyto požadavky je serverem zaslána odpověď, součástí odpovědi je stavový kód. Tím je jasně dáno, zda všechno proběhlo v pořádku, nebo došlo k nějakým problémům. Stavový kód označující, že vše proběhlo bez problémů, je 200. Kódy 300 – 399 jsou přiřazeny problémům se spojením, kódy 400 – 499 chybám souvisejícím s vyřízením požadavku, např. nedostupnost stránky, kódy 500 – 599 jsou přiřazeny interním chybám serveru [2].

### **2.2.2 Software zabezpečující službu web serveru**

Programů zabezpečujících službu webového serveru je na trhu k dostání celá řada. Mezi tři nejvýznamnější patří: Apache HTTP server, Internet Information Services a Sun Java System Web Server [2].

Apache HTTP server je v současné době nejrozšířenějším webovým serverem. Jeho hlavní výhodou je, že je šířen s otevřeným kódem a je možné ho spustit na mnoha platformách, jako např.: GNU/Linux, BSD, Solaris, Mac OS X, Microsoft Windows a další [3].

## **2.3 Značkovací jazyky**

Historie značkovacích jazyků začala zavedením jazyka SGML, pokračovala několika generacemi HTML a jejich rozšířeními, až došla k XML. Posledním článkem série značkovacích jazyků je XHTML [4].

### **2.3.1 SGML**

Jazyk SGML byl schválen v roce 1986 podle standardu ISO. Je to předchůdce jazyka HTML. Standard je používán obchodními organizacemi po celém světě k uveřejňování dokumentů obsahujících text, odstavce, nadpisy různých úrovní, multimediální prvky a několik formátovacích prvků.

SGML se v původní podobě moc nerozšířil, je však používán i v dnešní době. Jeho velkou předností je možnost definice nových elementů a atributů, proto je v současné době používán jako systém pro tvorbu nových značkovacích jazyků, jako např. HTML a XML.

Každý značkovací jazyk vytvořený pomocí SGML je definován pomocí DTD. DTD definují formátovací značky a jejich vlastnosti a hodnoty, které lze v jazyce používat. Spojením vytvořeného dokumentu s určitou DTD se jednoznačně určí, které formátovací značky se mohou na stránce vyskytovat [4].

### **2.3.2 HTML**

HyperText Markup Language je značkovací jazyk definovaný v rámci SGML. Dokumenty napsané jazykem HTML mohou obsahovat hypertextové odkazy a pokročilejší formátování.

HTML poskytuje prvky, pomocí nichž lze vytvářet, formátovat a upravovat webové stránky. Prostřednictvím tohoto jazyka je možné pracovat s: textem, odkazy, obrázky, animacemi, ... [4]

### **2.3.3 XML**

Extensible Markup Language je podmnožinou SGML. Umožňuje napsat vlastní definici dokumentu (DTD) a vytvořit tak vlastní značkovací jazyk pro různé účely a různé typy dat. Jazyk je určen pro výměnu dat mezi aplikacemi a publikování dokumentů. Nezabývá se vzhledem, který může být definován např. pomocí kaskádových stylů CSS [5].

### **2.3.4 XHTML**

Extensible HyperText Markup Language je XML, jehož definice typu dokumentu obsahuje HTML. Jedná se v podstatě o vhodnou kombinaci nejlepších vlastností XML a HTML, která vznikla přeformulováním HTML 4 do aplikace XML. Z toho vyplývá možnost vytváření webových dokumentů v jazyce HTML stejně jako dříve s možností jednoduché definice vlastních formátovacích značek.

Na rozdíl od HTML je jazyk XHTML case sensitivní, tzn. že rozlišuje velká a malá písmena. Všechny formátovací značky musí být malými písmeny, všechny párové i nepárové značky musí být ukončeny, hodnoty atributů musí být v uvozovkách, kód stránky nesmí obsahovat znak &, definice kaskádových stylů CSS by měly být umístěny v externích souborech [4].

## **2.4 Programovací jazyky**

Programovací jazyky jsou jazyky sloužící k tvorbě počítačových programů. Programovací jazyky používané při tvorbě webových aplikací mohou být rozděleny do dvou skupin: běžící na serveru a běžící až v internetovém prohlížeči klienta, které mohou z bezpečnostních důvodů ovlivňovat jen internetovou stránku, nikoliv klientský počítač [6].

### **2.4.1 PHP**

PHP je skriptovací programovací jazyk, určený především pro tvorbu dynamických webových aplikací. Nejčastěji je začleňován přímo do struktury jazyka HTML, XHTML nebo WML. PHP je však možné použít i při tvorbě desktopových a konsolových aplikací.

Skripty jsou prováděny většinou na serveru, klientovi je zaslán výsledek jejich činnosti. Syntaxe jazyka je inspirována jazyky Perl, C, Pascal a Java. PHP je velice oblíbeným programovacím jazykem webových aplikací, především kvůli své jednoduchosti použití, nezávislosti na platformách a možnosti přístupu prostřednictvím knihoven k většině databázových systémů. PHP je otevřený projekt s rozsáhlou podporou komunity [7].

### **2.4.2 ASP**

Active Server Pages je skriptovací jazyk od společnosti Microsoft. Jedná se o skriptovací jazyk spouštěný na serveru, stejně jako jazyk PHP. Oproti již zmiňovanému PHP má složitější syntaxi, je omezený pouze na servery na platformě Windows a má horší práci s objekty. Princip tvorby webových aplikací prostřednictvím tohoto jazyka je podobný tvorbě prostřednictvím jazyka PHP, kvůli zmiňovaným nevýhodám je však ASP méně využíváno [8].

### **2.4.3 JAVASCRIPT**

JavaScript je programovací jazyk vyvinutý společnostmi Netscape a Sun Microsystems. Je to multiplatformní, objektově orientovaný skriptovací jazyk, který je dnes zpravidla používán jako interpretovaný programovací jazyk pro webové stránky. Na rozdíl od zmiňovaných jazyků ASP a PHP je JavaScript spouštěn na straně klienta, tzn. že je zpravidla stažen spolu s obsahem stránky do prohlížeče a poté spuštěn. Jazyk je obvykle použit k ovládní různých interaktivních prvků GUI nebo k vytvoření animací, efektů obrázků atd. JavaScript je možné spustit i na straně serveru [6].

## **2.5 Databázové systémy**

Databázový systém je definován jako sloučení dat a nástrojů, pomocí kterých jsou tato data vytvářena, aktualizována a rušena. Kromě nástrojů, které již byly zmíněny, musí každý databázový systém obsahovat nástroje pro definici struktur dat, definici a zajištění integrity dat a zajištění fyzické a logické nezávislosti dat. Případně může dále obsahovat nástroje pro zálohování dat a nástroje pro podporu více uživatelů, zejména definici transakcí a přístupových práv.

Již zmíněná fyzická nezávislost dat je definována jako oddělení způsobu fyzického uložení dat, např. na pevném disku, od způsobu práce s nimi. O logické nezávislosti dat se hovoří tehdy, když změna logické struktury dat, její rozšíření o další tabulky nebo sloupce v existující tabulce, nevyžaduje úpravu již existujících programů nebo dotazů pracujících s daty. Pojem integrita dat je definován tak, že data věrně zobrazují reálný stav, který je těmito daty zobrazován. Důvodem nekonzistence mezi daty a realitou může být nedostatečná aktualizace dat, nebo nedostatečné zachování referenční integrity při mazání a úpravě dat [9].

### **2.5.1 Oracle database**

Oracle Database je databázový systém od společnosti Oracle Corporation. Jedná se o moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou rozšiřitelností. Aktuální verzí je Oracle Database 11g. V systému je zabudována podpora pro standardní relační dotazovací jazyk SQL, proprietární rozšíření společnosti Oracle pro hierarchické dotazy, podpora objektových databází a databází uložených v hierarchickém modelu. Dále je v systému podporován imperativní programovací jazyk PL/SQL, pomocí kterého jsou možnosti Oracle Database rozšířeny o tvorbu uživatelských procedur, funkcí, programových balíků a trigerrů. Systém

dále obsahuje širokou paletu nástrojů pro podporu snadného nasazení na gridových sítích [10].

### **2.5.2 MySQL**

MySQL je multiplatformní databázový systém, který je považován za úspěšného průkopníka dvojího licencování. Je šířen jak pod volnou licenci GPL, tak pod komerční placenou licenci. Komunikace s databází probíhá v dialektu jazyka SQL, který obsahuje některá rozšíření proti samotnému jazyku SQL.

Díky tomu, že se jedná o volně šiřitelný systém s vysokým výkonem a snadnou implementací na různé operační systémy, je MySQL v současné době hojně používáno při tvorbě webových aplikací. Systém má však několik slabších míst, jež jsou postupem času odstraňována. Mezi tyto slabiny může být zařazeno zálohování, práce s pohledy, triggerem a uloženými procedurami [11].

### **2.5.3 Firebird**

Firebird nebo také FirebirdSQL je stejně jako MySQL multiplatformní relační databázový systém. Jedná se o fork neboli alternativní větev programu databázového systému InterBase od společnosti Borland. Výhodou tohoto systému je široká podpora vývojových nástrojů, především Delphi, .NET a Java. Mezi další výhody systému může být zařazeno především jednoduché zálohování, jazyk PL/SQL na dobré úrovni a možnost tvorby vlastních UDF neboli uživatelsky definovaných funkcí [12].

### **2.5.4 PostgreSQL**

PostgreSQL je objektově-relační open source databázový systém. Primárně je vyvíjen pro Unixové systémy, existují však i balíčky pro platformu win32. Tento databázový systém získal řadu prestižních ocenění, především za dobré možnosti práce s funkcemi, triggerem, indexy, za možnosti při definici vlastních objektů a možnosti používání dědičnosti při tvorbě tabulek. Modifikovanou verzí tohoto systému používá například společnost Yahoo k ukládání a analýze chování uživatelů na internetu. Další uživatelem PostgreSQL je komunitní server MySpace [13].

## **2.6 Design webové aplikace a CSS**

Při návrhu designu webové aplikace by měla být dodržena některá základní pravidla, protože špatně navržený design je častou příčinou neúspěchu webové aplikace. Webová aplikace by měla být co nejpřehlednější, požadované informace by měly být podávány v takové podobě, ve které je zachována souvislost, a měly by být přehledně zobrazeny. Aplikace by měla dodržovat jednotný přehledný styl s citlivě zvolenými barvami a rozložením obsahu a ovládacích prvků. Celkově by měla být navržena co nejvíce intuitivně a co nejpohodlněji pro uživatele.



### **2.6.1 Předchůdci kaskádových stylů**

V současné době jsou nejvíce používaným způsobem designu webových aplikací kaskádové styly, neboli CSS. V některých, ve většině případů starších, webových aplikacích může být použit i jiný způsob tvorby designu aplikace.

Jednou z možností, jak vytvořit design webové aplikace, je rozdělení stránky na části za pomoci HTML tabulek. Použitím vlastností HTML tabulek, jako je například sloučení buněk tabulky v řádku nebo ve sloupci, je vytvořeno jednoduché rozložení stránky. Do příslušných buněk tabulky je poté vložen požadovaný obsah, např. menu, hlavička a patka stránky, část se samotným obsahem,...

Další možností, jak rozvrhnout webovou aplikaci, je použití ráků neboli framů. Webová stránka je rozdělena na obdélníkové oblasti, framy, do kterých jsou načteny příslušné HTML stránky obsahující menu, hlavičku a patku stránky, samotný obsah,.... Tato technika byla v dřívější době poměrně populární. Postupně bylo od tohoto způsobu rozložení ustoupeno, protože s sebou přináší jistá úskalí. Jednotlivé HTML soubory, použité při této technice, jsou špatně zpracovatelné vyhledávacími roboty na internetu a může dojít ke snížení tzv. page rank, čímž je samozřejmě zhoršena pozice takové webové stránky ve vyhledávači.

### **2.6.2 Kaskádové styly**

Kaskádové styly neboli CSS jsou jazyk pro popis zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Je to v současné době nejpoužívanější způsob tvorby designu webových aplikací. Hlavním smyslem CSS je umožnění oddělení návrhu designu dokumentu od samotné struktury a obsahu. Tento koncept je žádoucí zejména z hlediska zpracování dokumentů a vyhledávání informací.

Kaskádové styly jsou založeny na principu uložení pravidel a jejich vrstvení, kde platí, že platná je vždy až poslední definice. Takovýmto způsobem tvorby designu webové aplikace je dána spousta výhod, příkladem je možnost rozsáhlejšího formátování, jednodušší údržba designu aplikace, přehlednější struktura zdrojového kódu, oddělení struktury a stylu, možnost definice různých stylů pro různá výstupní zařízení, použití uživatelem definovaných stylů. Nevýhodou kaskádových stylů je různá podpora v různých webových prohlížečích [14].

## **2.7 SEO optimalizace**

SEO neboli Searching Engine Optimization je metodika vytváření a upravování internetových stránek takovým způsobem, aby jejich forma a obsah byly vhodné pro automatizované zpracování v internetových vyhledávačích. Jedná se o komplexní a rozsáhlou problematiku, v této podkapitole je popsán pouze hrubý nástin tohoto tématu.

Internetový vyhledávač za pomoci automatizovaných robotů stahuje v určitém časovém intervalu obsah webových stránek, který je dále analyzován a ohodnocen podle algoritmu vyhledávače. Webovým stránkám jsou poté přiděleny body, které jsou uloženy

a používány při vyhledávání příslušných klíčových slov. Většina používaných metod SEO optimalizace, které byly dříve považovány za výhodu, jsou dnes spíše nutným standardem.

### **2.7.1 Validita zdrojových kódů**

Validnost zdrojových kódů je považována za nejnужnější standard. Webové stránky s nevalidním kódem jsou ohodnoceny velmi nízkým počtem bodů. Nevalidní kód je ve většině případů důvodem nekorektního zobrazení webových stránek v některých prohlížečích, nebo různého zobrazení v různých prohlížečích. Validnost kódu je možné ověřit za pomoci spousty volně dostupných validátorů na internetu.

### **2.7.2 Přehled některých metod SEO optimalizace**

Jednou ze základních metod optimalizace je volba vhodné domény. Při tvorbě webové aplikace jsou na výběr dvě možnosti: využití nějakého freehostingu s doménou třetího řádu nebo zakoupení vlastní domény. První možnost je sice na první pohled jednodušší, ale z hlediska SEO optimalizace není tato volba příliš vhodná. Daleko výhodnější je zakoupení vlastní domény v kombinaci s vhodným a intuitivním názvem domény.

Další metodou je volba vhodná struktury celé webové aplikace. Typickým příkladem častých chyb z hlediska SEO optimalizace jsou parametry předávané odkazem u dynamických aplikací v nečitelném formátu, kde se vyskytují různé kombinace id kategorií a produktů. Nejvhodnější je jasný, na první pohled pochopitelný, strukturovaný a čitelný systém předávání parametrů odkazem, ze kterého uživatel i robot vyhledávače hned zjistí, jaký obsah je cílem odkazu.

Jednou z často zapomínaných metod je tvorba webových aplikací se strukturovaným a přehledným sémantickým kódem, ze kterého je jasně poznat, co je titulek, co nadpis, co je důležitý text. Tato metoda je velice důležitá především kvůli způsobům, jakým roboti vyhledávače analyzují zdrojový kód aplikace.

Dalším způsobem je vhodné používání odkazů. Pokud robot nalezne větší množství odkazů na stránku s nějakým klíčovým slovem, pak předpokládá, že dané téma na této stránce nalezne. Algoritmus vyhodnocování odkazů je v praxi mnohem složitější, do hodnocení je zahrnut počet bodů stránky, ze které tento odkaz vede, jakým textem je na tuto stránku odkazováno, jestli je stránka stejně tématicky zaměřena a mnoho dalších parametrů. Odkazy jsou vyhledávacími roboty chápány jako sbírka referencí na internetovou stránku. Tato vlastnost vyhledávacích robotů je často zneužívána tvorbou falešných zpětných odkazů za účelem zlepšení pozice ve vyhledávači, pokud však robot odhalí takový pokus, je webová stránka potrestána výraznou bodovou srážkou.

Metod SEO optimalizace je celá řada, některé jsou všeobecně známé a některé jsou jen výsledkem odhadu chování vyhledávacích robotů. Spousty metod a vlastností vyhledávacích robotů je zneužíváno ke zlepšení pozice ve vyhledávačích, po odhalení takového pokusu však dojde spíše k výraznému zhoršení pozice [15].

## 2.8 Bezpečnost webových aplikací

Správné zabezpečení je velice důležitou součástí tvorby webových aplikací. Velice často však bývá podceňováno. Vhodně navržená bezpečnostní politika aplikace je určitým kompromisem mezi možnostmi práce uživatele a bezpečností dat. Výsledkem používání vhodně navržené bezpečnostní politiky je zpřístupnění příslušných funkcí a dat příslušným uživatelům aplikace a nikomu jinému.

### 2.8.1 Autentizace a autorizace uživatelů

Pojem autentizace je definován jako prokázání identity nějakého subjektu, obvykle uživatele. Mechanismů použitelných pro autentizaci uživatelů webových aplikací je celá řada. Příkladem je HTTP autentizace, SSL klientský certifikát, vyplnění hesla a jména do formuláře a prostřednictvím poskytovatele identity.

Pojem autorizace je definován jako rozhodnutí o povolení přístupu autentizovaného subjektu. U webových aplikací je většinou autorizace řešena omezením přístupu neautentizovaným uživatelům. Autentizovaným uživatelům je přidělena role, na základě této role je pak rozhodnuto o přístupu do jednotlivých částí aplikace podle potřeb konkrétní aplikace a zvolené bezpečnostní politiky [16].

### 2.8.2 Šifrování

Šifrování může být ve webových aplikacích použito dvojím způsobem, prvním je šifrování přenášených dat a druhým je šifrování uložených dat. Využívá se především v aplikacích, kde jsou používána citlivá data, např. on-line bankovníctví, komunikace mezi obchodními partnery, vzdálený přístup do informačních systémů a jejich dálková správa. V běžných webových aplikacích, které nepracují s tak citlivými daty, je někdy opomenuto šifrování přenosů při autentizaci a šifrování uložených hesel, což může mít v případě zneužití pro aplikaci fatální následky.

Bezpečnost šifrování je založena na tom, že většinou není v reálném čase možné získat klíč a tím přístup k šifrovaným datům. Existuje celá řada šifrovacích algoritmů s různou úrovní bezpečnosti.

### 2.8.3 Bezpečnostní mezery

Zvolené technologie tvorby webových aplikací mohou s sebou přinášet řadu bezpečnostních mezer a chyb. Proto je povinností tvůrce webové aplikace seznámit se s možnými bezpečnostními riziky dané technologie a učinit kroky zabráňující zneužití těchto mezer [16].

### 2.8.4 SQL injection

Nedostatečné zabezpečení proti SQL injection je jednou z častých bezpečnostních chyb při tvorbě webových aplikací. SQL injection je napadení databázové vrstvy aplikace založené na principu podstrčení vlastního kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu.

Na straně aplikace je možné se tomuto útoku bránit jednoduchou kontrolou vstupních dat a případným převedením nebezpečných znaků na bezpečnou sekvenci znaků. Tato funkce je zpravidla prováděna před samotným vykonáním SQL dotazu. Na straně databáze je možné útoku zabránit, nebo ho přinejmenším extrémně ztížit vhodným nastavením práv uživatelského účtu, přes který webová aplikace přistupuje do databáze [17].

## **2.9 Volba vhodných technologií**

Již na začátku před tvorbou návrhu a psáním samotné webové aplikace by měla být důkladně promyšlena volba vhodných technologií a prostředků, které budou použity. Nevhodným výběrem může dojít ke značným komplikacím při tvorbě, úpravách, ale i provozu samotné aplikace. Je třeba se vyvarovat již zmíněným bezpečnostním rizikům a chybám.

## **3 Informační systémy pro sportovní svazy**

### **3.1 Význam IS pro činnost sportovních svazů**

V dnešní době potřebují sportovní svazy nějakým způsobem prezentovat svoji činnost. Nejvhodnější možností je prezentace prostřednictvím webových stránek. Pro zlepšení činnosti sportovního svazu je však zapotřebí víc než jen reklama na internetu. Na informační systém jsou proto kladeny požadavky jako možnost uložení databáze funkcionářů svazu, závodníků, výsledků, soutěží, nominačního systému,...

Pokud je informační systém správně navržen, může výrazným způsobem zlepšit administrativní činnost svazu, zpřehlednit soutěžní a nominační systém, usnadnit práci pořadatelům soutěží, celkově zlepšit systém uveřejňování výsledků a postupových žebříčků atd. Elektronický informační systém, který využívá databázový systém a který je zavedený správným způsobem, je určitě přínosem oproti klasickému systému ukládání dat na papír nebo do různých dokumentů bez zachování souvislostí mezi informacemi.

### **3.2 Požadavky na IS**

Na informační systémy jsou v různých svazech kladeny různé požadavky. Základní, jako je možnost uložení informací strukturovaně do databáze a vytvoření prezentace činnosti prostřednictvím webových stránek, jsou již zmíněny v předchozí podkapitole.

Mezi další požadavky je možné zařadit přehledné administrační rozhraní, které umožňuje spravovat systém uživatelem, který nerozumí problematice tvorby webových aplikací.

Od systému, ve kterém jsou ukládány výsledky a nominační systém na soutěže vyšších úrovní, jakými může být mistrovství Evropy nebo mistrovství světa, je určitě vyžadována vysoká úroveň zabezpečení, zachování integrity a zamezení neoprávněné manipulace s uloženými daty.

Dalším požadavkem je otevřenost systému, tzn. možnost exportovat nebo importovat potřebná data ve vhodném formátu. V současné době je nejvíce používaným způsobem pro přenášání dat mezi různými systémy použití souborů ve formátu XML.

Na informační systém mohou být kladeny ještě spousty dalších požadavků, které jsou specifické pro daný sportovní svaz.

### **3.3 Současný stav**

#### **3.3.1 Svazy karate**

V České republice je v současné době pět sportovních svazů karate republikové úrovně, které svou činností navazují na nějaký světový sportovní svaz, tzn. že závodníci mají možnost postoupit na mezinárodní šampionáty, jako je mistrovství Evropy a světa. Mezi sebou se liší především různým pojetím sportovních pravidel, jako je používání různých chráničů, dovolení kontaktů na různé části těla při zápase a filosofií přístupu ke sportovní formě karate.

Je pozoruhodné, že v době obrovského rozvoje informačních technologií využívá komplexní elektronický informační systém pouze jeden z těchto sportovních svazů. Zbývající čtyři používají webové stránky pouze jako vývěsku aktuálních informací a celý soutěžní a nominační systém těchto svazů je uložen v papírové podobě nebo v podobě nesourodých elektronických dokumentů, čímž absolutně přichází o možnost automatizovaného zpracování a vyhodnocování výsledků, žebříčků závodníků a klubů, nominací na soutěže vyšší úrovně a nominací do reprezentace.

#### **3.3.2 Ostatní sportovní svazy**

Jako příklad informačních systémů ostatních sportovních svazů je vybrán systém Českého atletického svazu. Jedná se o poměrně kvalitně zpracovaný systém. Vzhledem k povaze soutěžních disciplín, které jsou ve většině případů hodnoceny porovnáváním měřitelných výkonů, jsou výsledky v systému ukládány přehledným a strukturovaným způsobem, který umožňuje vyhodnocení nejlepších výkonů, uložení různých rekordních výkonů jednotlivých atletů a uložení českých a mezinárodních rekordů.

Informační systém Českého atletického svazu je naprogramován za použití jazyka ASP a JavaScriptů, design je vytvořen pomocí kaskádových stylů a k ukládání dat je použit databázový systém MySQL. Celkově se jedná o přehledný, funkční a propracovaný informační systém. Výraznější chybou tohoto systému je, že není napsán validním kódem a z hlediska SEO optimalizace používá občas chybně titulky stránky. Tyto nedostatky jsou však z hlediska pozice ve vyhledávacích pravděpodobně vykompenzovány velikým počtem zpětných odkazů a hodnotným obsahem stránek.

## 4 Informační systém pro svaz karate

### 4.1 Požadavky

Většina obecných požadavků na informační systém pro sportovní svaz je již zmíněna v podkapitole 3.2 Požadavky na IS. Na informační systém pro svaz karate jsou však kladeny ještě další požadavky, které jsou specifické pro tento druh sportu a jeho soutěžní systém.

#### 4.1.1 Vyhodnocení bodování pomocí koeficientů

Jedním z takových požadavků je možnost vyhodnocování bodování soutěží prostřednictvím koeficientů, tzn. každé soutěži je přiřazen jeden koeficient, který určuje úroveň soutěže. Koeficientem je dáno bodové ohodnocení v případě obsazení bodovaného umístění a je jím také určeno, která umístění jsou bodována. V tabulce 4.1 je zobrazen příklad bodového hodnocení. Pro karate je typické ohodnocení dvou třetích, dvou pátých a dvou sedmých míst, protože na většině soutěží se z časových důvodů nepořádají zápasy o čtvrté, šesté a osmé místo. V prvním sloupci je číslo koeficientu, v druhém jsou bodovaná umístění a ve třetím jsou bodová hodnocení za jednotlivá umístění napsaná v řadě za sebou od hodnocení za první místo.

Tabulka 4.1- Bodové hodnocení

koeficient	bodovaná umístění	body za jednotlivá umístění
1	1.-3.	12-8-4-4
2	1.-7.	24-16-8-8-4-4-2-2
3	1.-7.	36-24-12-12-6-6-3-3
4	1.-7.	48-32-16-16-8-8-4-4
5	1.-7.	60-40-20-20-10-10-5-5

#### 4.1.2 Skupiny soutěží

Dalším požadavkem na systém je možnost zařazování soutěží do skupin, příkladem skupiny je Euro Karate Grand Prix, Golden League, Central European Karate League, skupina nominačních soutěží na mistrovství České republiky, skupina českých, skupina zahraničních soutěží,...

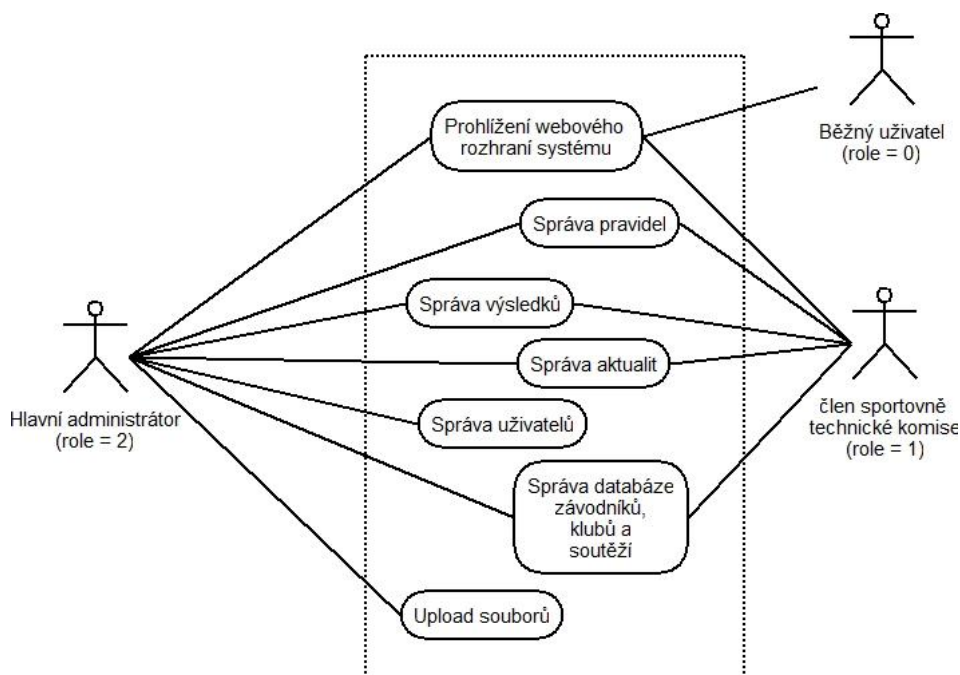
#### 4.1.3 Role uživatelů

V informačním systému pro svaz karate jsou požadovány celkem tři uživatelské role. První rolí je obyčejný uživatel, který pouze prohlíží webové stránky bez zadání hesla. Tato role je označována jako role 0, nemá žádná zvláštní oprávnění a možnosti. Uživateli s touto rolí je umožněno pouze prohlížet některé části webového rozhraní informačního systému, jako je např. kalendář akcí, žebříčky závodníků, katalog klubů a závodníků atd.

Dalším typem uživatele je člen sportovně technické komise. Tato role je označena jako role 1. Po zadání správného hesla je tomuto uživateli zpřístupněno další menu s rozšířenými možnostmi. Tento uživatel má oprávnění k prohlížení veškerého obsahu informačního systému kromě části, která se týká správy uživatelských účtů.

V administračním menu má pak tento uživatel právo spravovat všechny části systému, které se týkají pravidel, soutěžního systému, výsledků, databáze závodníků, klubů a soutěží, reprezentace a vývěsky aktualit.

Třetím typem uživatele systému je hlavní administrátor. Je mu přiřazena role 2. Tento uživatel má nejvyšší prioritu a jsou mu přidělena všechna oprávnění jako uživateli s rolí 1. Hlavní administrátor je navíc oprávněn ke správě uživatelských účtů a k nahrávání souborů na server. To je používáno při přidávání propozic soutěží a sportovních akcí do kalendáře. Z bezpečnostních důvodů může tuto akci provádět pouze uživatel s nejvyšší prioritou přístupu. Možnosti jednotlivých rolí uživatelů jsou zobrazeny v Use case diagramu na obrázku 4.1.



Obrázek 4.1 - Use case diagram

#### 4.1.4 Další požadavky na systém

Na systém jsou kladeny ještě další požadavky jako např. WYSIWYG editor vývěsky aktuálních informací, možnost přímo spravovat nominaci do reprezentace, filtrování kalendáře akcí podle skupin a roku pořádání aj.

## 4.2 Volba technologií

Jako značkovací jazyk pro tvorbu webového rozhraní informačního systému bylo vybráno XHTML, výhody tohoto jazyka jsou zmíněny v podkapitole 2.3.

K vytvoření aplikační vrstvy systému byl vybrán programovací jazyk PHP, tento jazyk byl vybrán z důvodu široké podpory na webových serverech a dalším výhodám, které jsou zmíněny v podkapitole 2.4.

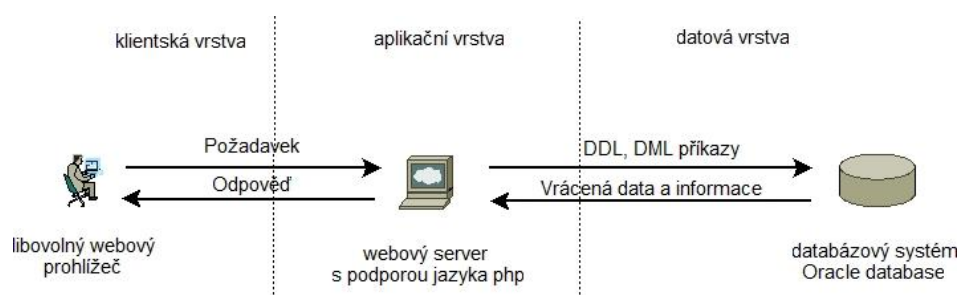


Databázový systém byl zvolen Oracle database zejména kvůli širokým možnostem, výkonu, snadnému zálohování a exportu vytvořených struktur a dat.

K rozložení webových stránek a popisu designu webového rozhraní byly zvoleny kaskádové styly CSS. K vyřešení WYSIWYG editoru vývěsky aktuálních informací byl vybrán open source editor Tiny\_mce, který je napsaný v jazyce JavaScript.

### 4.3 Rozvrstvení informačního systému

Systém je rozdělen do tří vrstev. První vrstvou je klientská, tou je libovolný prohlížeč internetových stránek se zapnutou podporou JavaScriptů. Druhou vrstvou je vrstva aplikační složená z web serveru s podporou jazyka PHP a aplikací napsanou v jazyce PHP, která je spuštěna na tomto serveru. Třetí vrstvou je datová vrstva, která je složena z databázového systému Oracle Database a napsaných funkcí uložených v souboru db\_functions.php, pomocí kterých aplikace přistupuje k databázovému systému. Na obrázku 4.2 je zobrazeno rozvrstvení systému.



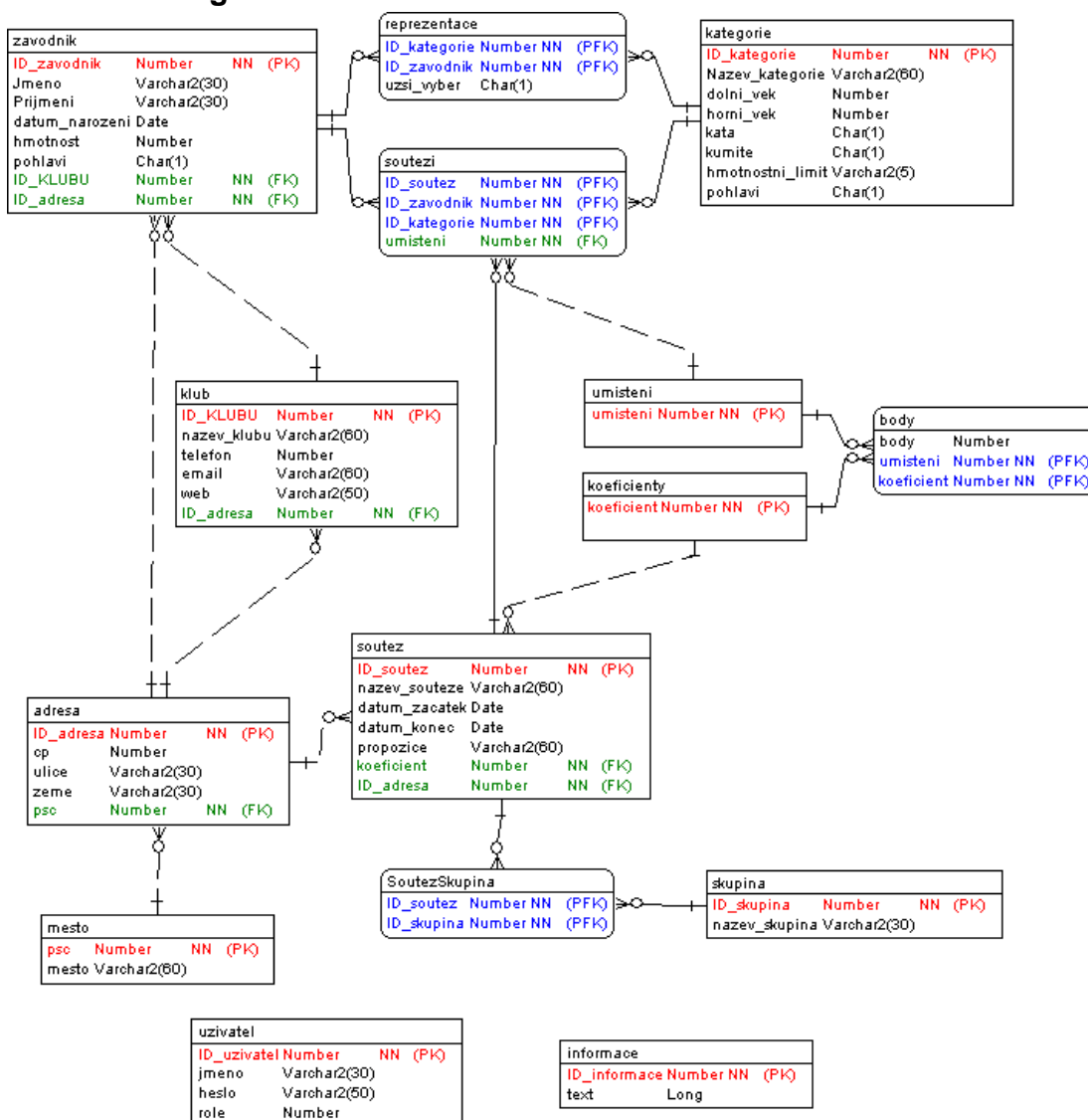
Obrázek 4.2 - Rozvrstvení informačního systému

Několikvrstvé pojetí systému bylo zvoleno z důvodu spousty výhod, které přináší. Příkladem je použití webového prohlížeče jako klientské vrstvy, díky tomu je možné k systému přistoupit ze zařízení, které je spuštěno na libovolné platformě, pouze prostřednictvím již zmíněného webového prohlížeče. Další výhodou několikvrstvého pojetí je oddělení datové vrstvy a funkcí, pomocí kterých je k této vrstvě umožněn přístup. Díky tomu je výrazně zjednodušen případný přechod na jiné datové úložiště. Výhodou oddělení aplikační vrstvy od klientské a datové je především možnost přechodu na aplikaci napsanou v jiném programovacím jazyce, nebo založenou na nějakém jiném principu.

### 4.4 Návrh databáze

Databáze je navržena tak, aby splňovala podmínky třetí normální formy a aby byla zachována integrita a bezpečnost dat.

#### 4.4.1 E-R diagram



Obrázek 4.3 - E-R diagram

#### 4.4.2 Popis tabulek a jejich atributů

##### Tabulka závodník

V tabulce jsou uloženy informace o jednotlivých závodnících. Každý závodník je členem jednoho klubu a je mu přidělena jedna adresa. Několik závodníků může být členem reprezentace v několika kategoriích. Několik závodníků může soutěžit na několika soutěžích v několika kategoriích. V tabulce 4.2 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.2 - Popis tabulky závodník**

<b>Zavodnik</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_zavodnik	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
Jmeno	Varchar2 (30 CHAR)	Jméno závodníka
Prijmeni	Varchar2 (30 CHAR)	Příjmení závodníka
ID_adresa	Number NOT NULL	<b>FK - Cizí klíč tabulky adresa</b>
datum_narozeni	date	Datum narození závodníka
hmotnost	Number	Hmotnost závodníka
pohlavi	CHAR(1)	Příznak pohlaví
ID_klub	Number NOT NULL	<b>FK - Cizí klíč tabulky klub</b>

### **Tabulka klub**

V tabulce jsou uloženy informace o jednotlivých klubech. Každý klub má nula až n závodníků a je mu přidělena jedna adresa. V tabulce 4.3 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.3 - Popis tabulky klub**

<b>Klub</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_klub	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
nazev_klubu	Varchar2 (60 CHAR)	Název klubu
ID_adresa	Number NOT NULL	<b>FK - Cizí klíč tabulky adresa</b>
telefon	Number NOT NULL	Telefonní číslo klubu
email	Varchar2 (60 CHAR)	Email klubu
web	Varchar2 (50 CHAR)	Internetové stránky klubu

### **Tabulka soutěž**

V tabulce jsou uloženy informace o jednotlivých soutěžích. Každá soutěž má přidělený jeden koeficient a jednu adresu. Několik soutěží může být členem několika skupin. V tabulce 4.4 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.4 - Popis tabulky soutěž**

<b>Soutez</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_soutez	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
ID_adresa	Number NOT NULL	<b>FK - Cizí klíč tabulky adresa</b>
nazev_souteze	Varchar2 (60 CHAR)	Název soutěže
datum_zacatek	date	Datum začátku soutěže
datum_konec	date	Datum ukončení soutěže
koeficient	Number NOT NULL	<b>FK - Cizí klíč tabulky koeficient</b>
propozice	Varchar2 (60 CHAR)	Název souboru s propozicemi

### **Tabulka kategorie**

V tabulce jsou uloženy informace o jednotlivých kategoriích. V každé kategorii může soutěžit nebo být v reprezentačním výběru nula až n závodníků. V tabulce 4.5 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.5 - popis tabulky kategorie**

<b>Kategorie</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_kategorie	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
nazev_kategorie	Varchar2 (60 CHAR)	Název kategorie
dolni_vek	Number	Dolní mez věku
horni_vek	Number	Horní mez věku
kata	CHAR (1)	Příznak kategorie kata
kumite	CHAR (1)	Příznak kategorie kumite
hmotnostni_limit	Varchar2 (5 CHAR)	Hmotnostní limit: znak +/- a číslo
pohlavi	CHAR (1)	Příznak pohlaví

### **Tabulka adresa**

V tabulce jsou uloženy jednotlivé adresy klubů, soutěží a závodníků. Tabulka je propojena s tabulkou město prostřednictvím klíče psc, tím je dodržena třetí normální forma a je tím umožněno připojení číselníku měst od společnosti Česká pošta. V tabulce 4.6 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.6 - Popis tabulky adresa**

<b>Adresa</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_adresa	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
cp	Number	Číslo popisné
ulice	Varchar2 (30 CHAR)	Název ulice
zeme	Varchar2 (30 CHAR)	Název země
psc	Number NOT NULL	<b>FK - Cizí klíč tabulky město</b>

### **Tabulka skupina**

V tabulce jsou uloženy jednotlivé skupiny soutěží. V jedné skupině může být zařazeno nula až n soutěží. Každá soutěž musí být zařazena minimálně v jedné skupině. V tabulce 4.7 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.7 - Popis tabulky skupina**

<b>Skupina</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_skupina	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
nazev_skupina	Varchar2 (30 CHAR)	Název skupiny

### **Tabulka město**

V tabulce jsou uloženy názvy měst podle PSČ. Místo této tabulky může být použit číselník společnosti Česká pošta. V následující tabulce jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.8 - Popis tabulky město**

<b>Mesto</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
psc	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
mesto	Varchar2 (60 CHAR)	Název města

### **Tabulka soutěží**

V tabulce je uloženo, který závodník v které kategorii soutěžil na které soutěži a jaké získal umístění. Tabulka je vyjádřením spojení tří tabulek ve vztahu m/n, proto je primární klíč této tabulky složen ze tří cizích klíčů. Jednotlivé atributy této tabulky jsou popsány v tabulce 4.9.

**Tabulka 4.9 - Popis tabulky soutěží**

<b>Soutezi</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_soutez	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky soutěž</b>
ID_zavodnik	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky závodník</b>
ID_kategorie	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky kategorie</b>
umisteni	Number NOT NULL	<b>FK - Cizí klíč tabulky umístění</b>

### **Tabulka reprezentace**

V tabulce jsou uloženy informace o tom, který závodník je v které kategorii v reprezentaci a zda je v užším nebo v širším výběru, to je určeno příznakem uzsi\_vyber. Pokud je nastaven, pak je závodník zařazen do užšího výběru. Jednotlivé atributy této tabulky jsou popsány v následující tabulce.

**Tabulka 4.10 - Popis tabulky reprezentace**

<b>Reprezentace</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_kategorie	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky kategorie</b>
ID_zavodnik	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky závodník</b>
uzsi_vyber	CHAR(1)	Příznak užší výběr

### **Tabulka umístění**

V tabulce jsou uložena bodovaná umístění, v karate je to většinou první, druhé, třetí, páté a sedmé místo. V následující tabulce je popis atributů tabulky.

**Tabulka 4.11 - Popis tabulky umístění**

<b>Umisteni</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
umisteni	Number NOT NULL	<b>PK - Primární klíč tabulky</b>

### **Tabulka koeficienty**

V tabulce jsou uloženy existující koeficienty. Koeficient je přiřazen soutěži a určuje úroveň bodového ohodnocení soutěže. V tabulce 4.12 jsou popsány jednotlivé atributy této tabulky.

**Tabulka 4.12 - Popis tabulky koeficienty**

<b>Koeficienty</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
koeficient	Number NOT NULL	<b>PK - Primární klíč tabulky</b>

### **Tabulka body**

V tabulce je uložena hodnota bodů pro každé hodnocené umístění pro daný koeficient. Protože je tabulka vyjádřením vztahu m/n mezi umístěními a koeficienty, je primární klíč tabulky složen ze dvou cizích klíčů těchto tabulek. Popis jednotlivých atributů je v tabulce 4.13.

**Tabulka 4.13 - Popis tabulky body**

<b>Body</b>		
<b>Atribut</b>	<b>datový typ</b>	<b>popis</b>
umisteni	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky umístění</b>
koeficient	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky koeficienty</b>
body	Number	Hodnota bodů

### **Tabulka SoutěžSkupina**

V tabulce je uloženo zařazení soutěží do jednotlivých skupin. Každá soutěž musí být zařazena minimálně v jedné skupině. Primární klíč je složen ze dvou cizích klíčů tabulek umístění a koeficienty, protože tabulka je vyjádřením vazby m/n mezi nimi. V následující tabulce je popis jednotlivých atributů.

**Tabulka 4.14 - Popis tabulky SoutezSkupina**

<b>Skupina</b>		
<b>atribut</b>	<b>datový typ</b>	<b>popis</b>
ID_skupina	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky skupina</b>
ID_soutez	Number NOT NULL	<b>PFK - Primární cizí klíč z tabulky soutěž</b>

### **Tabulka uživatel**

V této tabulce jsou uloženy přihlašovací údaje a role jednotlivých uživatelů. Rolí je jednoznačně určena úroveň oprávnění, s jakou uživatel pracuje s aplikací.

Tabulka 4.15 - Popis tabulky uživatel

Uživatel		
atribut	datový typ	popis
ID_uživatel	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
jmeno	Varchar2 (30 CHAR)	Přihlašovací jméno uživatele
heslo	Varchar2 (50 CHAR)	Hash hesla
role	Number	Role uživatele

### Tabulka informace

Tabulka je použita jako úložiště aktualit vygenerovaných za použití open source WYSIWYG editoru Tiny\_mce. Jednotlivé atributy tabulky jsou popsány v tabulce 4.16.

Tabulka 4.16 - Popis tabulky informace

Informace		
atribut	datový typ	popis
ID_informace	Number NOT NULL	<b>PK - Primární klíč tabulky</b>
text	Long	Text informace

#### 4.4.3 Popis a syntaxe použitých pohledů

Běžný uživatel, který používá informační systém bez přihlášení (v aplikaci pro svaz karate je označen rolí 0), přistupuje k databázi prostřednictvím pohledů s omezením read only. Tím je zamezeno neoprávněné manipulaci s daty v případě úspěšně provedeného útoku za pomoci technik sql injection. Zabezpečení aplikace je podrobněji popsáno v podkapitole 4.5.2.

Pohledy jsou také použity na zjednodušení složitějších dotazů do databáze. Dlouhý dotaz je rozdělen na několik kratších pohledů, čímž je dosaženo větší přehlednosti a možnosti tyto části použít znovu v jiných dotazech.

Syntaxe vytvoření pohledů je předvedena na pohledu viewSoutezKal. Tento pohled je použit při vypisování kalendáře soutěží. Názvy pohledů jsou konstruovány tak, aby bylo možné intuitivně odvodit, k čemu je daný pohled použit.

#### Příklad syntaxe vytvoření pohledu viewSoutezKal

```
create view viewsoutezkal as
select nazev_souteze, datum_zacatek, datum_konec, propozice,
nazev_skupina,zeme,mesto
from soutez,soutezskupina,skupina,adresa,mesto
where soutez.id_soutez=soutezskupina.id_souteze and
soutezskupina.id_skupina=skupina.id_skupina and
soutez.id_adresa= adresa.id_adresa and
adresa.psc= mesto.psc WITH READ ONLY;
```



### Přehled použitých pohledů

V tabulce 4.17 je popis použitých pohledů. V prvním sloupci je název pohledu a v druhém sloupci je stručný popis jeho použití v informačním systému pro svaz karate.

Tabulka 4.17 - Přehled použitých pohledů

Přehled použitých pohledů	
Název pohledu	Použití pohledu
viewZav	Pohled použitý při vyhledávání závodníků a výpisu informací o nich.
viewKlub	Pohled použitý při vyhledávání klubů a výpisu informací o nich.
viewSoutez	Pohled použitý při vyhledávání soutěží a výpisu informací o nich.
viewVysledek	Pohled použitý při výpisu soutěží, které mají výsledky, a při výpisu výsledků.
viewSoutezKal	Pohled použitý při výpisu kalendáře soutěží.
viewRepre	Pohled použitý při výpisu závodníků v reprezentaci.
viewZebricek	Pohled použitý při sestavování žebříčku a zobrazování bodů u závodníků.
viewKategorie	Pohled použitý při zobrazování kategorií.
viewSkupina	Pohled použitý při zobrazování skupin.

#### 4.4.4 Indexy

Indexy jsou v databázovém systému používány za účelem zrychlení vyhledání požadovaných dat. Bez použití indexů jsou data vyhledávána sekvenčním procházením řádků tabulky. S použitím indexů dojde k výraznému zrychlení, protože při hledání podle indexovaného sloupce je použita stromová struktura indexů. Procházení uspořádaného stromu je průměrně rychlejší než prohledávání neuspořádaného pole hodnot sloupce.

V informačním systému pro svaz karate je indexován každý sloupec, ve kterém je uložen primární klíč tabulky. Dále jsou indexovány sloupce, pomocí kterých je vyhledáváno zadáváním klíčových slov v aplikační vrstvě, je to sloupec příjmení z tabulky závodník a název\_klubu z tabulky klub.

#### Příklad vytvoření indexu nad sloupcem příjmení z tabulky závodník

```
create index indexPrijmeni ON zavodnik(prijmeni)
```

#### Přehled použitých indexů

V tabulce 4.18 je popis použitých indexů. V prvním sloupci je název indexu a v druhém sloupci je stručný popis použití v IS pro svaz karate.

**Tabulka 4.18 - Přehled použitých indexů**

Přehled použitých indexů	
Název indexu	Popis a použití indexu
indexPrijmeni	Indexuje sloupec příjmení z tabulky závodník. Je použit při vyhledávání závodníka v katalogu závodníků podle příjmení.
indexNazevKlubu	Indexuje sloupec nazev_klubu z tabulky klub. Je použit při vyhledávání klubu v katalogu klubů podle názvu klubu.
key1 až key15	Indexují primární klíč příslušné tabulky. Jsou vytvořeny automaticky.

#### 4.4.5 Sekvence

Sekvence jsou v databázovém systému Oracle database použity ke generování jedinečných identifikátorů. V informačním systému pro svaz karate jsou použity ke generování hodnoty primárního klíče jednotlivých tabulek. Syntaxe vytvoření sekvencí je předvedena na sekvenci seq\_id\_adresa. Názvy sekvencí jsou voleny tak, aby bylo možné z názvu poznat, k jaké tabulce se vztahují.

##### Příklad syntaxe vytvoření sekvence seq\_id\_adresa

```
create sequence seq_id_adresa
minvalue 1
maxvalue 9999999999999999999999999999
increment by 1 start with 1
cache 20 noorder nocycle
```

Příklad použití sekvence ke generování hodnoty primárního klíče tabulky při vkládání dat je předveden na sekvenci seq\_id\_adresa, která se vztahuje k tabulce adresa, jak je patrné z názvu této sekvence.

##### Příklad použití sekvence seq\_id\_adresa

```
insert into adresa (ID_adresa, cp, ulice, zeme, psc)
values (seq_id_adresa.nextval,:cp,:ul,:zeme,:psc)
```

#### Přehled použitých sekvencí

V tabulce 4.19 je popis použitých sekvencí. V prvním sloupci je název sekvence a v druhém sloupci je stručný popis použití v informačním systému pro svaz karate.

**Tabulka 4.19 - Přehled použitých sekvencí**

Použité sekvence	
Název sekvence	Použití sekvence
seq_id_adresa	Generování hodnot primárního klíče tabulky adresa
seq_id_kategorie	Generování hodnot primárního klíče tabulky kategorie
seq_id_klub	Generování hodnot primárního klíče tabulky klub
seq_id_skupina	Generování hodnot primárního klíče tabulky skupina
seq_id_soutez	Generování hodnot primárního klíče tabulky soutez
seq_id_uzivatel	Generování hodnot primárního klíče tabulky uživatel
seq_id_zavodnik	Generování hodnot primárního klíče tabulky závodník

#### 4.4.6 Funkce

Funkce je definována jako posloupnost příkazů, které jsou provedeny v okamžiku spuštění. Na základě vstupních parametrů jsou vráceny výstupní parametry. Pro tvorbu funkcí je v databázovém systému Oracle database použit jazyk PL/SQL.

V informačním systému pro svaz karate jsou použity dvě funkce: jedna pro zjištění aktuálního počtu závodníků a druhá pro zjištění aktuálního počtu soutěží. Názvy jsou opět zvoleny tak, aby bylo možné intuitivně odhadnout účel funkcí.

##### **Funkce GetPocetSoutez**

```
create or replace function GetPocetSoutez return NUMBER
AS
  v_pocet NUMBER;
BEGIN
  select count(*) into v_pocet from soutez;
  return v_pocet;
END;
```

##### **Příklad použití funkce GetPocetSoutez**

```
select GetPocetSoutez() as pocetSoutez from dual
```

### **Funkce GetPocetZavodnik**

```
create or replace function GetPocetZavodnik
return NUMBER
AS
  v_pocet NUMBER;
BEGIN
  select count(*) as POCET into v_pocet from zavodnik;
  return v_pocet;
end;
```

### **Příklad použití funkce GetPocetZavodnik**

```
select GetPocetZavodnik() as pocetZavodnik from dual
```

#### **4.4.7 Triggery a referenční integrita dat**

Trigger je definován jako posloupnost příkazů, která je automaticky provedena v případě předem definované operace s daty. Příkladem operace s daty je vložení, smazání a úprava dat. Posloupnost příkazů může být spuštěna buď před, nebo po provedení operace.

V informačním systému pro svaz karate jsou triggery použity pro zachování referenční integrity dat. Nad každou tabulkou, která je ve vazbě s nějakou další tabulkou a u které je to zapotřebí, je napsán trigger, který má za úkol dodržet zachování referenční integrity dat.

Příkladem tabulky, která používá trigger k zachování referenční integrity dat, je tabulka soutěž. V případě vymazání soutěže musí být zajištěno odstranění zbytků dat po této soutěži v okolních tabulkách. To je zajištěno za použití triggeru TriggerSoutDelete.

Referenční integrita dat může být řešena i za pomoci aplikační vrstvy, z hlediska rychlosti a zachování možnosti připojit se k databázi dat z jiné aplikace je výhodnější používat triggery. V informačním systému pro svaz karate je použita kombinace ošetření referenční integrity dat v aplikační vrstvě za pomoci testování datových typů a použití podmínek a v databázové vrstvě prostřednictvím triggerů.

### **Příklad triggeru - TriggerSoutDelete**

```
create or replace trigger TriggerSoutDelete
before delete on soutez
for each row
begin
  delete from soutezi where id_souteze=:old.id_soutez;
  delete from soutezskupina where id_souteze=:old.id_soutez;
end;
```

## Přehled použitých triggerů

V tabulce 4.20 je popis použitých triggerů. V prvním sloupci je název triggeru a ve druhém sloupci je stručně popsáno jeho použití v informačním systému.

Tabulka 4.20 - Přehled použitých triggerů

použitých triggerů	
Název triggeru	Použití triggeru
TriggerKatDelete	Trigger odstraní zbylá data v okolních tabulkách po smazané kategorii.
TriggerSoutDelete	Trigger odstraní zbylá data v okolních tabulkách po smazané soutěži.
TriggerSkupDelete	Trigger odstraní zbylá data v okolních tabulkách po smazané skupině.
TriggerZavDelete	Trigger odstraní zbylá data v okolních tabulkách po smazaném závodníkovi.

## 4.5 Návrh webové aplikace

### 4.5.1 Architektura aplikace

Webová aplikace je napsána v jazyce PHP. Hlavním souborem, do kterého jsou pomocí funkce include připojovány jednotlivé soubory, je index.php. Tento soubor je automaticky vyhledán a spuštěn web serverem po zadání internetové adresy webové aplikace do prohlížeče.

#### Stránkování

Stránkování je vyřešeno pomocí předávání parametru strana v odkazu. V souboru index.php je tento parametr odchycen a na základě jeho hodnoty je rozhodnuto, která stránka webové aplikace bude zobrazena. V následující ukázce je část zdrojového kódu aplikace, která řeší problematiku stránkování.

```
if(isset($_GET['strana'])){
    $strana = $_GET['strana'];
    switch ($strana){
        case 'zavodnici': include './zavodnici.php'; break;
        case 'kluby': include './kluby.php'; break;
        case 'zebricek': include './zebricek.php'; break;
        case 'vysledky': include './vysledky.php'; break;
        case 'kalendar': include './kalendar.php'; break;
        case 'nominace': include './nominace.php'; break;
        case 'reprezentace': include './reprezentace.php'; break;
        case 'informace': include './informace.php'; break;
        default: include './error.php'; break;
    }
}
```

Z ukázky je patrné, že zobrazení příslušné stránky je provedeno připojením příslušného php souboru. Soubory jsou pojmenovány stejně jako odpovídající položka v menu. Pokud je odkazem předána neplatná hodnota parametru strana, je automaticky připojen soubor error.php, který obsahuje přesměrování na úvodní stranu a případné ošetření vzniklé situace.

#### 4.5.2 Zabezpečení

Aplikace je zabezpečena na několika různých úrovních, které jsou popsány v následujících odstavcích.

##### Role uživatelů

Jedním z požadavků na zabezpečení webové aplikace je možnost rozdělit uživatele systému do několika různých skupin s různými možnostmi práce se systémem. K tomu jsou použity role uživatelů. Uživatel s nejmenšími právy má přidělenou roli 0, k tomu aby mohl aplikaci používat, se nemusí nijak autentizovat. Uživatelé s vyššími oprávněními se musí do systému přihlásit za použití uživatelského jména a hesla. Po přihlášení je do session uložen příznak autorizace, který je nastavený na true, uživatelské jméno a role. Na základě přidělené role jsou uživateli zpřístupněny příslušné položky z administračního menu.

##### Zamezení neoprávněného přístupu k souborům

V informačním systému jsou použity dva typy zabezpečení proti neoprávněnému přístupu k souborům. Prvním je zabezpečení proti pokusu o přístup k administračním souborům neautorizovaným uživatelům. Na začátku každého administračního souboru je proveden dotaz na autorizaci uživatele. Neautorizovaný uživatel je přesměrován na stranu error.php, kde dojde k ošetření vzniklé situace. V následující ukázce je část kódu, která řeší toto zabezpečení.

```
if (!isset($_SESSION['autorizace'])){\n    header ('Location: ./error.php');\n}
```

Druhým typem je zabezpečení proti přímému přístupu k souborům, které mají být přístupné pouze prostřednictvím vložení do hlavního souboru index.php. Takovým souborem je například config.php, ve kterém jsou uložena přístupová hesla k databázovému systému. Na začátku souboru index.php je nadefinována konstanta in\_code, která je nastavena na true.

```
//inicializace ochranné konstanty\ndefine('IN_CODE',true);
```

Na začátku každého zabezpečeného souboru je proveden dotaz na nastavení této konstanty.

```
if (!defined('IN_CODE')): die('Nepovoleny pristup!'); endif;
```

V případě, že není konstanta nastavena, dojde k ukončení výpisu stránky a vypsání hlášení, že se jedná o nepovolený přístup.

### **Zabezpečení hesel uživatelů**

Každý uživatel má přiděleno svoje tajné heslo, prostřednictvím kterého se přihlašuje do administrační části aplikace. Toto heslo však není z bezpečnostních důvodů uloženo v databázi ve své původní podobě, ale v podobě tzv. hashe. Hash je zašifrovaná podoba hesla takovým způsobem, že by nemělo být možné v reálném čase získat z hashe původní heslo. V případě napadení databázového systému získá útočník místo hesel pouze sled nepřehledným a bezcenných znaků.

Samotné ověření správnosti zadaného hesla při přihlašování je potom provedeno takovým způsobem, že je zadané heslo nejprve zašifrováno stejným algoritmem jako hashe, které jsou uloženy v databázi, a porovnáno s uloženým hashem hesla příslušného uživatele. V případě shody je uživatel autorizován.

V následující ukázce je zobrazen kód funkce DB\_LoginUser. Tato funkce ověří pravost kombinace zadaného uživatelského jména a hesla. Funkce vrací true v případě úspěšné autorizace.

```
function DB_LoginUser($name,$passwd){
    $hash=sha1($passwd);
    $con=DB_Pripoj();
    $select="select ROLE,id_uzivatel from UZIVATEL
        where jmeno like :na and heslo like :pa";
    $res = oci_parse($con,$select);
    oci_bind_by_name($res, ":na", $name);
    oci_bind_by_name($res, ":pa", $hash);
    oci_execute($res);
    $row = oci_fetch_array($res);
    if(isset($row['ROLE'])){
        $_SESSION['autorizace'] = true;
        $_SESSION['role'] = $row['ROLE'];
        $_SESSION['jmeno'] = $name;
        $_SESSION['id_uzivatel']=$row['ID_UZIVATEL'];
        $aut=true;
    }
    else{
        $aut=false;
    }
    DB_Odpoj($con);
    return $aut;
}
```

Na druhém řádku ukázky je již zmíněné převedení zadaného hesla na jeho hash. Dále následuje připojení k databázi, sestavení a provedení dotazu, zabezpečení proti sql injection, které je podrobněji popsáno v následujícím odstavci, poté již zmíněné uložení příznaku autorizace a parametrů uživatele a vrácení návratové hodnoty funkce.

### **Zabezpečení proti sql injection**

Způsob útoku na systém za pomoci technik sql injection je stručně popsán v podkapitole 2.8 Bezpečnost webových aplikací.

Jedním ze způsobů, jak ztížit útok pomocí sql injection, je přístupování k databázi za použití pohledů s omezením read only místo použití přímého přístupu k databázovým tabulkám. Tento způsob je použit při práci běžného uživatele bez přihlášení do systému. V případě úspěšně provedeného útoku je možné pouze získat data, ale už není možné je nějak upravovat nebo mazat. V následující ukázce je zobrazeno sestavení dotazu pomocí pohledu viewZav při vyhledání závodníka.

```
$select="select jmeno, prijmeni, nazev_klubu, zeme, id_zavodnik  
from viewZav  
where upper(viewZav.prijmeni) like upper(:jm)";
```

Dalším z použitých způsobů zabezpečení je použití funkce, která odstraní, nebo doplní o znak zpětného lomítka znaky, které mohou být zneužity při metodě sql injection. V informačním systému je použita funkce oci\_bind\_by\_name(). V následující ukázce je zobrazeno sestavení dotazu, které je použito při zjištění celkového počtu bodů konkrétního závodníka. Místo parametru id závodníka předaného z webové aplikace, tzn. že by mohl být obsah ovlivněn technikou sql injection, jsou přímo v dotazu pouze zástupné znaky uvedené dvojtečkou.

```
$select="select sum(body) as celkem_body from viewzebricek  
where id_zavodnik=:id";
```

Funkce oci\_bind\_by\_name() poté ověří vstupní data, která jsou uložena v proměnné s názvem \$id. Pokud jsou nalezeny nějaké znaky použitelné při sql injection, dojde k jejich ošetření, výsledný řetězec je poté dosazen do dotazu na místo zástupných znaků. Syntaxe použití funkce oci\_bind\_by\_name() je zobrazena v následující ukázce.

```
oci_bind_by_name($res, ":id", $id);
```

### **4.5.3 Integrita dat**

V informačním systému je zachování integrity dat zabezpečeno na dvou úrovních. První za použití triggerů v databázové vrstvě je zmíněno v části 4.4.7 Triggery a referenční integrita dat.



Druhé ošetření zachování integrity dat je provedeno přímo v aplikaci několika způsoby. Prvním způsobem je ošetření zadávání správných datových typů do vstupních polí formulářů. V následující ukázce je zobrazeno ošetření několika polí použitých při upravení nebo přidání nové kategorie do systému.

Přednastavení počátečních hodnot proměnných

```
$zprava=""; $navez_kategorie = $horni_vek = $dolni_vek = $hmotnostni_limit =  
$pohlavi = $kata_kumite = false;
```

Ošetření správnosti datových typů vstupních dat některých polí

```
//navez_kategorie  
if(empty($_POST['navez_kategorie'])){  
    $zprava.='prazdne pole navez kategorie <br />';  
}  
else{  
    $navez_kategorie=$_POST['navez_kategorie'];  
}  
//horni_vek  
if(empty($_POST['horni_vek']) || (is_numeric($_POST['horni_vek'])!=1)){  
    $zprava.='chybně vyplněno pole horní věk <br />';  
}  
else{  
    $horni_vek=$_POST['horni_vek'];  
}  
...  
...
```

V případě zadání všech dat ve správném tvaru dojde k uložení, nebo upravení kategorie, jinak je vypsána chybová zpráva.

```
if($navez_kategorie&&$horni_vek&&$dolni_vek&&...){  
    //osetreni pridani kategorie  
    if (isset($_POST['odeslano-kategorie-pridat'])){  
        DB_A_PridatKategorie($navez_kategorie,$horni_vek,$dolni_vek,...);  
        echo 'uloženo <br />';  
    }  
    //osetreni upraveni kategorie  
    if (isset($_POST['odeslano-kategorie-upravit-ulozit'])){  
        DB_A_UpravitKategorie($navez_kategorie,$horni_vek,$dolni_vek,...);  
        echo 'uloženo <br />';  
    }  
}  
else  
    echo $zprava;  
}
```

Druhým způsobem je zachování referenční integrity dat. Typickým příkladem je situace, kdy dojde k odstranění závodníka, soutěže nebo klubu. Tyto entity totiž mohou sdílet některá společná data, například adresu. Pokud dojde k vymazání některé z těchto entit, je aplikací ověřena platnost dat, která měla s mazanou entitou souvislost. Pokud jsou nalezena data, která měla souvislost pouze s mazanou entitou, pak dojde k jejich odstranění také. Příkladem takových dat je již zmiňovaná adresa, kterou má například pouze vymazávaný závodník.

#### 4.5.4 Syntaxe přístupu k databázi

Před každou prací s databází je informačním systémem provedeno nejprve připojení k databázovému systému. K tomuto účelu je použita funkce DB\_Pripoj(), která je zobrazena v následující ukázce. Funkce ke své činnosti používá parametry připojení, které jsou v zabezpečeném souboru config.php.

```
function DB_Pripoj(){
    $con = oci_connect(DB_name,DB_passwd,DB_server,'UTF8') or
die(print_r("chyba pripojeni"));
    return $con;
}
```

Po každé práci s databází je informačním systémem provedeno odpojení od databázového systému. K tomuto účelu je použita funkce DB\_Odpoj(), která je zobrazena v další ukázce.

```
function DB_Odpoj($con){
    oci_close($con);
}
```

V několika následujících ukázkách je předvedena syntaxe přístupu k databázovému systému. Jako příklad byly zvoleny části kódu použité při zjišťování měsíců pro nadpisy v kalendáři akcí.

Zavolání funkce DB\_pripoj():

```
$con=DB_Pripoj();
```

Sestavení sql dotazu:

```
$select="SELECT distinct to_char(datum_zacatek,'yyyy') as rok,
to_char(datum_zacatek,'MM') as mesic
FROM viewsoutezkal
where to_char(datum_zacatek,'yyyy') = :ro
order by mesic desc";
```

Ošetření proti sql injection a provedení dotazu:

```
$res = oci_parse($con,$select);
oci_bind_by_name($res, ":ro", $rok);
oci_execute($res);
```

Odchycení jednoho řádku vrácených dat:

```
$row = oci_fetch_array($res);
```

Uvolnění \$res a odpojení od databáze zavoláním funkce DB\_Odpoj():

```
oci_free_statement($res);
DB_Odpoj($con);
```

#### 4.5.5 Administrační rozhraní

Po autorizaci uživatele je pod běžným menu zobrazeno ještě administrační menu, které obsahuje více možností práce s informačním systémem. Samotné zobrazení je realizováno připojením souboru menu-admin.php. V tomto souboru je na základě role uživatele rozhodnuto o zpřístupnění jednotlivých položek administračního menu. Připojení souboru menu-admin.php je zobrazeno v následující ukázce.

```
if(isset($_SESSION['autorizace'])){
    include './menu-admin.php';
}
```

V jednotlivých položkách administračního menu je umožněno spravovat příslušné části informačního systému. První položka uživatel je zpřístupněna pouze uživateli s nejvyšším oprávněním. Je zde umožněno mazat a přidávat uživatelské účty.

V položce závodníci je možné provádět přidání, upravení nebo smazání závodníka.

Obdobné funkce je možné využít i při správě soutěží v položce soutěže. V této části administračního menu je navíc možné provést upload propozic soutěže nebo akce, která bude zařazena v kalendáři. Z bezpečnostních důvodů je upload souborů povolen pouze uživateli s nejvyšším oprávněním. Dále je zde možné zadávat výsledky jednotlivých soutěží.

V položce kluby a v položce kategorie je možné provádět přidání, upravení nebo smazání klubu, resp. kategorie.

Položka menu s názvem STK je použita ke správě částí systému, které se týkají nastavení pravidel soutěží nebo ke správě věcí, které jsou v kompetenci sportovně-technické komise svazu karate. Je zde možné spravovat skupiny, koeficienty, bodovaná umístění, body a vývěsku aktuálních informací, která je řešena pomocí open source editoru Tiny\_mce, který je napsán v jazyce JavaScript. Dále je zde umožněno spravovat nominaci závodníků do širšího nebo užšího výběru reprezentace.

#### 4.5.6 Rozhraní běžného uživatele

V části aplikace, která je přístupná bez přihlášení uživatele, není možné provádět žádné úpravy dat v informačním systému. Je zde umožněno pouze prohlížet některá uložená data.

Běžnému uživateli je umožněno prohlížení katalogu závodníků a katalogu klubů, výsledků a kalendáře soutěží, nominací na mistrovství České republiky, žebříčků sestavených podle zadaných kritérií, vývěsky aktuálních informací a prohlížení aktuální sestavy reprezentace. Dále je možné v této části aplikace exportovat katalog závodníků nebo klubů do formátu XML.

#### 4.5.7 Rozvržení a vzhled

Vzhled jednotlivých částí aplikace je popsán v připojeném souboru style.css. Syntaxe připojení externího souboru se stylopisem je zobrazena v následující ukázce.

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

Pokud je zadán příkaz k vytisknutí aktuální stránky, je připojen další soubor s kaskádovými styly, které přenastaví aktuální vzhled tak, aby byl optimální pro tisk, tzn. bez hlavičky a paty stránky, vše jen černou a bílou barvou. Nastavení kaskádových stylů pro tisk je uloženo v souboru style\_print.css, připojení tohoto souboru je zobrazeno v další ukázce.

```
<link href="style_print.css" rel="stylesheet" media="print" />
```

Webové stránka je rozdělena pomocí kontejnerů div na několik částí. První, základní částí, je hlava stránky, zde je zobrazeno logo informačního systému a tabulka s výpisem statistik. Pod hlavou je zobrazeno horizontální uživatelské menu. Po přihlášení je pod tímto menu zobrazeno administrační menu, které je rovněž v horizontálním provedení. Všechny výše uvedené části jsou součástí kontejneru s názvem topPan, který je při tisku celý vynechán.

Pod kontejnerem topPan je další s označením bodyPan, ve kterém je zobrazován příslušný obsah stránek podle zadané volby v uživatelském nebo administračním menu. Tato část je při tisku nastavena na černo-bílé zobrazení.

Poslední částí je kontejner s názvem footerMainPan, který obsahuje záhlaví stránky. Zde je zobrazen odkaz na stránky, ze kterých je možné stáhnout open source template, ze kterého částečně vychází rozvržení stránek informačního systému pro svaz karate. Dále je zde tlačítko login, které slouží pro přihlášení uživatele. V případě, že je již uživatel přihlášen, je zde tlačítko logout, které slouží k odhlášení uživatele. Tato část je při tisku vynechána.

Zobrazení vzhledu webového rozhraní informačního systému je v příloze A, v příloze B a C.

## 4.6 Nástroje použité při realizaci projektu

Při realizaci projektu byla použita řada softwarových nástrojů. Většina z nich je šířena pod licencí freeware, ostatní byly použity se studentskou licencí. V tabulce 4.21 je přehled použitých softwarových nástrojů. V prvním sloupci tabulky je název zvoleného softwaru, ve druhém sloupci pak licence, pod kterou je šířen, a ve třetím sloupci je popsáno, k jakému účelu byl tento software při práci použit.

Tabulka 4.21 - Nástroje použité při realizaci projektu

Nástroje použité při realizaci projektu		
Název software	Licence	Použití
DiaCze	freeware	tvorba digramů
Easy php	freeware	server pro vývoj aplikace
Microsoft Office 2007	studentská licence	tvorba dokumentace
Oracle database 10g	studentská licence	databázový systém
PhotoFiltre	freeware	úprava grafických prvků webové aplikace
PSPad editor	freeware	editace zdrojových kódů
SQL developer	freeware	správa databázového systému
Tiny_mce	open source	řešení WYSIWYG editoru v administračním rozhraní
Toad data modeler	studentská licence	návrh a tvorba databázových objektů
World-template free css	open source	základ rozvržení webového rozhraní informačního systému

## 5 ZÁVĚR

Cílem práce bylo vytvořit informační systém pro svaz karate jako webovou aplikaci s využitím relačního databázového systému. Na tento systém byly kladeny požadavky vycházející ze soutěžního systému sportovních svazů karate. Dále bylo požadováno rozdělení aplikace na dvě části: veřejnou, která je přístupná všem uživatelům bez hesla, a administrační, která je přístupná pouze členům sportovně technické komise a funkcionářům svazu po vyplnění přihlašovacích údajů.

Byl vytvořen funkční informační systém, který splňuje zadané požadavky. Po vhodném upravení pro potřeby konkrétního svazu je možné systém prakticky použít. V porovnání s dosavadním způsobem zpracovávání informací je tento systém značným přínosem. V současné době totiž využívá komplexní informační systém pouze jeden svaz karate, ostatní ukládají výsledky a data ve formě elektronických dokumentů nebo dokonce ještě v papírové podobě.

Zdrojové kódy aplikace jsou validní podle normy XHTML 1.0 Transitional a kaskádové styly jsou validní podle normy CSS level 2.1.

Před praktickým nasazením bude informační systém ještě rozšířen o možnost importování dat z losovacího programu. Dále bude z důvodu bezpečnosti systém rozšířen o šifrování přenosu dat při přihlašování a při práci v administračním režimu. V současné době systém využívá pouze šifrování hesla při přihlašování. Dále bude systém upraven tak, aby k ukládání dat používal databázi MySQL, což vzhledem k oddělení datové vrstvy nebude problém.

Práce na tomto projektu pro mě byla velikým přínosem, protože již při tvorbě teoretické části jsem si ujasnil některé důležité pojmy z tohoto oboru. Při programování praktické části jsem si v praxi vyzkoušel tvorbu uceleného projektu a díky tomu, že jsem celou aplikaci napsal ručně bez použití nějakého frameworku, jsem si uvědomil, jak velkým usnadněním práce může být použití tohoto nástroje. To je pro mě motivací k dalšímu vzdělávání v tomto oboru a k naučení práce s nějakým frameworkem.

## POUŽITÁ LITERATURA

- [1] *Webová aplikace* In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010 [cit. 31.3.2010]. Dostupné na: <[http://cs.wikipedia.org/wiki/Webová\\_aplikace](http://cs.wikipedia.org/wiki/Webová_aplikace)>.
- [2] *Webový server* In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010 [cit. 31.3.2010]. Dostupné na: <[http://cs.wikipedia.org/wiki/Web\\_server](http://cs.wikipedia.org/wiki/Web_server)>.
- [3] *Apache HTTP server project* [online]. 2010 [cit. 1.4.2010] Dostupné na: <<http://httpd.apache.org/>>.
- [4] PÍSEK, J. *HTML a XHTML – Začínáme programovat*. Praha : Grada, 2003. 256 s. ISBN 80-247-0571-0.
- [5] *Extensible Markup Language* In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010 [cit. 1.4.2010]. Dostupné na: <<http://cs.wikipedia.org/wiki/Xml>>.
- [6] VÁCLAVEK, P. *Javascript – Hotová řešení*. Brno: Computer Press, 2003. 256 s. ISBN 80-7226-854-6.
- [7] ZAJÍC, P. *Seriál o PHP* [online]. 2004 [cit. 1.4.2010] Dostupné na: <<http://www.linuxsoft.cz/php/>>.
- [8] *Active Server Pages* In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010 [cit. 1.4.2010]. Dostupné na: <<http://cs.wikipedia.org/wiki/ASP>>.
- [9] ŠIMŮNEK, M. *SQL – Kompletní kapesní průvodce*. Praha : Grada, 1999. 248 s. ISBN 80-7169-692-7.
- [10] LACKO, L. *Oracle, správa, programování a použití databázového systému*. Praha : Computer Press, 2002. 464 s. ISBN 80-7226-699-3.
- [11] GILMORE, J. *Velká kniha PHP a MySQL 5*. Brno : Zoner Press, 2005. 864 s. ISBN 80-86815-53-6.
- [12] *FIREBIRD innovative RDBMS software* [online]. 2010 [cit. 5.4.2010] Dostupné na: <<http://www.firebirdsql.org/index.php?id=about-firebird&nosb=1>>.
- [13] *About PostgreSQL* [online]. 2010 [cit. 5.4.2010] Dostupné na: <<http://www.postgresql.org/about/>>.
- [14] DRUSKA, P. *CSS A XHTML – Tvorba dokonalých webových stránek krok za krokem*. Praha : Grada, 2006. 200 s. ISBN 80-247-1382-9.

- [15] *SEO a SEM* [online]. 2010 [cit. 5.4.2010] Dostupné na:  
<<http://interval.cz/webdesign/seo-a-sem/>>.
- [16] KOSEK, J. *Bezpečnost webových aplikací* [online]. 2010 [cit. 6.4.2010]  
Dostupné na:  
<<http://www.kosek.cz/vyuka/4iz228/prednasky/bezpecnost/frames.html>>.
- [17] VRÁNA, J. *Obrana proti SQL injection* [online]. 2008 [cit. 6.4.2010] Dostupné  
na: <<http://php.vrana.cz/obrana-proti-sql-injection.php>>.



## Příloha A – Ukázka rozvržení webové aplikace – běžný uživatel – vyhledání závodníka



The screenshot displays the Karate Database website interface. At the top, there is a blue header with a silhouette of a karateka performing a high kick. The text reads "Karate Database" and "Kompletní výsledky a žebříčky karate". To the right, it says "Statistiky: 6 závodníků" and "3 soutěže". Below the header is a navigation menu with links: "Závodníci", "Kluby", "Žebříček", "Výsledky", "Kalendář", "Nominace", "Reprezentace", and "Informace". The main content area is titled "Závodníci" and features a search box with the label "Hledat příjmení:" and a "Hledat" button. Below the search box is a dropdown menu showing "-české - zahraniční - všechny -". A "Nápověda:" section contains a list of instructions: "Můžete zadat jen část příjmení", "Můžete zobrazit jen české, zahraniční nebo všechny závodníky", "Výsledky hledání závodníků můžete seřadit podle: Jméno, klub, země", and "Můžete exportovat katalog závodníků do XML". At the bottom, a footer contains the copyright information: "© Karate database 2009-2010 | design by: template World - free css | login | email: info@karatedatabase.com".

## Příloha B – Ukázka rozvržení webové aplikace – běžný uživatel – sestavení žebříčku závodníků



The screenshot shows the Karate Database website interface. At the top, there is a blue header with a silhouette of a karateka and the text "Karate Database" and "Kompletní výsledky a žebříčky karate". A navigation menu includes "Závodníci", "Kluby", "Žebříček", "Výsledky", "Kalendář", "Nominace", "Reprezentace", and "Informace". A statistics box indicates "Statistiky: 6 závodníků" and "3 soutěže". The main content area is titled "Žebříček závodníků" and features a filter bar with "Všechny skupiny", "kata mladší žáci", "od roku: 2008", "do roku: 2009", and a "Zobrazit" button. Below the filter is a table with the following data:

Pořadí	Jméno	Příjmení	Klub	Celkem bodů
1.	Pepa	Novak	Prag	72
2.	Johny	Abraham	Fight Club	28
3.	Jan	Novak	Prag	24
4.	Pepa	Novacek	Fight Club	8

At the bottom, a footer contains the text: "© Karate database 2009-2010 | design by: template World - free css | login | email: info@karatedatabase.com".

## Příloha C – Ukázka rozvržení webové aplikace – administrační rozhraní

**Karate Database**  
Kompletní výsledky a žebříčky karate

Statistiky: 8 závodníků  
3 soutěže

---

Závodníci | Kluby | Žebříček | Výsledky | Kalendář | Nomínace | Reprezentace | Informace

administrační menu ( uživatel: andrew, role: 2 )

Uživatelé | Závodníci | Soutěže | Kluby | Kategorie | STK

### Administrace - upravit/smazat závodníka

Závodník:

### Administrace - upravit závodníka

**Osobní údaje:**

Jméno:

Příjmení:

Datum narození:

Hmotnost:

Pohlaví:

**Adresa:**

Ulice:

Čp:

Psč:

Stát:

**Další údaje:**

Klub:

---

© Karate database 2009-2010 | design by: template World - free css | [logout](#) | email: [info@karatedatabase.com](mailto:info@karatedatabase.com)