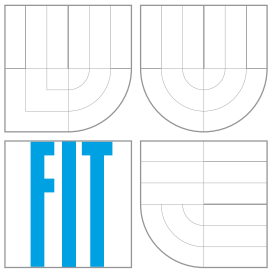


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

POKROČILÁ WEBOVÁ APLIKACE  
MISTROVSTVÍ SVĚTA V ATLETICE  
WEB APPLICATION OF WORLD ATHLETICS CHAMPIONSHIP

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

VEDOUCÍ PRÁCE  
SUPERVISOR

MIROSLAV HORÁČEK

Ing. ROMAN LUKÁŠ, Ph.D.

BRNO 2007

## Zadání bakalářské práce

Řešitel: **Horáček Miroslav**

Obor: Informační technologie

Téma: **Pokročilá webová aplikace Mistrovství světa v atletice**

Kategorie: Web

Pokyny:

1. Seznamte se s jazyky a prostředky pro tvorbu webových informačních systémů (XHTML, CSS, PHP, Javascript, MySQL).
2. Seznamte se s požadavky kladenými na IS Mistrovství světa v atletice. Rozsah systému konzultujte s vedoucím BP. Požadavky podrobně analyzujte.
3. Proveďte návrh systému. Při analýze požadavků a návrhu využijte vhodných modelovacích technik. Systém musí obsahovat i speciální funkčnost založenou na nějaké metodě z umělé inteligence. Výběr této metody konzultujte s vedoucím BP.
4. Daný systém implementujte.
5. Zhodnoťte dosažené výsledky, porovnejte váš systém s existujícími systémy, navrhněte další možné rozšíření do budoucna.

Literatura:

- Kosek, J.: PHP - Tvorba interaktivních internetových aplikací
- Maslakowski, M.: Naučte se MySQL za 21 dní
- Písek, S.: JavaScript - efektní nástroj oživení WWW stránek

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: **Lukáš Roman, Ing., Ph.D., 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, Ústřední třída 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: **Miroslav Horáček**  
Id studenta: 85622  
Bytem: Na Vyhlídce 995, 685 01 Bučovice  
Narozen: 25. 09. 1984, Vyškov  
(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: Pokročilá webová aplikace Mistrovství světa v atletice  
Vedoucí/školitel VŠKP: Lukáš Roman, Ing., Ph.D.  
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 rozmnožení.
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 nabyvatelem 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**

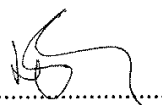
### **Závěrečná ustanovení**

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.

V Brně dne: .....

.....

Nabyvatel



Autor

## Abstrakt

Cílem této bakalářské práce bylo seznámit se s jazyky a dostupnými nástroji pro tvorbu webových informačních systémů. Hlavní část je zaměřena na analýzu, návrh a implementaci webové aplikace pro podporu konání Mistrovství světa v atletice. Účelem tohoto systému je poskytnout rozhraní a potřebné funkce předpokládaným uživatelům.

## Klíčová slova

informační systém, atletika, XHTML, CSS, Javascript, PHP, MySQL

## Abstract

Aim of this essay was to learn programming languages and available tools for creation of information systems. Main part is focused on analyse, design and implementation of the Web Application of World Athletics Championship. This information system provides interface and support for all supposed users.

## Keywords

information system, athletics, XHTML, CSS, Javascript, PHP, MySQL

## Citace

Miroslav Horáček: Pokročilá webová aplikace

Mistrovství světa v atletice, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Pokročilá webová aplikace Mistrovství světa v atletice

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Romana Lukáše, Ph.D.

.....  
Miroslav Horáček  
13. května 2007

© Miroslav Horáček, 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

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Informační systém . . . . .	3
1.2	World-Wide Web (WWW) . . . . .	4
<b>2</b>	<b>Specifikace požadavků na navrhovaný systém</b>	<b>6</b>
2.1	Vymezení cílů bakalářské práce . . . . .	6
2.2	Seznam požadavků . . . . .	6
<b>3</b>	<b>Analýza řešení</b>	<b>9</b>
3.1	Architektura systému . . . . .	9
3.2	Aktéři . . . . .	10
3.3	Volba použitých technologií a nástrojů . . . . .	10
3.3.1	Volba programovacího jazyka . . . . .	12
3.3.2	Volba databáze . . . . .	13
3.3.3	Volba webového serveru . . . . .	14
3.3.4	Volba způsobu zabezpečení aplikace . . . . .	15
<b>4</b>	<b>Implementace</b>	<b>18</b>
4.1	GUI . . . . .	18
4.2	Kvalifikace a rozběhy . . . . .	21
4.3	Funkce pro správce systému . . . . .	21
4.4	Funkce pro rozhodčí . . . . .	23
4.4.1	Třídění do rozběhů a kvalifikací . . . . .	23
4.4.2	Převod na body . . . . .	23
4.4.3	Parser na časy . . . . .	23
4.4.4	Generování náhodných výsledků . . . . .	24
4.5	Bezpečnost . . . . .	24
<b>5</b>	<b>Testování</b>	<b>25</b>
5.1	Kontrola správnosti zadávaných dat . . . . .	25
5.2	Neoprávněný přístup . . . . .	25
5.3	Čeština . . . . .	26
<b>6</b>	<b>Závěr</b>	<b>27</b>
	<b>Seznam použitých zdrojů</b>	<b>28</b>
	<b>Seznam použitých zkratk a symbolů</b>	<b>29</b>

Seznam příloh	30
A Adresářová struktura a obsah CD	31
B E-R diagram	32
C Nápověda	33



# Kapitola 1

## Úvod

V pravěku se člověk řídil převážně svým nejzákladnějším pudem – přežít, a bohatě si vystačil s informacemi a poznatky, které k tomu nezbytně potřeboval. Zajímal ho, jak si postavit obydlí a kde obstarat potravu. Přežíval v nevelkých společenstvích a nevlastnil prakticky žádný majetek, který by potřeboval nějakým způsobem evidovat.

Protože je však člověk tvor pohodlný, brzo mu takový život přestal stačit. Začal zdokonalovat své nástroje, stavěl si obydlí, od sběru plodin a lovu přešel k zemědělství a domestikaci zvířat, specializoval se na určité činnosti a jejich produkty vzájemně začal směňovat, čímž také položil základy pro budoucí obchod.

Spolu s tím vším se objevilo něco, co do té doby neznal – množství informací, které si už jednotlivec nebyl schopen jednoduše zapamatovat. Začal si proto vytvářet své vlastní „informační systémy“, ve kterých mohl uchovávat a snadno vyhledávat potřebné informace.

Velký rozvoj nastal nejprve s příchodem písma a posléze zejména s vynálezem papíru a knihtisku. Když byl popsán nebo potištěn jeden papír, začalo se psát na druhý. Jednotlivé papíry se svazovaly k sobě a vznikaly tak první kartotéky, v nichž byly stránky seřazeny podle nějakého systému.

Zkonstruováním prvního počítače však začala zcela nová éra. Počítače nejprve sloužily převážně „jen“ pro výpočty, postupem času se však zdokonalily natolik, že je bylo možné využít na uchování informací. Brzy však přestalo stačit mít všechny informace v jediném počítači a začaly vznikat první počítačové sítě. S jejich využitím se člověk stával mobilnějším, když dokázal přistupovat k potřebným informacím prakticky odkudkoli. To vše předcházelo vzniku Internetu.

V moderním světě jsme doslova zavaleni obrovským množstvím informací. Bez jejich uchování v rozumné podobě by nám brzy byly k ničemu, jelikož bychom je nedokázali efektivně využívat. Vznikají proto stále nové a nové informační systémy a v budoucnu se nedá očekávat změna tohoto trendu. Právě naopak, tempo přibývání nových informací se stále zrychluje a proto se zdá, že informační systémy čeká opravdu světlá budoucnost.

### 1.1 Informační systém

Co tedy ve skutečnosti je „informační systém“? Nejprve je vhodné si ozřejmit dvě části, ze kterých se tento pojem skládá:

1. Informační (informace) – Pro pojem informace neexistuje jednotná definice a můžeme se na ni dívat z několika pohledů. Nechci se zde zabývat slovníkovými definicemi

nebo odbornými frázemi, jde mi spíše o demonstraci odlišného chápání daného pojmu z různých úhlů lidského pohledu a odlišného kontextu. V běžné mluvě se pojmem informace obvykle rozumí zpráva, jazykový projev nebo znalost, kterou chceme sdělit jinému člověku. Z filozofického pohledu, který je pro naši potřebu přeci jen málo vhodný, ji něteří lidé chápou jako poznání, hmotu, vědomí či myšlení. Naopak v přírodních vědách, které by měly být našemu účelu nejbližší, se za informaci pokládá nějaká veličina, popřípadě konkrétní poznatek, často také na určitém stupni abstrakce. Myslím si, že pro obor informačních technologií je tato definice nejbližší, v dalším textu proto za informaci pokládejte nějaký faktický poznatek.

2. Systém – Systém lze jednoduše chápat jako množinu nějakých prvků a vazeb mezi nimi.

Informační systém je vlastně určitým virtuálním modelem skutečného systému. Tento model však nikdy nemůže postihnou veškeré chování a vlastnosti svého vzoru, je proto vždy pořízován na určité úrovni abstrakce.

Nenechte se mýlit technologickou povahou “informačního systému”, nemusí být nutně automatizovaný pomocí počítačů a může být i v papírové podobě. Každý z vás už ho určitě někdy v životě viděl a troufnu si říci, že i (možná nevědomě) vytvářel. Více o této problematice naleznete v [3].

## 1.2 World-Wide Web (WWW)

Ted', když už víme, co je to informační systém, je třeba si objasnit pojem WWW, abychom pro náš informační systém mohli používat přívlastek webový.

World-Wide Web, někdy zkráceně jen Web, je služba, která byla poprvé spuštěna v roce 1990 na půdě výzkumného centra CERN ve švýcarské Ženevě. Skládá se ze 3 základních technologií:

1. HTML (HyperText Markup Language) – Značkovací jazyk, sloužící k zápisu webových stránek.
2. HTTP (HyperText Transfer Protocol) – Protokol, zajišťující přenos HTML stránek z WWW serveru do prohlížeče.
3. URL (Uniform Resource Locator) – Každý objekt přístupný na webu má svoji jedinečnou URL adresu, která slouží k vytváření odkazů na něj.

Postupem času však jen tyto technologie přestaly dostačovat a bylo potřeba udělat weby funkčnější a interaktivnější.

První inovací byla možnost automatického generování stránek, které obsahovaly informace proměnlivé v čase. URL pak neukazuje na uloženou HTML stránku, ale na spustitelný soubor (program), který stránku, podle aktuálních informací, vygeneruje. Protože je program prováděn WWW serverem, bylo zapotřebí rozhraní, které by definovalo jejich vzájemnou komunikaci.

Toto rozhraní se jmenuje CGI (Common Gateway Interface) a programům se proto často říká CGI skripty. Tyto skripty bylo možné napsat prakticky v libovolném programovacím jazyce. Návštěvníkovi stránek pak byl nabídnut nový element jazyka HTML, formulář, který

byl právě s nějakým skriptem svázán a uživatel tak mohl aktivně ovlivňovat obsah zobrazených stránek. Nevýhodou CGI skriptů bylo složité zabezpečení a nutnost neustálé komunikace se serverem a nemožnost přesunout částečné provádění skriptů na stranu klienta. Více o CGI naleznete na [10].

Nevýhody CGI byly vyřešeny s příchodem dvou nových technologií.

První byl jazyk Java společnosti Sun Microsystems, který umožňoval začlenit krátké programy (tzv. Java applety) přímo do HTML stránky. Jeho velkou výhodou byly velké možnosti a platformová nezávislost, která si ovšem vyžádala svou daň v podobě vyšší náročnosti na systémové zdroje počítače. O této technologii se ještě zmíním v části 3.3.1.

Druhou novou technologií byl JavaScript společnosti Netscape, která jej také začlenila do svého stejnojmenného prohlížeče. Syntaxí se podobal Javě, ale pracoval zcela odlišně. Jeho nejčastější použití bylo (a dodnes je) ve spojení s formuláři. V něm napsané krátké skripty mohly kontrolovat správnost údajů v jednotlivých formulářových polích ještě před jeho odesláním na server. Dosáhlo se tak nesrovnatelně rychlejší celkové odezvy než při využití CGI skriptů. Druhou oblastí uplatnění JavaScriptu je větší interaktivita stránek podle chování uživatele. Více se o Javascriptu můžete dozvědět například z [9].

Úspěch JavaScriptu podnítil vývojáře Nescapu k jeho využití nejen na straně klienta, ale i serveru. Vzniklo tak řešení, dnes nazývané jako SSJS (Server Side JavaScript), které umožnilo některé části skriptů interpretovat serverem a výsledek vracet jako obyčejný HTML kód, začleněný do zbytku stránky.

S podobnou technologií přišla brzo i společnost Microsoft se svým ASP (Active Server Pages). Jako programovací jazyk zde byl použit buď VBScript, JScript (což je implementace JavaScriptu od Microsoftu) nebo některé jazyky „třetích stran“. Jeho velkou nevýhodou byla svázanost webového serveru s operačním systémem Windows.

SSJS a ASP mají jednu společnou nevýhodu – jedná se o komerční, a nikterak levné, produkty. Tento a spoustu dalších nedostatků odstraňuje systém PHP, který spatřil světlo světa v roce 1994. Podrobněji si ho popíšeme v části 3.3.1.

Skutečně interaktivní WWW stránky se však objevily až s příchodem technologie dynamického HTML (DHTML) a prohlížeče Microsoft Internet Explorer (IE) 4.0. Od té doby bylo možné manipulovat prakticky s každým objektem webové stránky a pomocí stylů docílit velmi zajímavých vizuálních efektů. Zároveň však zůstala zachována zpětná kompatibilita se „starými“ HTML stránkami. Kompletní výčet možností DHTML můžete najít v [8].

V nedávné minulosti se objevily další nové technologie jako třeba DOM nebo AJAX. Vývoj webových stránek (a informačních technologií obecně) jde však velmi rychle kupředu a řešení, která dnes můžu označit za „moderní“ už budou pravděpodobně za pár let působit archaicky.

## Kapitola 2

# Specifikace požadavků na navrhovaný systém

Vývoj softwarového produktu je velmi náročný a zdlouhavý proces. Zahrnuje celou řadu činností a pracovních postupů. Při začátku práce na jakémkoli projektu je však zcela zásadní přesné vymezení cílů a požadavků na vytvářený produkt, popis konkrétních funkcí a vlastností, které musíme jako vývojáři implementovat.

### 2.1 Vymezení cílů bakalářské práce

Tato práce s klade za cíl obecně prostudovat problematiku tvorby informačních systémů, naučit se používat příslušné pracovní nástroje, osvojit si postupy a celkově navrhnout a implementovat webový informační systém pro podporu atletického mistrovství světa s respektováním obecně daných zásad a konvencí.

### 2.2 Seznam požadavků

Zde se pokusím, co možná nejvýstižněji, popsat, jaké požadavky a cíle jsou na výsledný produkt kladeny.

1. Registrace  
Nový uživatel se musí před prvním použitím aplikace nejprve zaregistrovat.
2. Přihlášení do systému.  
Uživatel se do systému přihlašuje:
  - (a) uživatelským jménem a heslem, které zadal při registraci (obyčejný uživatel)
  - (b) přihlašovacími údaji, které mu byly přiděleny administrátorem (rozhodčí)
  - (c) jako administrátor
3. Odhlášení ze systému  
Uživateli musí být umožněno, pro zajištění větší bezpečnosti, se libovolně odhlásit. Navíc je nutné implementovat funkci pro automatické odhlášení po určité době nečinnosti.
4. Editace osobních údajů  
Uživatel může své osobní údaje libovolně upravovat.

## 5. Správa uživatelů

Administrátor spravuje osobní informace o registrovaných uživatelích. Je mu umožněno:

- (a) vyhledávání a filtrování záznamů o uživatelích
- (b) přidání nového uživatele
- (c) editace stávajícího uživatele a změna typu jeho účtu
- (d) smazání uživatele

## 6. Správa atletů

Administrátor spravuje osobní informace o závodnících. Je mu umožněno:

- (a) vyhledávání a filtrování záznamů o atletech
- (b) přidání nového atleta, jeho osobních údajů, disciplín, které provozuje a jeho osobních rekordů
- (c) editace stávajícího atleta, jeho osobních údajů, disciplín, které provozuje a jeho osobních rekordů
- (d) smazání atleta

## 7. Správa rozhodčích

Administrátor spravuje osobní informace o rozhodčích a řídí nasazování rozhodčích na jednotlivé disciplíny. Je mu umožněno:

- (a) vyhledávání a filtrování záznamů o rozhodčích
- (b) přidání nového rozhodčího
- (c) editace stávajícího rozhodčího
- (d) smazání rozhodčího
- (e) inteligentní a efektivní nasazení rozhodčích na jednotlivé disciplíny
- (f) sledování pracovního vytížení rozhodčích

## 8. Správa startovních listin

Systém musí zobrazovat kompletní seznam závodníků, kteří se účastní dané atletické disciplíny a umožnit rozhodčím podle přihlášených účastníků automaticky generovat startovní listiny závodů. Závodníci jsou do jednotlivých drah při závodech nasazováni preferenčně podle předchozí výkonnosti.

## 9. Správa výsledků

V závislosti na vytvořené startovní listině jsou rozhodčími k jednotlivým atletům zadávány jejich výkony. Zadané výsledky lze následně editovat a mazat. Pro účely zjednodušení testování a následné demonstrace funkčnosti bude vytvořen jednoduchý generátor pseudonáhodných výsledků

## 10. Kvalifikace a rozběhy

Navrhněte optimální model rozběhů a kvalifikací pro jednotlivé skupiny disciplín.

## 11. Pořadí národů

Systém poskytuje statistické pořadí jednotlivých zúčastněných zemí:

- (a) podle počtu získaných medailí, přičemž prioritu má celkový počet medailí, dále se řadí vždy podle počtu cennějších medailí

- (b) podle celkového počtu získaných bodů; bodově ohodnoceno je prvních 8 míst v každém závodě, přičemž za 1.místo se uděluje 8b, za každé další o bod méně.
12. Správa kvalifikačních limitů  
Rozhodčí může zadávat a editovat kvalifikační limity pro jednotlivé technické disciplíny, po jejichž splnění postupuje závodník automaticky do finále.
  13. Správa rekordů  
Rozlišujeme 4 různé typy rekordů:
    - (a) PB – absolutní nejlepší životní výkon každého jednotlivce v dané disciplíně
    - (b) SB – nejlepší výkon závodníka v dané disciplíně v aktuálním kalendářním roce
    - (c) WR – absolutní světový rekord v dané disciplíně
    - (d) WL – nejlepší výkon v dané disciplíně v aktuálním kalendářním roce
  14. Integrita a správnost dat  
Systém musí zajistit integritu a konzistenci zpracovávaných dat. Musí být také ošetřeny vstupy uživatelů.
  15. Zabezpečení přístupu  
Systém musí zajistit přístup pouze oprávněným osobám a podle typu uživatelského účtu zobrazovat příslušné funkce.

## Kapitola 3

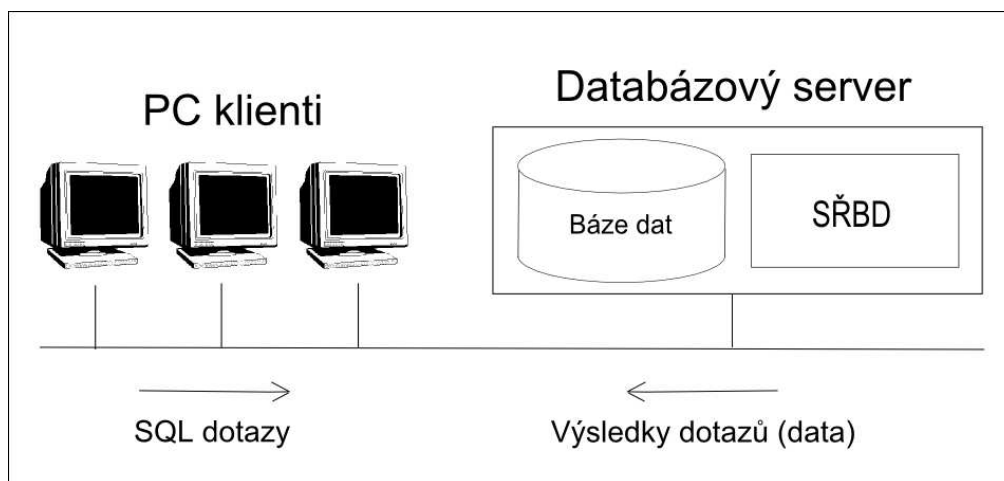
# Analýza řešení

V této kapitole popíši, jak jsem postupoval při návrhu aplikace. Nejprve navrhnu základní architekturu celého systému. Následně vymezím okruh aktérů, kteří budou do aplikace přistupovat a určím požadavky na jejich role. Na závěr se budu podrobněji věnovat výběru technologií a nástrojů, použitých při tvorbě celého systému.

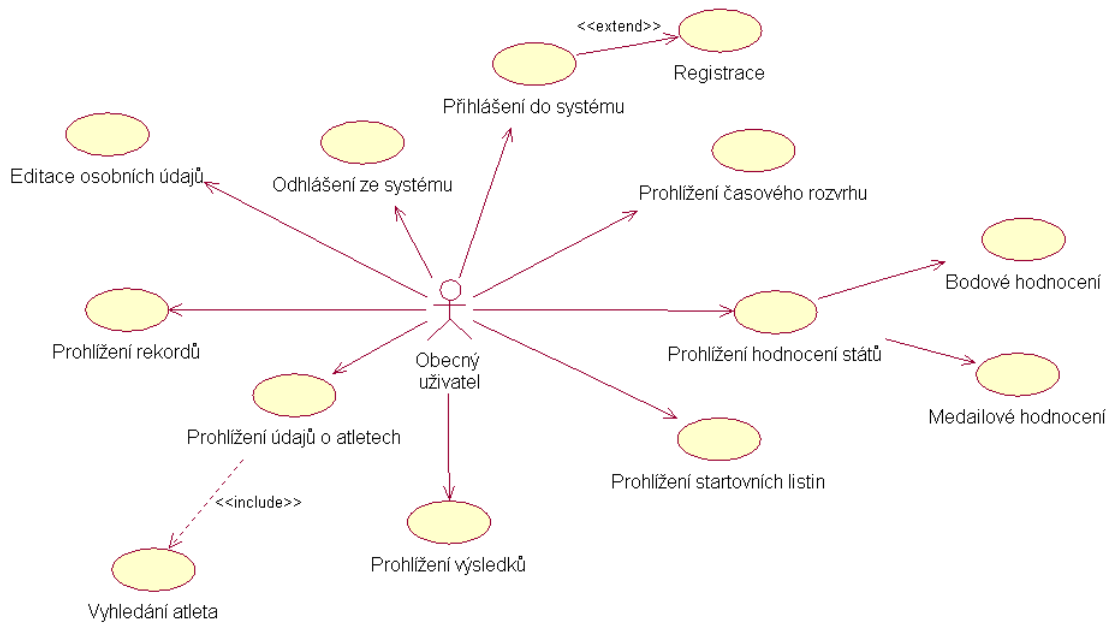
### 3.1 Architektura systému

Hlavní funkce aplikace spočívá v uchovávání dat a v poskytnutí prostředí pro jejich správu (vyhledání, přidání, editaci a vymazání). Vzhledem k tomu, že je systém určen pro širokou škálu uživatelů, je jasné, že do něj bude přistupovat velké množství uživatelů, kteří jsou navíc různě geograficky rozmístěni.

Základem systému bude společná databáze. Ta bude spolu s vlastní aplikací umístěna na internetu a může k ní tedy přistupovat každý, kdo má připojení na internet a disponuje internetovým prohlížečem. Zvolená architektura se nazývá klient/server a znázorňuje ji obrázek 3.1.



Obrázek 3.1: Architektura Klient-Server



Obrázek 3.2: UCD Obecný uživatel

## 3.2 Aktéři

Pojmem aktér se rozumí role, ve které vystupuje uživatel v rámci jeho komunikace se systémem.

Z požadavků na aplikaci vyplynulo, že jednotlivým uživatelům mají být umožněny jen ty funkce, na které mají patřičná oprávnění. Vytvářet pro každého uživatele jemu specifický seznam funkcí by ale bylo značně neefektivní. Daleko lepší možností bude rozdělit uživatele do skupin. To znamená, že každý uživatel, kterému chceme umožnit přístup do systému, bude mít přidělenou konkrétní uživatelskou skupinu.

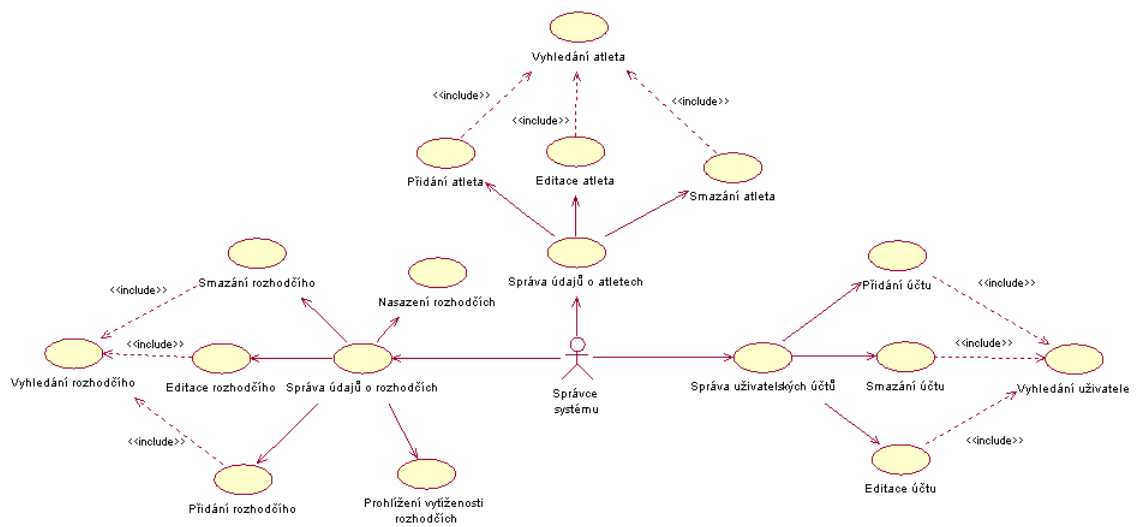
V tuto chvíli se nabízí otázka, kolik takovýchto skupin bude. Při prostudování seznamu požadavků jsem zjistil, že v systému budou vystupovat pouze 3 aktéři. Těmi, obecně pro celý systém, jsou:

- Obecný uživatel – Vyhledává a prochází jemu dostupné informace a výsledky. Jeho role je zobrazena na obrázku 3.2.
- Správce systému – Řídí chod celé aplikace, spravuje účastníky závodů a účty uživatelů. Jeho role je zobrazena na obrázku 3.3.
- Rozhodčí – Má na starosti správu všech závodů. Jeho role je zobrazena na obrázku 3.4.

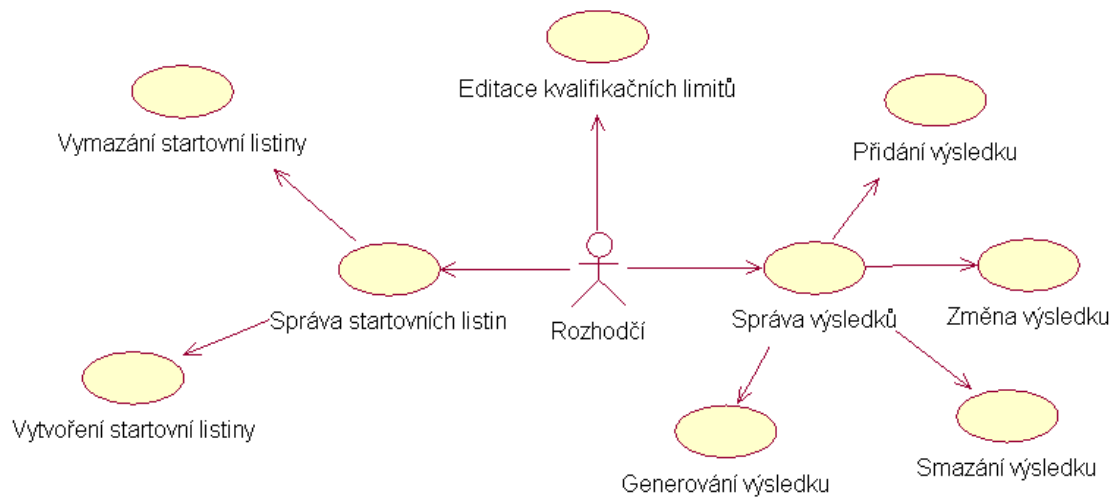
## 3.3 Volba použitých technologií a nástrojů

Volba technologií a vývojářských nástrojů je pro většinu softwarových projektů naprosto klíčovou záležitostí, která ovlivní celý proces návrhu, tvorby a následné údržby finálního produktu.





Obrázek 3.3: UCD Správce systému



Obrázek 3.4: UCD Rozhodčí

Pro základní návrh daného systému z hlediska uživatelských rolí jsem použil grafický jazyk UML (Unified Modeling Language), který se v softwarovém inženýrství používá na vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů, neobsahuje však způsob, jak se má používat, ani neobsahuje metodiku, jak analyzovat, specifikovat či navrhovat programové systémy. Pro zájemce existuje na toto téma velmi hodnotná publikace [1].

K samotnému psaní zdrojového kódu nebylo potřeba využít žádné sofistikovanější vývojové prostředí, jelikož celý zdrojový kód byl psán „ručně“. Použil jsem proto, zejména mezi programátory, oblíbený textový editor PSPad.

Uživatelská část aplikace je tvořena trojicí technologií:

- HTML – Použití tohoto jazyka (aktuální jsou verze HTML 4.01 a XHTML 1.1) bylo samozřejmostí a tak je HTML jedním ze základních pilířů celého systému. Jeho obecnou charakteristikou jsem se už zabýval, proto dříve zmíněné informace pouze doplním. V praxi je používáno několik prohlížečů WWW stránek a každý na vlastní vykreslení obsahu používá své metody. Pro odstranění tohoto nešvaru byly vytvořeny standardy, bohužel ne všechny prohlížeče (včetně obecně nejrozšířenějšího IE) se jimi stoprocentně řídí. Vývojáři webových stránek pak mají obecně dvě možnosti. Tou první je striktně dodržovat standardy a doufat, že uživatel bude buď používat „kvalitní“ prohlížeč nebo se standardy konečně začnou dodržovat. Druhá varianta je přizpůsobení se způsobu vykreslování v IE s pomocí různých „fíglů“, které zde nechci dále rozebírat. Jelikož validita stránek byla jednou z požadovaných vlastností a podpora standardů (především CSS) se v poslední verzi IE zlepšila, rozhodl jsem se pro první možnost a celou aplikaci jsem implementoval v souladu s HTML standardy. Více o standardech naleznete na stránkách konsorcia W3C [10] nebo na [4].
- JavaScript – Už také dříve zmíněný multiplatformní skriptovací jazyk. Syntaxí je podobný rodině jazyků C/C++/Java, jinak však zcela odlišný. Obvykle se vkládá přímo do HTML kódu stránky. Ke správnému chodu používá interpret, který postupně prochází zdrojový kód a vykonává každý příkaz za příkazem bez potřeby kompilace. Nevýhodou takových jazyků je jejich mnohem menší rychlost ve srovnání s klasickými jazyky, které vytváří binární spustitelné soubory, výhodou je snadnější údržba a vývoj.
- CSS – Jedná se o jazyk (v současnosti aktuální verze CSS 2.1) pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. O způsobu práce se styly v CSS se dále zmíním v části 4.1.

### 3.3.1 Volba programovacího jazyka

Pro tvorbu webových aplikací se využívá celá řada programovacích jazyků. Zde uvádím ty nejrozšířenější.

- PHP – O tomhle velmi oblíbeném skriptovacím jazyku jsem se už také zmiňoval, jelikož jeho role v historickém vývoji webu je neoddiskutovatelná. Je relativně jednoduchý na ovládnutí, skripty jsou prováděny na straně serveru, k uživateli je pak přenášěn až výsledek činnosti. Velkou předností je jeho nezávislost na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. V aktuální verzi PHP 5 byla zřetelně vylepšena podpora objektově-orientovaného programování (OOP), nehodí se

však pro rozsáhlé projekty z důvodu tzv. řízení toku. Kvalitní učebnicí tohoto jazyka je [5].

- Perl – Byl vytvořen v roce 1987. Je to interpretovaný programovací jazyk, který se s rozvojem internetu stal velmi populárním nástrojem pro tvorbu CGI skriptů. Jde o jazyk rychlý, stabilní a podporující OOP. Má obrovské vyjadřovací schopnosti, ale často na úkor přehlednosti. Na moderní rozsáhlé projekty se už příliš nehodí.
- Python – Jedná se o interpretovaný objektivě orientovaný jazyk. Aplikace napsané v Pythonu jsou velmi výkonné, jelikož výkonnově kritické knihovny jsou efektivně napsány v jazyce C. Často se používá ke skriptování v aplikacích jako jsou grafické editory nebo kancelářské aplikace.
- Ruby – U zrodu tohoto, čím dál více se rozšiřujícího, jazyka stála snaha spojit pohotovost a flexibilitu skriptovacího jazyka s elegancí a silou OOP. Díky své přehledné syntaxi a striktní objektivosti se hodí i k psaní velkých projektů. Je přenositelný mezi platformami a podporuje například i regulární výrazy. Čas tohoto jazyka zřejmě ještě přijde.
- .NET – Tento termín zastřešuje celý soubor moderních technologií společnosti Microsoft, které jsou podporovány frameworkem Microsoft Visual Studio .NET.
- Java – Java je jedním z nejpoužívanějších programovacích jazyků na světě. Díky své přenositelnosti je používána pro programy, které mají pracovat na různých systémech, počínaje čipovými kartami (JavaCard), přes mobilní telefony a různá zabudovaná zařízení (Java ME), aplikace pro desktopové počítače (Java SE) až po rozsáhlé distribuované systémy pracující na řadě spolupracujících počítačů rozprostřené po celém světě (Java EE). Tyto technologie se jako celek nazývají platforma Java. Jeho syntaxe je zjednodušenou verzí jazyků C/C++. Namísto skutečného strojového kódu se vytváří pouze tzv. mezikód („bajtkód“), který je dále interpretován, za určitých okolností (pokud by to pomohlo zrychlení provádění) se však můžou některé části zkompileovat do strojového kódu a následně i optimalizovat. Tento jazyk je velmi vhodný pro psaní rozsáhlých a komplexních projektů. Prakticky jedinou nevýhodou je vyšší náročnost na systémové zdroje, která se projeví zejména u menších aplikací, protože je třeba mít v paměti počítače umístěné i celé běhové prostředí Java Virtual Machine (JVM).

Dlouho jsem svůj výběr v této otázce zvažoval, nemohl jsem se rozhodnout mezi jazyky Java a PHP. Nakonec jsem zvolil jazyk PHP, který si mě, relativního nováčka ve výše zmíněných technologiích, získal zejména svou jednoduchostí, přesto však širokou použitelností. Ne, že by Java nebyla pro mé konkrétní využití vhodná, spíše mi připadalo využití PHP v mé situaci jako zcela dostačující. Pro komplexnější projekt většího rozsahu bych si ovšem technologii založenou na platformě Java určitě vybral.

### 3.3.2 Volba databáze

Důležitou součástí každého informačního systému je databázový server. Protože ovlivňuje značnou část celého systému, ovlivní jeho volba i samotnou implementaci. Vliv na výběr použité databáze může mít spousta faktorů, zejména požadovaná funkčnost a celková rychlost. Mezi nejrozšířenější databáze v současnosti patří MS SQL, Oracle, MySQL, PostgreSQL, mSQL a Firebird.

Z hlediska požadované funkčnosti aplikace je třeba říci, že si vystačíme ve své podstatě s jednoduchou databází bez nutnosti nějakých specifických funkcí. Od serveru požadujeme „pouze“ zpracování základních dotazů typu SELECT, INSERT, UPDATE a DELETE. Proto je pro náš případ dostačující kterákoliv z výše uvedených databází.

Z hlediska obecného rozšíření a různého způsobu licencování jsem však prakticky vybíral jen mezi dvěma volně šiřitelnými produkty – MySQL a PostgreSQL. Nakonec jsem zvolil databázi MySQL, jelikož podle nezávislých testů dosahuje o něco vyššího výkonu než PostgreSQL. V problematice MySQL a databází obecně mi byla velmi nápomocna publikace [6].

### 3.3.3 Volba webového serveru

Termín webový server může znamenat v zásadě dvě věci:

- Počítač, který je odpovědný za vyřizování požadavků HTTP od klientů. Tyto požadavky obvykle přicházejí pomocí webových prohlížečů (IE, FF, Opera). Vyřízením požadavků se rozumí odeslání webové stránky (dokumentu HTML) zpět ke klientovi.
- Software, který provádí činnosti popsané v přechozím bodě.

Jednotlivé webové servery se mohou v různých aspektech značně lišit, mají však několik společných vlastností.

Každý webový server je připojen k počítačové síti a přijímá požadavky ve tvaru HTTP. Tyto požadavky vyřizuje a počítač, který požadavek vznesl, vrací odpověď. Odpověď obvykle představuje HTML dokument, může to být ale libovolný textový dokument, obrázek apod.

Součástí odpovědi je i stavový kód odpovědi. Ten udává, zda byl požadavek vyřízen v pořádku, či zda došlo k problémům. Běžným stavovým kódem, označující bezproblémové provedení požadavku je 200 (nebo kódy řady 2xx). Další řady 3xx, 4xx a 5xx značí chybu, ať už způsobenou přesměrováním, chybou na straně serveru nebo klienta.

Webový server má v zásadě dvě možnosti, jak získávat informace, které vrací klientům:

- Statický obsah. Server má předem připravené HTML stránky (nebo jiné dokumenty), které v nezměněné podobě poskytuje klientovi.
- Dynamický obsah. Data jsou shromážděna (například přečtena z databáze) a zformátována až na základě požadavků klienta a poskytnuta webovému prohlížeči.

K dynamickému vytváření obsahu se používá celá řada různých technologií (PERL, PHP, ASP, ASP.NET, JSP apod). Statický obsah je schopen server poskytnout významně rychleji než dynamický. Na druhé straně pomocí dynamického obsahu lze poskytovat mnohem větší obsah informací a lze reagovat i na různé „ad hoc“ dotazy klientů. Proto se v praxi v mnoha případech oba způsoby poskytování obsahu kombinují – například pomocí cachování.

Mezi nejpopulárnější webové servery současnosti patří:

- Apache HTTP Server. Volně šiřitelný produkt pod záštitou společnosti Apache Software Foundation, vyvíjený od roku 1994. Prakticky od začátku jde o celosvětově nejpoblárnější softwarový webový server, na kterém v současnosti běží skoro 2/3 všech webových stránek. Díky své modularitě poskytuje velkou možnost rozšíření serveru o další funkce.

- Internet Information Services. Druhý nejpopulárnější webový server, dosahující tržního podílu asi 30%. V minulosti měl velké problémy se zabezpečením, v aktuální verzi 7.0 došlo v této oblasti ke zlepšení. Bohužel se jedná o proprietární produkt společnosti Microsoft a nikoho snad nepřekvapí, že je tudíž vázán na OS Windows.

Některá další řešení jako Sun Java System Web Server, IBM HTTP Server, Oracle HTTP Server nebo Zeus Web Server zde uvádím už spíše pro ilustraci, protože v celkovém souhrnu dosahují podílu pouze v řádu jednotek procent a jejich využití je spíše ve specializované korporátní sféře. Více opět na [11].

Díky výše zmíněným výběrům programovacího jazyka a databáze byla volba webového serveru už spíše formalitou, jelikož kombinace Apache, MySQL a PHP (někdy ještě doplněná Linuxem na, ve světě svobodného software, téměř posvátnou zkratkou LAMP) je pro svou dobrou vzájemnou kompatibilitu a výkonnost nejpoužívanějším světovým řešením pro informační systémy.

### 3.3.4 Volba způsobu zabezpečení aplikace

V době kdy Internet vznikal, byl určen jen pro akademický svět a pro takový způsob využití, který vyšší úroveň bezpečnosti a spolehlivosti nevyžadoval. Tehdy ještě nikoho nenapadlo, že lidé budou jednou chtít nakupovat a platit po Internetu, budou mu svěřovat čísla svých kreditních karet, svou obchodní i soukromou korespondenci, a dokonce že na něj budou sázet své podnikání. A že v souvislosti s tím se zákonitě objeví i lidé, kteří budou chtít Internet zneužít k nejrůznějším nekalostem.

Dnes už však o nezbytnosti nějakým způsobem zabezpečit své aplikace, pracující v „Síti sítí“ snad nepochybuje téměř nikdo. Seznámíme se se třemi nezbytnými požadavky, které se navíc netýkají pouze Internetu, ale obecně jakéhokoli on-line prostředí ve kterém lidé chtějí vykonávat své každodenní aktivity.

- Identifikace (autentikace) – Podstatou tohoto požadavku je to, aby se vždy spolehlivě zjistilo „kdo je kým“ – ať již jde o odesílatele nějaké zprávy, autora WWW stránky, či třeba žadatele o nějakou službu či přístup k určitému zdroji atd. Oba pojmy, tedy „identifikace“ a „autentikace“, se často zaměňují, resp. považují za pojmy s identickým významem. V některých kontextech ale lze mezi nimi spatřovat rozdíly. Identifikace je o tom, aby se vůbec zjistilo, kdo je autorem, původcem, iniciátorem atd. určité aktivity. Autentikace následně spolehlivě ověří, že příslušné údaje jsou pravdivé.
- Autorizace – V rámci tohoto požadavku jde o to, aby konkrétní uživatel získal (či naopak nezískal) oprávnění přístupu k určitému zdroji (například dokumentu) či k provedení určité aktivity (spuštění programu, provedení transakce apod). Cílem je realizovat určitou „přístupovou politiku“, v rámci které mají různí uživatelé různá oprávnění. Autorizace je samozřejmě velmi závislá na správné identifikaci a autentikaci, neboli na spolehlivém zjištění, kdo o něco žádá – teprve pak je možné mu na základě stanovených pravidel buď oprávnění poskytnout nebo odmítnout.
- Integrita dat – Obsah nejrůznějších dat, přenášených či alespoň zpřístupňovaných prostřednictvím Internetu, může být různým způsobem pozměněn oproti stavu, kdy jej autor vytvořil – například v důsledku chyb při přenosu nebo třeba v důsledku toho, že někdo jiný než původní autor tato data z nějakého důvodu pozměnil. Obecně se zde hovoří o tom, že takováto data ztratila svou původní „integritu“. Snaha vyvarovat se něčemu takovému je pak označována jako snaha o „zachování integrity dat“. Leckdy

je ale takovýto požadavek na skutečné zachování integrity (vlastně na zachování dat v jejich původní podobě) příliš silný, resp. nerealizovatelný, a tak se celý požadavek redukuje alespoň na možnost spolehlivě zjistit, zda od vytvoření dat jejich původním autorem došlo či nedošlo k nějaké jejich změně (v tom smyslu, že pokud ke změně došlo, příjemce se dozví pouze fakt že nějaká změna nastala, ale již se nemusí dozvědět, o jakou změnu konkrétně šlo a v čem spočívá). Praktickým cílem je dát příjemci možnost spolehnout se na to, že data jsou skutečně „původní“ a nezměněná, nebo jej varovat že byla pozměněna.

Při respektování těchto požadavků jsem během hledání způsobu zabezpečení mého informačního systému dospěl k výčtu možných následujících technických řešení pro moji aplikaci:

- IP Adresa – Tento starý způsob už dnes není možné využívat z jednoho prostého důvodu. Spousta uživatelů má dnes dynamicky přidělované adresy nebo je „schována“ za proxy serverem a tudíž je jedna IP adresa společná pro více uživatelů. Ojedinelé nejsou i případy, kdy pod jednou IP adresou vystupuje celá organizace.
- Jedinečný identifikátor předávaný v URL – Další metoda využívaná hlavně dříve. Při vstupu uživatele na server je, pomocí některého z mnoha nabízejících se způsobů, vygenerován jedinečný řetězcový či celočíselný identifikátor, který se uloží do databáze spolu s časem provedení poslední akce uživatelem. Při dalším pohybu uživatele na serveru se tento identifikátor předává jako parametr v URL a při vstupu na další stránku se vygeneruje nový identifikátor a tímto identifikátorem a novým časem poslední akce uživatele se přepíše starý údaj v databázi. Výhodami tohoto způsobu jsou nízké nároky na server a žádné problémy s podporou ze strany prohlížečů. Nevýhodou je v porovnání s HTTP Autorizací a sessions nižší bezpečnost a zbytečný parametr v URL navíc.
- HTTP Autorizace – Spolu se sessions nejvyužívanější a s největší pravděpodobností také jeden z nejlepších způsobů autorizace. Pokud využíváte HTTP Autorizaci, nehrozí vám problémy s podporou ze strany prohlížečů, které jsou noční můrou všech programátorů využívajících ve svých skriptech cookies. Další velmi podstatnou výhodou oproti jiným způsobům autorizace je bezpečnost. Pokud pomineme odhlašování, nelze HTTP Autorizaci nijak prolomit. Nevýhody jsou dvě – PHP musí běžet jako modul serveru a nelze naprogramovat jednoduché odhlašování. Buď musíte ukládat pomocná data do databáze, čím ztrácí HTTP Autorizace výhodu oproti sessions, druhou možností je provést chybné přihlášení, což ale taktéž není optimálním řešením. Třetí a poslední možnost je zcela neprogramátorská – stačí prostě zavřít okno prohlížeče. Je to 100% bezpečné, ale velmi neelegantní.
- Cookies – Tento populární způsob zabezpečení trochu ztratil s příchodem PHP4. Výhodou cookies je to, že u nich lze nastavit delší časovou platnost a platí pro všechna okna prohlížeče a nikoli pro jedno, jak je tomu u všech ostatních způsobů. Je to způsobeno tím, že cookie neplatí pro konkrétní okno prohlížeče, ale pro doménu, kterou programátor definuje. Další výhodou je možnost uložit do cookies další data.
- Sessions – Český též relace, řeší problém bezstavovosti protokolu HTTP. Udržíme pomocí ní informace o stavu aplikace a práci uživatele s ní. Prakticky je toto řešení

použitelné až od PHP verze 4.1. Každý uživatel, který vstoupí na stránku, obdrží jednoznačné číslo (spíše shluk znaků) SID (Session ID, př. 601249534f314eccbdf44f34d2f235be). Tímto číslem se bude uživatel identifikovat a ve všech odkazech je zapotřebí předávat tento parametr (implicitně se předává v proměnné PHPSESSID).

Pro práci se sessions se používá několik funkcí:

- `session_start()` – Inicializuje relaci.
- `session_start()` – Odstraní všechna data relace.
- `$_SESSION['proměnná']` – Zápis lokální proměnné do superglobální proměnné, nahrazuje dřívější, potenciálně nebezpečnou, funkci `session_register()`.
- `unset($_SESSION['proměnná'])` – Zrušení hodnoty proměnné v session, dřívější funkce `session_unregister` je z důvodu bezpečnosti nedoporučená.
- SID – Konstanta, která v případě platné relace obsahuje identifikátor relace. Pokud jsou povolena cookies, potřebná data jsou předávána přes ně a konstanta SID neexistuje.

Implementačně nejjednodušší a celosvětově nejoblíbenější je zřejmě používání cookies. Bohužel existují (a zřejmě i vždy budou existovat) paranoidní uživatelé, kteří se obávají o ochranu svého soukromí, protože toto řešení umožňuje serveru ukládat si pomocí cookies o uživateli veškeré zjištěné informace. Pro potřeby svého systému jsem proto zvolil poslední jmenovanou variantu, sessions, jelikož funguje, i když jsou cookies zakázána, umožňuje přenášet mezi stránkami potřebné informace a umožňuje též bezproblémové odhlášení.

## Kapitola 4

# Implementace

V této kapitole se již budu věnovat samotné implementaci mého informačního systému. Popisovat zde detailně způsob realizace všech komponent by bylo příliš zdlouhavé a ve své podstatě i zbytečné. Nechci zde ani prezentovat byť jen útržky zdrojového kódu. Ten si ostatně každý zájemce může prohlédnout na příloženém CD.

### 4.1 GUI

Grafický návrh aplikace vycházel z obecných standardů a pro dosažení jednoduché orientace běžného uživatele jsem se nepokoušel nijak „experimentovat“. Vytvořil jsem 4 samostatné části okna aplikace, názorně zobrazené na obrázku 4.1:

Vzhledem k využití jazyka PHP jsem se rozhodl vytvořit samostatné soubory pro každé požadované menu a tyto jsem pak do každého potřebného souboru začlenil pomocí příkazu `include`.

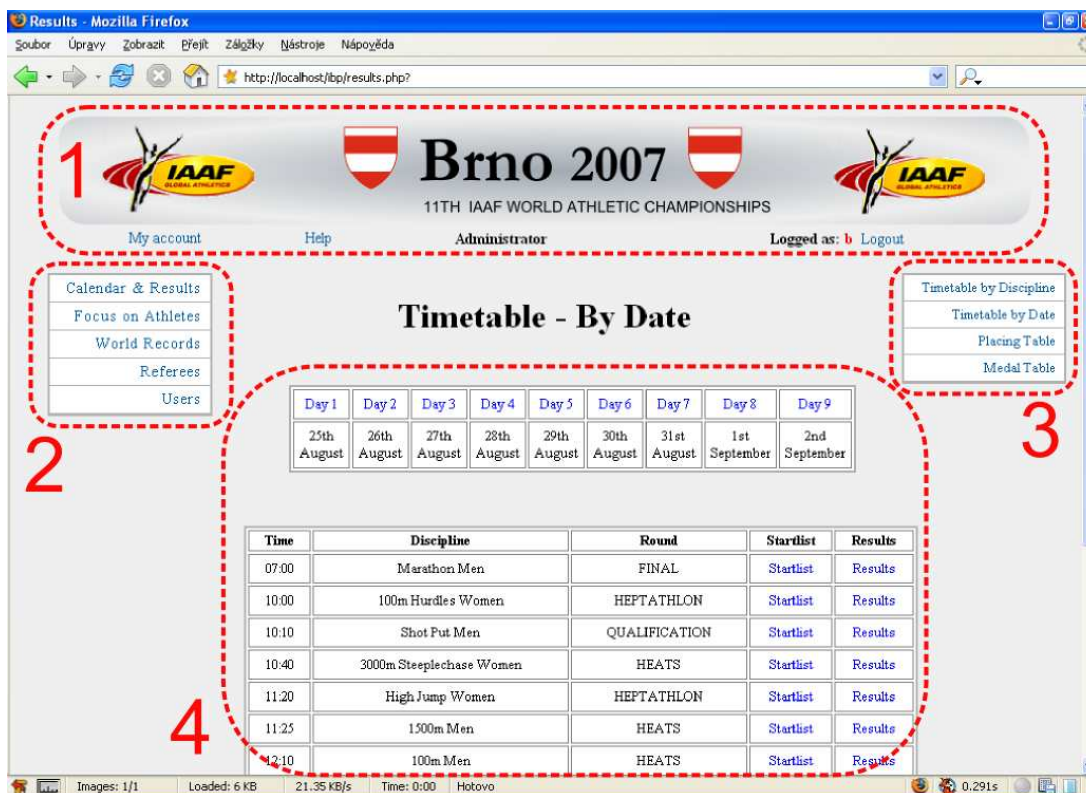
1. Do horní části stránky jsem umístil hlavní logo a základní informace o právě přihlášeném uživateli (`menu_top.php`).
2. V levé části stránky je trvale zobrazené plovoucí hlavní menu, jehož nabídka se mění jen podle typu uživatele (`menu_left.php`).
3. Na pravé straně se příležitostně objevuje menší kontextové menu s nabídkou funkcí, vztahující se k právě zobrazené stránce (kolekce souborů `menu_right_*.php`).
4. Největší část tvoří hlavní část okna, ve které se objevují požadované stránky.

Je logické, že správce systému požaduje odlišnou funkčnost aplikace než rozhodčí nebo obyčejný uživatel. Jednotlivé nabídky a menu s funkcemi se proto zobrazují odlišně na základě typu účtu přihlášeného uživatele.

Existuje několik možností, jak udělat layout (rozložení) HTML stránky. Společným cílem je vytvořit stránky přehledné, snadno použitelné a vizuálně přitažlivé.

1. Rámy – Zastaralý způsob, využívající HTML značky `<frameset>` a `<frame>`. Do této kategorie patří i vnořené rámy `<iframe>`. Obsah každého rámu je tvořen jedním samostatným HTML dokumentem, jeden kód je proto použitelný na více stránkách. Vzhledem k tomu, že je obsah rámu stále přístupný, používá se toto řešení zejména pro menu. Velikost rámu lze podle potřeby přizpůsobovat. Některá zařízení však rámy





Obrázek 4.1: Layout aplikace

nezobrazují, vyskytují se také problémy s vyhledávací a komplikovaný je i design a tisk stránek.

2. Tabulky – Dovedávná velmi rozšířený a i dnes velmi populární způsob, který nabízí, i přes svou relativní jednoduchost, velké možnosti pozicování elementů stránky. Má však řadu nevýhod, zejména dlouhý a nepřehledný kód a neoddělenost vzhledu od obsahu. Tabulka se v prohlížeči také zobrazí, až se celá načte, což může působit rušivě při pomalejším internetovém připojení.
3. Kaskádové styly (CSS) – Programátor vytváří na stránce, obvykle pomocí tagů `<div>` a `<span>`, logické vizuální bloky, uvnitř kterých je už jen samotný obsah. Vzhled a rozmístění těchto bloků je už definováno v CSS. Tenhle způsob je jednoznačně nejlepší a byl využit i v mé práci. Pro zájemce o tuto problematiku doporučuji publikaci [7].

Připojení stylu k HTML dokumentu:

- (a) Externí soubor se stylem – Myšlenka CSS počítá primárně s tím, že se definice stylů umísťuje do externího souboru, který se pak už jenom odkazuje libovolně mnoho stránek. Syntaxe jednotlivých řádků je `selektor{vlastnost: hodnota}`. Pravidla se aplikují postupně a vzniká tak kaskáda, která dala jméno samotným kaskádovým stylům.

*ukázka CSS souboru styl.css:*

```

p {text-indent: 30px; margin: 0px 0px 0px 0px}
a {text-decoration: none}
a:link {color: green}
a:visited {color: navy}
a:active {color: black}
a:hover {color: red; text-decoration: underline}
h2 {color: blue; font-style: italic}
h1 {color: green; text-align: center}

```

*ukázka připojení CSS souboru:*

```

...
<head>
  <title>Stránka s externím stylovisem</title>
  <link rel='stylesheet' type='text/css' href='styl.css'>
</head>
<body>
  <h1>Externě stylovaná stránka</h1>
...

```

- (b) Deklarace v hlavičce – Tímto způsobem je však nutné deklarovat styl v každém dokumentu a pracně se tudíž udržuje konzistence.

*ukázka:*

```

...
<head>
  <title>Titulek</title>
  <style type='text/css'>
    body {color: white;}
    h1 {color: red;}
  </style>
</head>
<body>
...

```

- (c) Inline styly – K jednotlivým elementům uvádíme jenom deklaraci vlastností bez selektoru a { }.

*ukázka:*

```

...
<body>
  <h1 style='color: red'>Nadpis</h1>
  <span style='font-weight: bold'>Tučné písmo</div>
  <div style='font-size: 20px'>
    Blok velkým písmem
  </div>
...

```

## 4.2 Kvalifikace a rozběhy

Vzhledem k velkému množství zúčastněných závodníků má každá disciplína definován systém kvalifikací (u technických disciplín) a rozběhů, čtvrtfinále a semifinále (u běžeckých disciplín). Výjimkami, u kterých se přistupuje rovnou k finálovým kláním, jsou mimodráhové disciplíny (maratón a chodecké soutěže), víceboje (sedmiboj a desetiboj) a taktéž nejdelší dráhová disciplína – běh na 10000m.

- Kvalifikace – S přihlédnutím k počtu přihlášených atletů jsou vytvořeny 2 kvalifikační skupiny. Z nich automaticky postoupí do finále všichni závodníci, kteří splnili vypsany kvalifikační limit (tito jsou označeni písmenem Q). Pokud je počet těchto atletů menší než počet předpokládaných postupujících do finále (tj.12) je zbývající počet vybrán podle dosažených výkonů z ostatních závodníků (q).
- Rozběhy – Závodníci jsou rozděleni do rozběhů, z nichž vždy určitý počet nejlepších postupuje do dalšího kola (Q), další postupující (q) jsou po skončení všech rozběhů vybráni podle nejlepších časů z těch, kteří nepostoupili přímo do dalších bojů.

Systém kvalifikací a rozběhů názorně zobrazuje tabulka 4.1.

Disciplíny	Kolo	Počet skupin	Q	q
100m, 200m	Rozběhy	5	4	12
	Čtvrtfinále	4	3	4
	Semifinále	2	4	0
400m, 800m, 1500m, 100m př., 110m př., 400m př.	Rozběhy	4	3	4
	Semifinále	2	4	0
5000m, 3000m př.	Rozběhy	3	4	4
4x100m, 4x400m	Rozběhy	3	2	2
Skoky a hody	Kvalifikace	2	0-12	0-12
ostatní	Přímo finále			

Tabulka 4.1: Systém kvalifikací a rozběhů pro jednotlivé disciplíny

## 4.3 Funkce pro správce systému

Pro správce systému jsou důležité zejména funkce, sloužící k vyhledání, přidání, úpravě a smazání jednotlivých atletů, rozhodčích a uživatelů. Tyto funkce nejsou nikterak zvláštní, v zásadě se jedná jen o formuláře svázané se skripty, pracujícími se základními tabulkovými operacemi. Nebudu je zde proto dále rozebírat.

Specialitou mé aplikace je inteligentní systém nasazování rozhodčích na jednotlivé závody, která je implementována ve skriptu `Sort_referees.php`. Každý rozhodčí má definované oprávnění pro řízení jednotlivých skupin disciplín. Jejich označení je následující: R – běhy, J – skoky, T – hody, W – chůze a X – víceboje. Systém prohledává volné rozhodčí s patřičným oprávněním a nasazuje ho na danou disciplínu. V případě, že žádného volného nenajde, nastává tzv. kolizní situace.

Při kolizních situacích se využívá tzv. Backtrackingu neboli „metody zpětného navrácení“. Jedná se o speciální metodu umělé inteligence, která prohledává stavový prostor

„do hloubky“. Místo expanze vybraného uzlu, tj. generování všech jeho bezprostředních následníků, se generuje pouze následník jeden a teprve v případě neúspěchu se generuje následník další, atd. Pokud uzel není na cestě k řešení (tj. všichni jeho bezprostřední následníci „ohlásili“ neúspěch, vrací uzel neúspěch svému bezprostřednímu předchůdci a popsaná situace s postupnou generací následníků se opakuje v tomto uzlu.

Algoritmus metody Backtracking je následující:

1. Sestrojí zásobník OPEN (bude obsahovat uzly určené k expanzi) a umístí do něj počáteční uzel.
2. Je-li zásobník OPEN prázdný, pak úloha nemá řešení a ukončí proto prohledávání jako neúspěšné. Jinak pokračuj.
3. Jde-li na uzel na vršku zásobníku aplikovat první/další operátor, tak tento operátor aplikuj a pokračuj bodem 4, v opačném případě odstraň testovaný uzel z vrcholu zásobníku a vrať se na bod 2.
4. Je-li nový vygenerovaný uzel, tj. uzel vzniklý aplikací operátoru na uzel na vršku zásobníku, uzlem cílovým, ukončí prohledávání jako úspěšné a vrať cestu od kořenového uzlu k uzlu cílovému (vrať se posloupnost stavů, nebo operátorů). Jinak ulož nový uzel na vršek zásobníku a vrať se na bod 2.

Metoda má extrémně nízkou paměťovou náročnost, protože v paměti (tj. v zásobníku OPEN) jsou uloženy pouze uzly aktuální cesty spolu označením právě použitých operátorů. Časová náročnost je exponenciální, metoda není ani optimální, ani úplná. Modifikace metody spočívá v úpravě bodu 4. Nový uzel se do zásobníku OPEN neukládá, pokud se již v tomto zásobníku nachází, tj. pokud nový uzel je již předchůdcem uzlu na vršku zásobníku. Další informace o metodách umělé inteligence můžete najít v [2].

Má funkce `Sort_referees()` funguje následovně:

- Nejprve vybere z databáze všechny rozhodčí, zjistí si jejich ID, stav a specializaci.
- Poté pro každý den:
  - Vybere z databáze všechny závody, zjistí si jejich ID, typ, začátek a dobu trvání.
  - Vypočte celkový potřebný čas rozhodčího na celou disciplínu včetně přípravy mezi závody, které má řídit.
- Následně pro každý den:
  1. Vytvoří pole `zasobnik_ok`, do kterého bude vkládat nasazené rozhodčí.
  2. Po minutách simuluje časový průběh.
  3. Pokud je v daný čas potřeba rozhodčího:
    - Vytvoří si pro každou disciplínu pole `zasobnik_vsech` a vloží do něj všechny vhodné rozhodčí pro danou disciplínu, zároveň si pamatuje i vrchol zásobníku.
    - Postupně testuje, jestli je některý ze zásobníku volný (a zároveň nastavuje nový vrchol zásobníku).
      - \* Pokud ANO - Vloží ho na vrchol `zasobnik_ok`, přiřadí mu ID závodu, nastaví mu čas uvolnění a pokračuje bodem 2.

\* Pokud NE - Vrací se na předchozí závod a zkouší na něm zabrat dalšího rozhodčího ze zásobníku. Tímto způsobem se zpětně navrácí ve „stromu“ a zkouší procházet nové „větve“. Pokud neúspěšně projde všechny „větve“, funkce oznámí, pro které dny nemá úloha řešení. V tomto případě musíme přidat další rozhodčí s potřebnou specializací.

- Pokud nastal čas pro uvolnění rozhodčího, uvolníme ho.
- Na závěr vkládá nasazené rozhodčí do tabulky `Referees_sort`.

## 4.4 Funkce pro rozhodčí

Pro některé specifické záležitosti rozhodčích jsem si musel vytvořit několik vlastních funkcí.

### 4.4.1 Třídění do rozběhů a kvalifikací

Do systému jsem implementoval skript `sort.php`, který umožňuje rozhodčím podle přihlášených účastníků automaticky generovat startovní listiny závodů. Závodníci jsou do jednotlivých drah při závodech nasazováni preferenčně:

- do rozběhů a kvalifikací podle nejlepších sezónních výkonů
- do čtvrtfinále, semifinále a finále podle výkonů v předešlém kole
- při vícebojích podle nejlepších sezónních výkonů v jednotlivých disciplínách

Při běžeckých disciplínách jsou jednotlivé dráhy ohodnoceny svou výhodností v pořadí (od nejvýhodnější k nejméně výhodné): 4-5-3-6-7-2-8-1.

U technických disciplín (skoky, hody, vrhy) nastupují závodníci v pořadí od nejhoršího k nejlepšímu.

Aplikace obsahuje taktéž skript `delete_startlist.php`, kterým lze danou startovní listinu vyčistit a posléze přistoupit k novému nasazení.

### 4.4.2 Převod na body

Pro vícebojařské disciplíny (sedmiboj pro ženy a desetiboj pro muže) jsem si vytvořil dvojici funkcí `points_ht(Disciplína,Výkon)` a `points_dt(Disciplína,Výkon)`, které dosažené výkony přepočítají na body podle dvojice vzorců:

1.  $A * (B - P)^C$  pro běžecké disciplíny
2.  $A * (P - B)^C$  pro technické disciplíny

ve kterých jsou A,B,C konstanty a P dosažený výkon v příslušných jednotkách. Konstanty jsou znázorněny v tabulkách 4.2 a 4.3.

### 4.4.3 Parser na časy

Jelikož funkce na výpočet bodů z dosažených výkonů nezvládá formát výsledků závěrečných disciplín obou vícebojů (800m pro sedmiboj a 1500m pro desetiboj), musel jsem si vytvořit vlastní funkci `Time_to_sec(Výkon ve tvaru „mm:ss.ms“)` pro přepočet časů na formát „ss.ms“.

Disciplína	A	B	C	P
100m	25.437	18.0	1.81	sec
Skok daleký	0.14354	220	1.40	cm
Vrh koulí	51.39	1.5	1.05	m
Skok vysoký	0.8465	75	1.42	cm
400m	1.53775	82	1.81	sec
110m překážek	5.74352	28.5	1.92	sec
Hod diskem	12.91	4.0	1.1	m
Skok o tyči	0.2797	100	1.35	cm
Hod oštěpem	10.14	7.0	1.08	m
1500m	0.03768	480	1.85	sec

Tabulka 4.2: Bodovací vzorce pro desetiboj

Disciplína	A	B	C	P
100m překážek	9.23076	26.7	1.835	sec
Skok vysoký	1.84523	75	1.348	cm
Vrh koulí	56.0211	1.5	1.05	m
200m	4.99087	42.5	1.81	sec
Skok daleký	0.188807	210	1.41	cm
Hod oštěpem	15.9803	3.8	1.04	m
800m	0.11193	254	1.88	sec

Tabulka 4.3: Bodovací vzorce pro sedmiboj

#### 4.4.4 Generování náhodných výsledků

Pro účely zjednodušení testování a následné demonstrace funkčnosti byl vytvořen skript `generate_result.php`, sloužící ke generování náhodných výkonů. Skript funguje analogicky jako, taktéž dostupná, dvojice skriptů `add_result.php` a `insert_result.php`, s tím rozdílem, že hodnoty nejsou přebírány od uživatele, nýbrž jsou pseudonáhodně generovány pomocí zabudované funkce jazyka PHP `int rand ([int minimum, int maximum])`. Rozsah hodnot je pro každou disciplínu nastavován individuálně a lze ho ovlivnit editací souboru `generator_values.php`. Obsažen je i skript `delete_results.php`, kterým příslušnou výsledkovou listinu vyčistíte.

## 4.5 Bezpečnost

Pomocí skriptu `verify.php`, který je příkazem `include` začleněn do každé stránky, je kontrolována platnost příslušné session. Pokud není session platná, je uživateli zobrazeno varovné hlášení a je přeměrován na přihlašovací stránku.

Stejně je do každé stránky vložen skript `autologout.php`, který uživatele po 10-ti minutách nečinnosti automaticky odhlásí.

## Kapitola 5

# Testování

Ověřování správné funkčnosti probíhalo ve dvou etapách. První byla kontrola a testování už během psaní zdrojového kódu, druhá, komplexní, až po celkovém dokončení.

V první fázi jsem průběžně testoval chování každé nové přidané funkce a kontroloval chování dříve vytvořených funkcí, které tato novinka mohla nějakým způsobem ovlivnit. Taktéž jsem se snažil uměle vyvolávat všechny možné situace a zaměřil se na pečlivé ošetření extrémních stavů.

Druhá fáze nastala až s dopsáním posledního řádku kódu. Zaměřil jsem se v ní na správnou komunikaci a interakci mezi jednotlivými součástmi aplikace, jestli jsou každému aktérovi k dispozici právě ty funkce, které požaduje a jestli je systém použitelný jako celek.

### 5.1 Kontrola správnosti zadávaných dat

Programátor musí vždy počítat, že uživatel je schopen do požadovaných polí napsat, ať už neúmyslně nebo zcela záměrně, opravdu cokoli. V horších případech některé potřebné položky nevyplní raději vůbec. Ve své aplikaci jsem striktně požadovaná pole označil hvězdičkou (\*) a bez jejich vyplnění nebude daná funkce provedena. Druhou věcí je správnost zadávaných údajů. Zde bohužel nelze nikdy zcela vyloučit uživatelské vstupy jako „ffdfs“ nebo „ijijiji“. Alespoň jsem tedy zajistil správný formát e-mailové adresy. K implementaci jsem si vytvořil javascriptovou funkci `check_email(e-mail)`, která využívá regulární výraz:

$$^[_a-zA-Z0-9\ \.-]+\@[_a-zA-Z0-9\ \.-]+\.[a-zA-Z]2,4\$$$

Ale ani to, že adresa bude mít správný formát nezaručí, že e-mail je opravdu funkční. Jedním z řešení tohoto problému by bylo odeslat na tuto adresu zprávu s potvrzovacím odkazem a až poté účet aktivovat.

### 5.2 Neoprávněný přístup

Na zabezpečení aplikace by měl být obecně dbán velký důraz. Při pokusu o neoprávněný přístup se uživateli objeví varovné okno a bude přesměrován na přihlašovací obrazovku.



Aplikace byla umístěna na internet na adresu <http://ibp.ic.cz>. Ve výchozím stavu jsou v systému vytvořeny účty s následujícími přihlašovacími údaji pro každý typ uživatele:

- Obecný uživatel - user/user
- Správce systému - admin/admin
- Rozhodčí - referee/referee

### 5.3 Čeština

Vzhledem k častým problémům s češtinou z důvodu rozlišných kódování na různých systémech jsem se rozhodl celou aplikaci napsat v anglickém jazyce. Jako kódování souborů jsem použil znakovou sadu Latin-2 (ISO/IEC 8859-2).



## Kapitola 6

### Závěr

Cílem této práce bylo prostudovat problematiku tvorby informačních systémů a jeden specifický typ, zabývající se problematikou Mistrovství světa v atletice, navrhnout a implementovat. Myslím, že mohu zodpovědně prohlásit, že se mi zadané úkoly podařilo splnit.

Výsledkem prostudování dané problematiky a důkladného rozboru celého systému je seznam požadavků, které jsou na podobné aplikace kladeny.

Jedním z nezbytných požadavků na systém byla jeho reálná použitelnost. Musím však přiznat, že profesionální návrh a realizace takto náročného a komplexního projektu by byl daleko za rozsahem bakalářské práce. Řadu problémů jsem musel zastříť velkým množstvím abstrakce a některé funkce dosti zjednodušit, jiné, dříve uvažované, vynechat úplně. Přesto jsem přesvědčen, že je aplikace ve své současné formě nebo jen po mírných úpravách reálně použitelná pro méně náročné uživatele.

Srovnání s podobnými produkty je v mé konkrétní situaci velmi obtížné, jelikož se jedná o velmi specifický typ softwarového projektu.

Mým osobním cílem bylo zejména osvojit si postupy při analýze projektu a jeho následné realizaci. Daná práce velmi významně přispěla k praktickému využití teoretických poznatků z oblasti vývoje softwarového produktu.

# Seznam použitých zdrojů

- [1] Jim Arlow a Ila Neustadt. *UML a unifikovaný proces vývoje aplikací*. Computer Press, 2003. ISBN 80-7226-947-X.
- [2] Doc. Ing. František Zbořil CSc. a Ing. František Zbořil Ph.D. *Studijní opora: Základy umělé inteligence*. FIT VUT v Brně, 2006.
- [3] Prof. Ing. Tomáš Hruška CSc. a Ing. Zbyněk Krivka. *Studijní opora: Pojem informačního systému, Data, Procesy, Transakce*. FIT VUT v Brně, 2006.
- [4] Dušan Janovský. Jak psát web, návod na html stránky.  
<http://www.jakpsatweb.cz>, 2007. [Online; cit. 27-April-2007].
- [5] Jiří Kosek. *PHP - tvorba interaktivních internetových aplikací*. Grada Publishing, Praha, 1. edition, 1999. ISBN 80-7169-373-1.
- [6] Mark Maslakowski. *Naučte se MySQL za 21 dní*. Computer Press, Praha, 1. edition, 2001. ISBN 80-7226-448-6.
- [7] Eric Meyer. *Eric Meyer o CSS - Ovládněte kaskádové styly!* Zoner Press, Brno, 2004. ISBN 80-86815-03-X.
- [8] RNDr. Pavol Mikle. *Rozšířené dynamické HTML - referenční příručka*. Zoner Press, 2004. ISBN 80-86815-01-3.
- [9] Slavoj Písek. *Javascript - efektní nástroj oživení WWW stránek*. Grada Publishing, Praha, 3. edition, 2001. ISBN 80-247-0014-X.
- [10] W3C. World wide web consortium. <http://www.w3.org/>, 2007. [Online; cit. 27-April-2007].
- [11] Wikipedia. Web server – wikipedia, the free encyclopedia.  
[http://en.wikipedia.org/w/index.php?title=Web\\_server&oldid=125669028](http://en.wikipedia.org/w/index.php?title=Web_server&oldid=125669028), 2007. [Online; cit. 27-April-2007].

# Seznam zkratek

<b>AJAX</b>	Asynchronous JavaScript and XML
<b>ASP</b>	Active Server Pages
<b>CGI</b>	Common Gateway Interface
<b>DHTML</b>	Dynamic HyperText Markup Language
<b>DOM</b>	Document Object Model
<b>FF</b>	Mozilla Firefox
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>IE</b>	Microsoft Internet Explorer
<b>JS</b>	JavaScript
<b>JVM</b>	Java Virtual Machine
<b>OOP</b>	Objektově orientované programování
<b>PDF</b>	Portable Document Format
<b>PHP</b>	PHP: Hypertext Preprocessor
<b>SSJS</b>	Server-side JavaScript
<b>UML</b>	Unified Modeling Language
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>W3C</b>	World Wide Web Consortium
<b>XHTML</b>	Extensible HyperText Markup Language

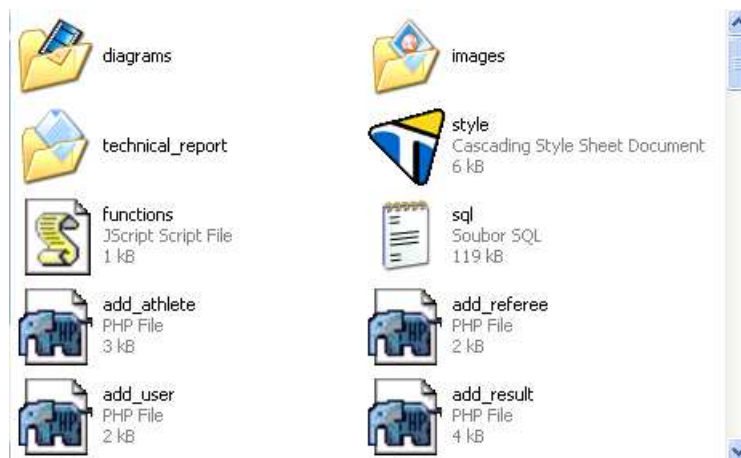
# Seznam příloh

- A** Adresářová struktura a obsah CD
- B** E-R diagram
- C** Náповěda

## Příloha A

# Adresářová struktura a obsah CD

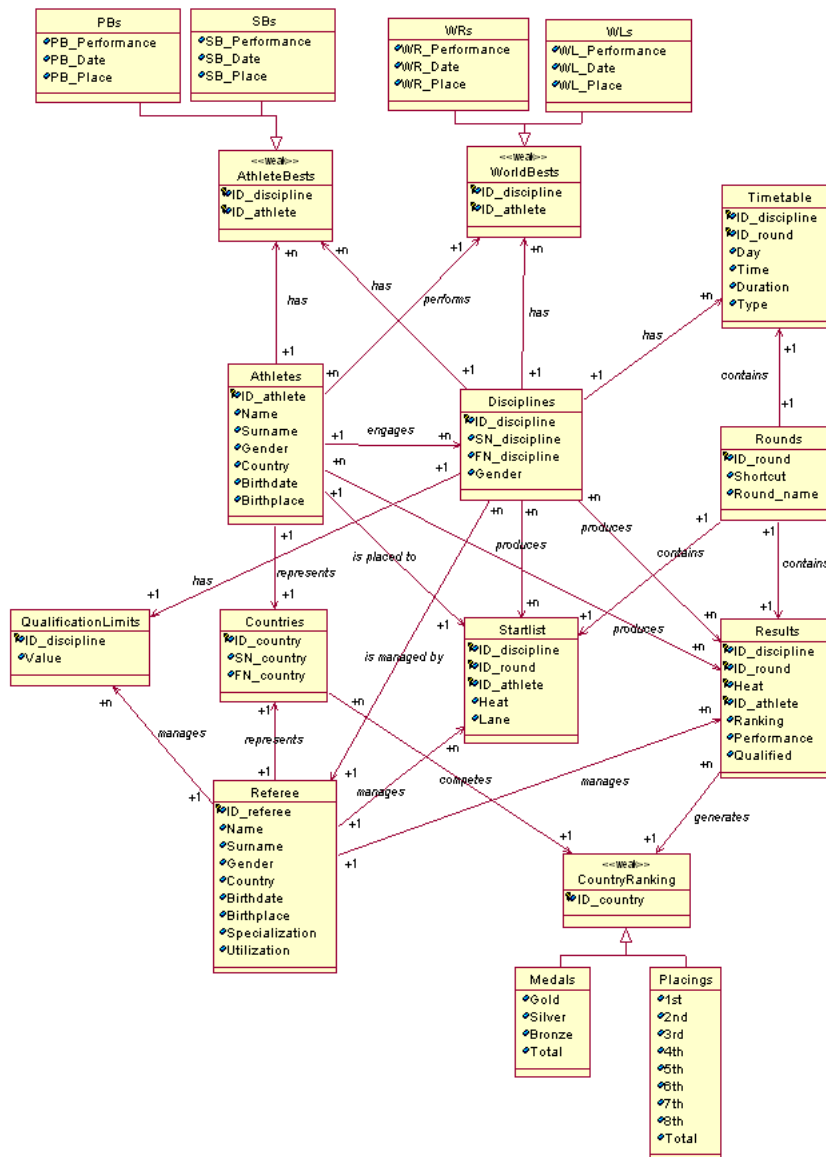
Aplikace je z hlavní části tvořena kolekcí souborů \*.php, umístěných v hlavním adresáři. Ten obsahuje taktéž tři speciální soubory – soubor `functions.js`, ve kterém jsou definovány javascriptové funkce, soubor `style.css`, který obsahuje definici kaskádových stylů a skript `sql.sql` pro vytvoření a naplnění databázových tabulek. V adresáři `images` se nachází všechny použité obrázky. V adresáři `diagrams` se nachází Use-Case a E-R diagramy. Adresář `technical_report` obsahuje kromě finální podoby technické zprávy ve formátu PDF také zdrojové kódy zprávy ve formátu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  a veškeré šablony, obrázky a diagramy použité při její tvorbě.



Obrázek A.1: Obsah přiloženého CD

# Příloha B

## E-R diagram



## Příloha C

# Nápověda

### Uživatel

Na úvodní stránce se jako uživatel musíte buď zaregistrovat nebo přihlásit vyplněním uživatelského jména a hesla.

The image contains two side-by-side screenshots of a web application interface. The left screenshot is titled "User authorization" and features a light blue background. It has a red message "User is log out." at the top. Below it are two input fields labeled "Login:" and "Password:". Underneath these fields are three radio buttons labeled "User", "Administrator", and "Referee". A "Log in" button is centered below the radio buttons. At the bottom of the form, there is a blue link labeled "Registration". The right screenshot is titled "New user registration" and has a light gray background. It contains five input fields labeled "Login:", "Password:", "Email:", "Name:", and "Surname:". Below these fields are two buttons labeled "Register" and "Close".

Při pokusu o neautorizovaný přístup se objeví varovné okno a budete přesměrováni na přihlašovací stránku.



Po úspěšném přihlášení budete přesměrováni na úvodní stránku.

My account Help User Logged as: a Logout

Calendar & Results  
Focus on Athletes  
World Records

## Timetable - By Discipline

Timetable by Discipline  
Timetable by Date  
Placing Table  
Medal Table

100m Men				
Date	Time	Round	Startlist	Results
2007-08-25	12:10	HEATS	<a href="#">Startlist</a>	<a href="#">Results</a>
2007-08-25	20:15	QUARTER-FINAL	<a href="#">Startlist</a>	<a href="#">Results</a>
2007-08-26	20:10	SEMI-FINAL	<a href="#">Startlist</a>	<a href="#">Results</a>
2007-08-26	22:20	FINAL	<a href="#">Startlist</a>	<a href="#">Results</a>
100m Women				
Date	Time	Round	Startlist	Results
2007-08-26	11:40	HEATS	<a href="#">Startlist</a>	<a href="#">Results</a>
2007-08-26	20:35	QUARTER-FINAL	<a href="#">Startlist</a>	<a href="#">Results</a>
2007-08-27	20:00	SEMI-FINAL	<a href="#">Startlist</a>	<a href="#">Results</a>

Základní rozložení bylo popsáno už dříve. Své osobní údaje, zadané při registraci, můžete změnit přes volbu „My account“ vlevo nahoře. Z aplikace se odhlásíte klepnutím na položku „Logout“ vpravo nahoře.

V pravé kontextové nabídce si můžete přepnout zobrazení časového rozvrhu podle jednotlivých soutěžních dnů nebo disciplín. V této části se také nachází odkazy na zobrazení medailových a bodových statistik.

My account Help User Logged as: a Logout

Calendar & Results  
Focus on Athletes  
World Records

### Medal table

Ranking	Country	Gold	Silver	Bronze	Total
1	Iran	0	0	0	0
2	Lithuania	0	0	0	0
3	Luxembourg	0	0	0	0
4	Mexico	0	0	0	0
5	Mold	0	0	0	0
6	Monaco	0	0	0	0
7	Morocco	0	0	0	0
8	Namibia	0	0	0	0
9	Netherlands	0	0	0	0
10	Latvia	0	0	0	0
11	Kuwait	0	0	0	0

Timetable by Discipline  
Timetable by Date  
Placing Table  
Medal Table

My account Help User Logged as: a Logout

Calendar & Results  
Focus on Athletes  
World Records

### Placing table

Ranking	Country	1st	2nd	3rd	4th	5th	6	7th	8th	Total points
1	Netherlands	0	0	0	0	0	0	0	0	0
2	Great Britain	0	0	0	0	0	0	0	0	0
3	Norway	0	0	0	0	0	0	0	0	0
4	Albania	0	0	0	0	0	0	0	0	0
5	China	0	0	0	0	0	0	0	0	0
6	Panama	0	0	0	0	0	0	0	0	0
7	Antigua and Barbuda	0	0	0	0	0	0	0	0	0
8	Guatemala	0	0	0	0	0	0	0	0	0
9	Paraguay	0	0	0	0	0	0	0	0	0
10	Argentina	0	0	0	0	0	0	0	0	0
11	Hungary	0	0	0	0	0	0	0	0	0

Timetable by Discipline  
Timetable by Date  
Placing Table  
Medal Table

U jednotlivých závodů si můžete zobrazit startovní a výsledkovou listinu. Vlevo nahoře lze přepínat mezi zobrazením všech záznamů nebo jen jednotlivých kol.

Calendar & Results  
Focus on Athletes  
World Records

### Startlist

100m Men

Referee: Butanck, Zdenek (CZE)

WR: 10:02 Kimp, Troy Marada 2003-08-02  
WL: 10:09 Barbora, Jose Luis Sylbery 2007-01-15

Entry list  
Start list Full  
Start list Final  
Start list Quarter-Final  
Start list Heats

#### Heats - Heat 1

Lane	ID	Name	Country
1	242	Changrill, Christian	GBR
2	137	Nyaga, David	KEN
3	243	Bush, Franklin	GBR
4	240	Spierberg, John	GBR
5	92	Futten, Heinz	GBR
6	48	Bradley, Don	CAN
7	248	Alphand, Richard	FRA
8	15	Mattosyan, Armen	ARM

Calendar & Results  
Focus on Athletes  
World Records

### Results

100m Men

Referee: Butanck, Zdenek (CZE)

WR: 10:02 Kimp, Troy Marada 2003-08-02  
WL: 10:09 Barbora, Jose Luis Sylbery 2007-01-15

#### Heats - Heat 1

Ranking	ID	Name	Country	Performance
1	92	Futten, Heinz	GBR	10:03
2	242	Changrill, Christian	GBR	10:08
3	243	Bush, Franklin	GBR	10:08
4	15	Mattosyan, Armen	ARM	10:42
5	240	Spierberg, John	GBR	10:63
6	137	Nyaga, David	KEN	10:72
7	48	Bradley, Don	CAN	10:77
8	248	Alphand, Richard	FRA	10:95

Results Full  
Results Final  
Results Quarter-Final  
Results Heats



Další položkou v levém menu jsou informace o atletech. Nachází se zde seznam všech závodníků, ve kterém můžete libovolně vyhledávat, filtrovat a řadit záznamy. Po kliknutí na některého atleta se vám o něm zobrazí detailní informace.

### Athletes

Name:  Discipline: All ▼

Surname:  Gender: All ▼ Reset

Country: All ▼ ID:  Search

Records: 1-20 from 256

[First](#) | [Previous](#) | [Next](#) | [Last](#)

ID	Name	Surname	Country	Discipline
70	Moonika	Aava	EST	Heptathlon Women
73	Abiyote	Abate	ETH	100m Men
74	Gezahagne	Abera	ETH	100m Men
212	Elvan	Abeylegesse	TUR	3000m Steeplechase Women
75	Berhane	Adere	ETH	Pole Vault Women
76	Yibeltal	Admasu	ETH	Long Jump Women
184	Yelena	Afanasyeva	RUS	Javelin Throw Women
3	Khoudir	Aggoune	ALG	1500m Men
3	Khoudir	Aggoune	ALG	3000m Men

#### Detail informations

Surname:	Aava	ID athlete:	70
Name:	Moonika	Birthdate:	1962-11-26
Country:	Estonia	Birthplace:	Tallin

#### Best performances

Disciplines	Best	Performance	Date	Place
Heptathlon Women	PB	6433	2006-12-29	New York
	SB	6006	2007-04-19	Moscow

Poslední dostupnou stránkou v režimu uživatele je seznam a detaily aktuálních světových rekordů.

### Records

Men	Women
<a href="#">100m Men</a>	<a href="#">100m Women</a>
<a href="#">200m Men</a>	<a href="#">200m Women</a>
<a href="#">400m Men</a>	<a href="#">400m Women</a>
<a href="#">800m Men</a>	<a href="#">800m Women</a>
<a href="#">1500m Men</a>	<a href="#">1500m Women</a>
<a href="#">5000m Men</a>	<a href="#">5000m Women</a>
<a href="#">10000m Men</a>	<a href="#">10000m Women</a>
<a href="#">Marathon Men</a>	<a href="#">Marathon Women</a>
<a href="#">110m Hurdles Men</a>	<a href="#">100m Hurdles Women</a>

### 100m Men

#### World Record

Performance	Athlete	Birthdate	Country	Place	Date
10.02	Kemp, Troy	1972-07-30	BAH	Marseille	2002-08-02

#### World Leading

Performance	Athlete	Birthdate	Country	Place	Date
10.09	Barbosa, Jose Luiz	1985-05-15	BRA	Sydney	2007-01-15

## Administrátor

Úvodní stránka administrátora je velmi podobná uživateli. Je navíc obohacena o funkce pro správu atletů, rozhodčích a uživatelských účtů.

### Athletes

Name:  Discipline:

Surname:  Gender:

Country:  ID:

Records: 1-20 from 256

[First](#) | [Previous](#) | [Next](#) | [Last](#)

ID	Name	Surname	Country	Discipline		
70	Moonika	Aava	EST	Heptathlon Women	<a href="#">Edit</a>	<a href="#">Delete</a>
73	Abiyote	Abate	ETH	100m Men	<a href="#">Edit</a>	<a href="#">Delete</a>
74	Gezahegne	Abera	ETH	100m Men	<a href="#">Edit</a>	<a href="#">Delete</a>
212	Elvan	Abeylegesse	TUR	3000m Steeplechase Women	<a href="#">Edit</a>	<a href="#">Delete</a>
75	Berhane	Adere	ETH	Pole Vault Women	<a href="#">Edit</a>	<a href="#">Delete</a>
76	Yibeltal	Admasu	ETH	Long Jump Women	<a href="#">Edit</a>	<a href="#">Delete</a>
184	Yelena	Afanasyeva	RUS	Javelin Throw Women	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Khoudir	Aggoune	ALG	1500m Men	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Khoudir	Aggoune	ALG	5000m Men	<a href="#">Edit</a>	<a href="#">Delete</a>

### Referees

Name:  Surname:

Records: 1-7 from 7

[First](#) | [Previous](#) | [Next](#) | [Last](#)

ID	Name	Surname	Country		
1	Zdenek	Buzanek	CZE	<a href="#">Edit</a>	<a href="#">Delete</a>
4	Ekaterini	Dafos	CYP	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Abdulah	Fayeh	TUR	<a href="#">Edit</a>	<a href="#">Delete</a>
6	Jim	Howard	AUS	<a href="#">Edit</a>	<a href="#">Delete</a>
7	Karin	Meier	AUT	<a href="#">Edit</a>	<a href="#">Delete</a>
2	Irina	Nikolchana	UKR	<a href="#">Edit</a>	<a href="#">Delete</a>
5	Luis	Sussez	ARG	<a href="#">Edit</a>	<a href="#">Delete</a>

[First](#) | [Previous](#) | [Next](#) | [Last](#)

### Users

Login:  User type:

Records: 1-10 from 15

[First](#) | [Previous](#) | [Next](#) | [Last](#)

Login	Password	Name	Surname	User type		
a	a	a	a	User	<a href="#">Edit</a>	<a href="#">Delete</a>
ab	a	a	a	User	<a href="#">Edit</a>	<a href="#">Delete</a>
abc	a	a	a	User	<a href="#">Edit</a>	<a href="#">Delete</a>
abcd	a	a	a	User	<a href="#">Edit</a>	<a href="#">Delete</a>
abcde	a	a	a	User	<a href="#">Edit</a>	<a href="#">Delete</a>
bedef	b	b	b	Administrator	<a href="#">Edit</a>	<a href="#">Delete</a>
bode	b	b	b	Administrator	<a href="#">Edit</a>	<a href="#">Delete</a>
bed	b	b	b	Administrator	<a href="#">Edit</a>	<a href="#">Delete</a>
bc	b	b	b	Administrator	<a href="#">Edit</a>	<a href="#">Delete</a>
b	b	b	b	Administrator	<a href="#">Edit</a>	<a href="#">Delete</a>

[First](#) | [Previous](#) | [Next](#) | [Last](#)

Dialogy pro vložení nových záznamů jsou zobrazovány po vyvolání funkcí z pravé kontextové nabídky a jsou si funkčně velmi podobné. Po stisknutí tlačítka „Reset“ se formulář vymaže, zatímco při kliknutí na „Insert“ je odeslán ke zpracování.

### Insert a new athlete

ID athlete:	<input type="text" value="253"/>			
Discipline:	<input type="text" value="----- choose -----"/>	Performance:	Date:	Place:
PB:		<input type="text"/>	<input type="text"/>	<input type="text"/>
SB:		<input type="text"/>	<input type="text"/>	<input type="text"/>
Name:	<input type="text"/>			
Surname:	<input type="text"/>			
Gender:	<input type="text" value="- choose -"/>			
Country:	<input type="text" value="----- choose -----"/>			
Birthdate:	<input type="text"/>	yyyy-mm-dd		
Birthplace:	<input type="text"/>			
<input type="button" value="Reset"/> <input type="button" value="Insert"/>				

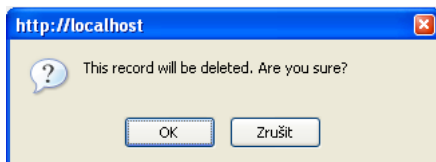
### Insert a new referee

ID:	<input type="text" value="8"/>
Name:	<input type="text"/>
Surname:	<input type="text"/>
Gender:	<input type="radio"/> Man <input type="radio"/> Woman
Country:	<input type="text" value="----- choose -----"/>
<input type="button" value="Reset"/> <input type="button" value="Insert"/>	

### Insert a new user

Login:	<input type="text"/>
Password:	<input type="text"/>
Name:	<input type="text"/>
Surname:	<input type="text"/>
Email:	<input type="text"/>
User type:	<input type="radio"/> User <input type="radio"/> Administrator <input type="radio"/> Referee
<input type="button" value="Reset"/> <input type="button" value="Insert"/>	

Administrátorovi jsou nabídnuty také funkce editaci a smazání stávajícího záznamu. Před jakýmkoli smazání je uživatel upozorněn.



## Rozhodčí

Rozhodčímu jsou dostupné všechny funkce běžného uživatele, navíc má na starosti správu startovních listin, výsledků a kvalifikačních limitů. Jeho hlavní stránka vypadá následovně.

Calendar & Results  
Qualification limits  
Focus on Athletes  
World Records

Timetable by Discipline  
Timetable by Date  
Placing Table  
Medal Table

### Timetable - By Discipline

100m Men										
Date	Time	Round	Startlist				Results			
2007-08-25	12:10	HEATS	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit
2007-08-25	20:15	QUARTER-FINAL	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit
2007-08-26	20:10	SEMI-FINAL	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit
2007-08-26	22:20	FINAL	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit

100m Women										
Date	Time	Round	Startlist				Results			
2007-08-26	11:40	HEATS	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit
2007-08-26	20:35	QUARTER-FINAL	Sort	Clean	Startlist	Add	Generate	Clean	Results	Edit

U jednotlivých závodů jsou funkce pro správu startovních listin a výsledků. Kliknutím na „Sort“ jsou závodníci nasazeni do jednotlivých drah, pomocí „Add result“ vyvoláme dialog pro zadání nového výsledku, „Edit“ nám dovolí výsledky upravit a dvojice příkazů „Clean“ vyčistí startovní nebo výsledkovou listinu.

### Insert result

Discipline: 100m Men  
Round: HEATS  
Heat: 1

Lane	Name	Country	Performance	Special values
1	Campwell, Christian	GBR		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
2	Nyaga, David	KEN		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
3	Bush, Franklin	GBR		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
4	Spielberg, John	GBR		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
5	Futterer, Heinz	GER		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
6	Bailey, Don	CAN		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
7	Alphand, Richard	FRA		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
8	Martirosyan, Armen	ARM		<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ

Reset Insert

### Edit result

Discipline: 100m Men  
Round: HEATS  
Heat: 1

Lane	Name	Country	Performance	Special values
1	Futterer, Heinz	GER	10.03	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
2	Campwell, Christian	GBR	10.08	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
3	Bush, Franklin	GBR	10.08	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
4	Martirosyan, Armen	ARM	10.42	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
5	Spielberg, John	GBR	10.63	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
6	Nyaga, David	KEN	10.72	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
7	Bailey, Don	CAN	10.77	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ
8	Alphand, Richard	FRA	10.95	<input type="radio"/> DNS <input type="radio"/> DNF <input type="radio"/> DSQ

Reset Update

Poslední speciální funkcí pro rozhodčí je správa kvalifikačních limitů.

## Qualification limits

Discipline	Limit	Discipline	Limit
High Jump Men	<input type="text" value="230"/>	High Jump Women	<input type="text" value="195"/>
Pole Vault Men	<input type="text" value="580"/>	Pole Vault Women	<input type="text" value="450"/>
Long Jump Men	<input type="text" value="800"/>	Long Jump Women	<input type="text" value="670"/>
Triple Jump Men	<input type="text" value="17.00"/>	Triple Jump Women	<input type="text" value="14.00"/>
Shot Put Men	<input type="text" value="20.00"/>	Shot Put Women	<input type="text" value="18.00"/>
Discus Throw Men	<input type="text" value="64.00"/>	Discus Throw Women	<input type="text" value="60.00"/>
Hammer Throw Men	<input type="text" value="77.00"/>	Hammer Throw Women	<input type="text" value="65.00"/>
Javelin Throw Men	<input type="text" value="80.00"/>	Javelin Throw Women	<input type="text" value="60.00"/>