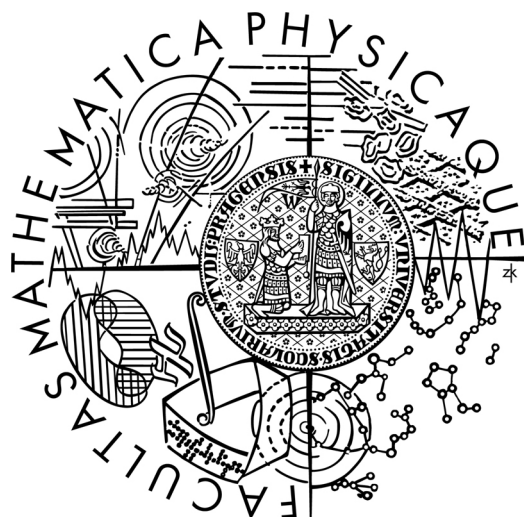


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Pavol Koman

Virtuální sázková kancelář

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Martin Trčka
Studijní program: Informatika, Programování

2008

Chcel by som poďakovať vedúcemu tejto bakalárskej práce pánovi Mgr. Martinovi Trčkovi za jeho podnety, pripomienky a pomoc pri vypracovaní tejto práce.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 27.9.2008

Pavol Koman

Obsah

1.	Úvod.....	4
2.	Zoznam očakávaných vlastností	5
3.	Analýza riešenia	6
3.1.	Získavanie kurzov športových udalostí.....	6
3.1.1.	Možnosti.....	6
3.1.2.	Príklady kurzových lístkov	6
3.1.3.	Hlavné problémy	7
3.2.	Načítanie údajov.....	7
3.3.	Rozlišovanie zhodných udalostí.....	8
3.3.1.	Nájdanie úplne zhodného zápasu:	8
3.3.2.	Vyhľadanie čiastočne zhodných zápasov	8
3.4.	Odstránenie nežiadúcich udalostí.....	9
3.5.	Automatizované získavanie údajov.....	9
3.6.	Popis konfiguračných súborov	9
3.6.1.	Konfiguračný súbor stávkovej kancelárie	9
3.6.2.	Hlavný konfiguračný súbor.....	10
3.7.	Konfigurovateľný parser	10
3.8.	Získavanie výsledkov.....	10
4.	Implementácia	12
4.1.	Výber jazyka a vývojového prostredia.....	12
4.2.	Databázová vrstva	12
4.2.1.	Popis databázy.....	12
4.2.2.	Triggery	13
4.2.3.	Periodicky sa opakujúce úkony.....	13
4.2.4.	Implementácia aliasov v DB	13
4.3.	Zabezpečenie aplikácie	14
4.3.1.	Bezpečnosť hesiel	14
4.4.	Implementácia konfiguračných súborov	15
4.4.1.	Konfiguračný súbor stávkovej kancelárie	15
4.4.2.	Hlavný konfiguračný súbor.....	16
4.5.	Implementácia získavania údajov z kurzových lístkov.....	17
4.5.1.	Trieda MyParser	17
4.5.2.	Trieda MyMatcher	18
4.5.3.	Získavanie údajov	18
4.5.4.	Príklad získavania dát	19
4.6.	Zhodnosť zápasov	21
4.7.	Automatizované získavanie údajov.....	21
5.	Záver	22
	Literatúra.....	23
	Prílohy	24

Název práce: Virtuální sázková kancelář
Autor: Pavol Koman
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: Mgr. Martin Trčka
E-mail vedoucího: Martin.Trcka@mff.cuni.cz

Abstrakt: Cílem práce je navrhnout a implementovat webovou aplikaci, která bude plnit funkci virtuální (na zkoušku) sázkové kanceláře. Aplikace bude získávat sázkové kurzy a výsledky zápasů z webového rozhraní reálných sázkových kanceláří (alespoň 2 sázkové kanceláře) a z nich bude tvořit nabídku virtuální kanceláře. Případy nekonzistence nebo nejasnosti nabídek vyřídí ručně administrátor systému, aplikace by se však měla snažit omezit počet těchto případů na minimum. Aplikace umožní vyhodnocení sázkových tiketů u jednotlivých reálných sázkových kanceláří a provede tak srovnání jejich nabídky. Mezi další vlastnosti patří podpora uživatelských účtů, archivace, správa a vyhledávání tiketů v databázi, export tiketů do formátu PDF.

Klíčová slova: sázková kancelář, sázky

Title: Virtual Betting Shop
Author: Pavol Koman
Department: Institute Department of Software Engineering
Supervisor: Mgr. Martin Trčka
Supervisor's e-mail address: Martin.Trcka@mff.cuni.cz

Abstract: Work is aimed to design and implement web application called virtual sportsbook. Application will be able to get match odds and results from real sportsbook's web pages (at least two sportsbooks) and create own offering by virtue of real data. Administrator will resolve conflict cases but application will try to minimize number of various conflicts. Application will be able to evaluate bet value from real sportsbooks and compares their offerings. Application will be able to archive, manage and search bets in database, register users and export bets into PDF file.

Keywords: sportsbook, bookmaker

1. Úvod

So zvyšujúcou sa dostupnosťou internetu narastá význam webových aplikácií. Množstvu ľudí dennodenne rôznymi spôsobmi uľahčujú život. Niektoré činnosti je možné, dokonca aj výhodné nahradiť ich internetovou obdobou. Jednou z týchto činností je tipovanie výsledkov športových udalostí cez internet. Vzhľadom k množstvu ľudí ktorí sa tipovaniu venujú stúpa aj obľuba stránok kde je možné tipovať iba zo zábavy. Na takýchto weboch užívateľ zistí ako by sa mu darilo ak by do svojho „hobby“ investoval skutočné peniaze.

Cieľom tejto bakalárskej práce je vytvoriť podobnú „virtuálnu“ stávkovú kanceláriu. Mala by umožňovať užívateľom tipovanie na výsledky športových udalostí za virtuálne body s reálnymi kurzami stávkovej kancelárie. Snažil som sa dosiahnuť aby aplikácia nebola z administrátorského hľadiska veľmi časovo náročná na obsluhu. To by malo byť dosiahnuté automatizovaným spracúvaním kurzových lístkov, s tým že administrátor bude riešiť sporné prípady.

2. Zoznam očakávaných vlastností

- Registrácia užívateľov
- Tipovanie výsledkov športových udalostí
- Vyhodnotenie kurzu tiketu u podporovaných skutočných stávkových kancelárií
- Archív tiketov
- Štatistiky úspešnosti
- Grafické znázornenie niektorých štatistík
- Porovnanie štatistík dvoch užívateľov
- Export do pdf
- Automatizované a ručné získavanie kurzov zápasov
- Automatizované a ručné získavanie výsledkov zápasov
- Automatizované vyhodnocovanie zápasov
- Automatizované vyhodnocovanie tiketov
- Detekcia rovnakých zápasov u rôznych stávkových kancelárií
- Správa užívateľov
- Správa stávkových kancelárií
- Konfigurácia jednotlivých stávkových kancelarií
- Správa výsledkov
- Správa teamov

3. Analýza riešenia

3.1. Získavanie kurzov športových udalostí

3.1.1. Možnosti

Jedným zo základných problémov je získavanie zdrojových dát. To je možné vyriešiť viacerými spôsobmi.

1. Ručné zadávanie dát správcom aplikácie,
2. Predplatenie si prístupu k dátam od služieb ktoré tieto dáta poskytujú,
3. Automatické získavanie dát z dostupných zdrojov.

Prvá možnosť, a to zadávanie dát ručne správcom aplikácie je implementačne najjednoduchšia, ale zároveň najnáročnejšia na čas strávený správou aplikácie.

Druhú možnosť (predplatenie si prístupu k dátam) som netestoval, ale už z povahy služby predpokladám že by to malo byť jednoducho integrovateľné do aplikácie a časovo nenáročné na správu. Služby poskytujúce dáta zabezpečujú ich aktuálnosť a všetky dáta z rôznorodých zdrojov dávajú k dispozícii v ich vlastnom jednotnom formáte. Najčastejšie je týmto formátom xml. Na druhej strane je v tomto prípade nevýhoda finančná náročnosť, čo môže byť u začínajúcich webových aplikácií veľmi podstatné.

Tretia možnosť (automatické získavanie dát z dostupných zdrojov) je kompromisom medzi predchádzajúcimi dvomi variantami. Dáta sa získavajú z kurzových lístkov ktoré majú reálne stávkové kancelárie zverejnené na stránke v txt verzii.

V čase vymýšľania zadania bc. práce (rok 2006) mala takúto verziu kurzového lístka väčšina stávkových kancelárií, avšak doteraz (rok 2008) už txt formát kurzového lístka je zverejnený iba na weboch niekoľkých stávkových kancelárií. V tomto prípade sa jedná vlastne o kompromis medzi prvým a druhým spôsobom získavania dát, keď vlastné voľne dostupné dáta sú sťahované z internetu a automatizovane spracované.

Správca stránky iba zasahuje do riešenia v sporných prípadoch. Práve toto tretie riešenie bolo zvolené pre implementáciu.

3.1.2. Príklady kurzových lístkov

Časť kurzového lístka SK Chance:

Fotbal / Anglie / Dohravky / 11.tyden							
	12.3.20:45	1	0	2	10	12	02
938	Aston Villa-Middlesbrough	1.62	3.5	4.8	1.1	1.21	2.02
940	Portsmouth-Birmingham	1.72	3.4	4.25	1.14	1.22	1.88
	12.3.21:00						
939	Chelsea-Derby	1.1	8	13	-	1.01	4.95
Fotbal / Ceska rep. / 1.liga / 20.kolo							
	10.3.17:15	1	0	2	10	12	02
107 +	Sparta-Teplice	1.42	3.8	7	1.03	1.18	2.46

Časť kurzového lístka SK Victoria-tip:

Fotbal / Copa Libertadores CF 1.z.

číslo 1 10 0 20 2 AKO datum

1591 AMERICA -SANTOS 2,25 1,30 3,10 1,50 2,90 15.05 22:00

1592 SAN LORENZO-LDU QUITO 1,40 1,06 3,90 2,51 7,12 15.05 22:00

1593 SAO PAULO -FLUMINENSE 1,80 1,16 3,30 1,80 3,96 14.05 22:00

1594 BOCA JUNIOR-ATLAS GUADA 1,35 1,04 4,30 2,69 7,25 14.05 22:00

Fotbal / Major League Soccer

číslo 1 10 0 20 2 AKO datum

821 COLUMBUS -SALT LAKE 1,75 1,15 3,40 1,85 4,07 0 15.05 22:00

3.1.3. Hlavné problémy

Za najvhodnejšie riešenie som zvolil tretiu možnosť, teda získavanie kurzov z dostupných textových zdrojov.

Najzásadnejšie problémy pri tomto spôsobe získavania kurzov sú:

- získavanie dát z rôznorodých zdrojov
- rozlíšenie spolu súvisiacich dát
- rozumná možnosť ovplyvňovania toho, ktoré športové udalosti chceme načítať, a ktoré už nie
- automatizácia získavania kurzov

Analýza týchto problémov je v nasledujúcom texte.

3.2. Načítanie údajov

Pri vymýšľaní vlastného spôsobu načítania dát z textového súboru som uvažoval nad viacerými možnosťami.

1. Na základe určitého formátu kurzového lístka u každej stávkovej kancelárie načítať presne dáta z určitých pozícií v riadku (reťazci). Problém je v tom, že niektoré SK nemajú úplne presne daný formát jednotlivých riadkov. Napríklad u SK Chance sú teamy oddelené pomlčkou, pričom iba od dĺžky názvu prvého teamu závisí pozícia začiatku názvu druhého teamu.
2. Využitie regulárnych výrazov. Regulárne výrazy umožňujú získavať rozličné dáta nezávisle od presnej pozície v reťazci, čo je veľká výhoda oproti predchádzajúcemu spôsobu.

Kvôli potrebe jednotného systému získavania údajov z kurzového lístka som zvolil možnosť využitia regulárnych výrazov.

Keďže každá stávková kancelária má vlastný formát kurzového lístka, je potrebné kvôli jednotnému postupu nutné formát kurzového lístka nejako abstraktne opísať. Na popis formátu sa využívajú regulárne výrazy a „pravidlá“ riadiace čítanie kurzového lístka. Regulárne výrazy sa používajú na rozlišovanie jednotlivých riadkov a pre získavanie dát.

Pre každú stávkovú kanceláriu využívanú v aplikácii sa musí vytvoriť konfiguračný súbor, v ktorom sú tieto pravidlá a regulárne výrazy špecifikované. Kvôli predpokladu že určitá časť nastavení bude zhodná u viacerých stávkových kancelárií je vytvorený ešte jeden konfiguračný súbor, nazvaný „hlavný

konfiguračný súbor“, ktorý obsahuje spoločné konfigurácie využívané viacerými stávkovými kancelárkami.

Pri čítaní kurzového lístka je potrebné načítať nasledujúce dáta:

- čas a dátum zápasu – potrebujeme vedieť kedy sa začína zápas
- názvy teamov – potrebujeme vedieť kto hraje
- názov športu – potrebný kvôli odfiltrovaní nepodporovaných športov
- názov ligy – potrebný kvôli odfiltrovaní nechcených líg. Ako názov ligy označujeme celý názov ktorým je liga označená u stávkovej kancelárie.
- id zápasu – id ktoré má daný zápas u našej hlavnej stávkovej kancelárie. Je potrebné kvôli neskoršiemu získavaniu výsledkov. Výsledky zverejňované stávkovou kanceláriou sú identifikované práve týmto Id.
- kurzy vypísané na jednotlivé možné výsledky (výhra domácich, remíza, výhra hostí atď).

Aplikácia pracuje iba s najbežnejším druhom tipov, a to s tipmi na výhru/prehru/remízu/nevýhru/neprehru domácich.

Iné druhy tipov (handicapy, na počty gólov/minút atď.) nie sú podporované. Je to hlavne kvôli veľkému zvýšeniu rozsahu implementácie a tiež kvôli malej popularite týchto druhov. Rozsah implementácie zvyšuje hlavne odlišné vyhodnocovanie výsledkov u jednotlivých druhov tipov.

3.3. Rozlišovanie zhodných udalostí

Pri pridávaní kurzov udalostiam ktoré už máme načítané zo zdrojovej kancelárie je potrebné sa vyrovnat' s problémom identifikácie zhodných športových udalostí. Riešenie tohoto problému je postavené na slovníku aliasov názvov teamov.

Alias názvov znamenajú že team má priradených viacero názvov(aliasov), z ktorých jeden sa používa ako oficiálny názov (zobrazuje sa na webe). Ostatné aliasy slúžia na identifikáciu toho že sa jedná o ten istý team.

Napríklad: Zdrojová stávková kancelária používa názov teamu „Manchester Utd.“, iná stávková kancelária používa názov teamu „Man. Utd.“. Po prvotnom priradení aliasu „Man. Utd.“ teamu s oficiálnym aliasom „Manchester Utd.“ sa s oboma názvami pracuje ako s tým istým teamom.

Automaticky sa rozlišuje vo dvoch stupňoch presnosti:

3.3.1. Nájdenie úplne zhodného zápasu:

Nájde v DB zápas so zhodným športom, dátumom zhodným na minútu a so zhodnými domácimi teamami a teamami hostí. V prípade že sa takýto zápas nájde, pridáme mu pridávaný kurz a pokračujeme pridávaním ďalšieho kurzu.

Ak sa takýto zápas nenájde, tak nasleduje druhý stupeň:

3.3.2. Vyhľadanie čiastočne zhodných zápasov

Za čiastočne zhodný je považovaný zápas so zhodným športom, dátumom

zhodným na minútu a zhodným teamom domácich ALEBO teamom hostí. U teamu u ktorého nebola nájdená zhodnosť sa najprv pridá názov teamu zo spracúvaného zápasu ako alias teamu ktorý máme v DB. Týmto dosiahneme úplnú zhodnosť. Potom sa nájdenému zápasu pridajú kurzy spracúvaného zápasu a pokračuje sa spracúvaním ďalšieho zápasu.

Ak sa ani takýto zápas nenájde, spracúvaná udalosť sa uchová ako udalosť s nepriradeným kurzom a o priradení k zápasu v databáze rozhodne administrátor.

3.4. Odstránenie nežiadúcich udalostí

Odfiltrovanie športových udalostí ktoré nás nezaujímajú je dosiahnuté definovaním dvoch typov položiek v hlavnom konfiguračnom súbore. A to sú názov športu a zakázaný podreťazec v názve ligy. Športová udalosť sa nespracúva v prípade ak sa názov športu nenachádza medzi deklarovanými v hlavnom konfiguračnom súbore, alebo ak názov ligy obsahuje nejaký podreťazec deklarováný v hlavnom konfiguračnom súbore.

Napríklad v prípade deklarácie povoleného športu „Fotbal“ a zakázaných podreťazcov „polocas“ a „pocet голу“ nebudú spracúvané žiadne športové udalosti ktoré ako šport nemajú uvedený „Fotbal“. Ďalej z tých udalostí ktoré majú ako šport uvedený „Fotbal“ sa nespracúvajú ani tie, ktoré v názve ligy obsahujú reťazec „polocas“ alebo reťazec „pocet голу“.

3.5. Automatizované získavanie údajov

Automatizácia získavania kurzov je zavedená pre zjednodušenie údržby aplikácie. Správcovi aplikácie tak odpadá povinnosť pravidelne sťahovať aktuálne kurzové lístky a vkladať ich do aplikácie. Stačí ak zabezpečí periodické vykonávanie určitej akcie, vďaka čomu sa bude ponuka zápasov v aplikácii udržiavať aktuálna.

Touto akciou sa myslí volanie webovej adresy (stránky), ktorej spustenie vyvolá získavanie údajov.

3.6. Popis konfiguračných súborov

3.6.1. Konfiguračný súbor stávkovej kancelárie

Konfiguračný súbor stávkovej kancelárie obsahuje popis štruktúry kurzového lístka danej SK a regulárne výrazy využívané pre prechod kurzovým lístkom z získanie dát z neho.

Na popis štruktúry kurzového lístka slúžia tzv. pravidlá. Pravidlá definujú štruktúru kurzového lístka a riadi sa nimi algoritmus pre získavanie dát z neho.

Na popis tvaru jednotlivých riadkov a dát v týchto riadkoch obsiahnutých slúžia pomenované regulárne výrazy spolu s označením pozícií na ktorých sa získavané dáta na riadku nachádzajú.

3.6.2. Hlavný konfiguračný súbor

Hlavný konfiguračný súbor obsahuje definície čiastkových regulárnych výrazov, povolených športov a zakázaných podreťazcov v názvoch líg. Tento konfiguračný súbor sa využíva pri získavaní dát z kurzových lístkov všetkých stávkových kancelárií.

Čiastkové regulárne výrazy sú vlastne regulárne výrazy s prideleným názvom. Využívajú sa pri vytváraní konečných regulárnych výrazov používaných na parsovanie kurzového lístka.

V definíciách povolených športov máme vymenované športy na ktoré je možné v našej aplikácii tipovať.

Definovaním zakázaných podreťazcov pre názvy líg dosiahneme to, že po načítaní dát z kurzového lístka a odstránení zápasov z nepodporovaných športov ešte v rámci povolených športov môžeme odfiltrovať celé skupiny zápasov na základe podreťazcov obsiahnutých v názve „ligy“.

Ako názov ligy označujeme celý názov ktorým je liga označená u stávkovej kancelárie. Napríklad „Italie / 2.liga / 41.kolo“ alebo „EURO 2008 / Lepsi ve skupine / Skupina_A“.

3.7. Konfigurovateľný parser

Konfiguračnými súbormi sa riadi získavanie dát z kurzových lístkov. Vlastné získavanie dát zabezpečuje modul – parser. Úlohou tohoto modulu je spracovať kurzové lístky a vrátiť objekty reprezentujúce jednotlivé športové udalosti.

Beh parsera sa riadi pravidlami a regulárnymi výrazmi zapísanými v konfiguračnom súbore stávkovej kancelárie. Tieto regulárne výrazy sa aplikujú na jednotlivé riadky kurzového lístka na základe pravidiel riadiacich prechod.

Tento postup som vymyslel kvôli tomu že textové kurzové lístky majú jasnú a pomerne jednoduchú štruktúru ktorá sa dá skvele popísať pravidlami.

Získavanie dát priamo z HTML kódu nebolo od začiatku vôbec plánované, preto ani nie sú podporované multiline regulárne výrazy.

3.8. Získavanie výsledkov

Výsledky športových udalostí je možné dopĺňať dvomi spôsobmi.

1. Ručne jednotlivito doplniť výsledok každej športovej udalosti.
2. Automatizovane – stiahnuť textový súbor s výsledkami a ten spracovať.

V aplikácii je umožnená práca s výsledkami oboma spôsobmi.

Automatizované získavanie výsledkov znamená to, že po zavolaní určitej adresy alebo po vyvolaní akcie z administrátorskej časti sa spracuje textový súbor s výsledkami zápasov.

U tohoto súboru sa predpokladá že je vo formáte jedna športová udalosť na jeden riadok. Regulárnymi výrazmi sa na každom riadku prečíta ID športovej udalosti ktoré mala pridelené u zdrojovej stávkovej kancelárie, skóre domácich a skóre

hostí. Pre každý takto načítaný riadok sa potom podľa získaného ID vyhľadá v databáze uzatvoriteľná športová udalosť a doplní sa jej načítaný výsledok.

Ďalšia možnosť je stornovanie športovej udalosti. V tomto prípade sa so športovou udalosťou pracuje ako s výhernou s kurzom 1.0

3.8.1. Opakované využívanie rovnakých ID stávkovými kancelármi

Stávkové kancelárie využívajú na označovanie vypísaných športových udalostí zvyčajne iba obmedzený počet id (do 10000). Po určitej dobe od zverejnenia výsledku nejakej športovej udalosti je s jej pôvodným id vypísaná ďalšia športová udalosť.

Tieto Id používané u zdrojovej stávkovej kancelárie sú dôležité pri vyhodnocovaní zápasov. Podľa nich sa pridávajú výsledky jednotlivým zápasom.

Tento moment je zdroj možných problémov. V prípade že sme dlhšiu dobu iba sťahovali nové zápasy a nesťahovali výsledky ani neuzatvárali jednotlivé zápasy sa pravdepodobne stane že bude v DB viacero zápasov bez výsledku s rovnakým Id zdrojovej stávkovej kancelárie.

Pri automatickom sťahovaní výsledkov sa potom týmto zápasom výsledok nepriradí, keďže už nevieme ktorému zápasu mal tento výsledok patriť. Riešením tejto situácie je ručné pridanie výsledkov týmto zápasom (alebo stornovanie týchto zápasov).

4. Implementácia

4.1. Výber jazyka a vývojového prostredia

Pri vývoji webových aplikácií sa dnes využívajú hlavne jazyky PHP, ASP.Net a Java (rôzne frameworky). Ja som zvolil Javu vzhľadom k tomu že počas štúdia som s ňou už pracoval a v budúcnosti by som s ňou chcel pracovať aj v zamestnaní. V rámci javy som zvolil framework JSF (Java Server Faces). Jedná sa o moderný a jednoduchý framework na tvorbu webových aplikácií. Uprednostnil som ho pred kombináciou JSP (Java Server Pages) + servlety kvôli tomu že je mladší a ja osobne si myslím že v budúcnosti bude populárnejší ako spomenutá kombinácia JSP a servletov.

Za vývojové prostredie som zvolil Sun Java Studio Creator 2, ktoré poskytuje dostatok možností a funkcií k tvorbe celej práce. V čase vymýšľania témy bakalárskej práce to bolo najkomfortnejšie prostredie pre vývoj JSF aplikácií. Dnes ho už firma SUN nepodporuje a všetka funkčnosť bola integrovaná do prostredia NetBeans. Prostredie NetBeans som využil pri vývoji časti aplikácie (samostatný package) na získavanie dát z kurzových lístkov.

Pre prácu na databázovej vrstve som využil prostredie SQL Developer od firmy Oracle.

Na správu verzií som použil program Subversion (SVN). Tento program ukladá po každej zmene súbory so zdrojovými kódmi pod poradovým číslom verzie a dokáže zobrazit' všetky zmeny, ktoré v nich boli vykonané oproti predošlej verzii. V prípade, že sa pri programovaní dostala do programu neočakávaná neznáma chyba, je nám umožnené zobrazit' všetky zmeny vykonané od poslednej funkčnej verzie, prípadne sa k tejto verzii hneď vrátiť a pokračovať v práci na nej.

Pri písaní aplikácie som ešte využil knižnicu iText na generovanie pdf tiketov a knižnicu Jsf Chart Creator na generovanie grafov zobrazených na stránke so štatistikami.

4.2. Databázová vrstva

4.2.1. Popis databázy

Databáza je tvorená osemnástimi tabuľkami. Nad týmito tabuľkami je vytvorené množstvo pohľadov (views). Tieto pohľady sa využívajú prevažne pre zobrazovanie dát v tabuľkách na prezentačnej vrstve (web stránke). Ďalšie ich využitie je zjednodušenie sql dotazov.

Ďalej je vytvorený package TIPSERVER_PKG obsahujúci uložené procedúry (stored procedures) a funkcie (stored functions) využívané hlavne na volania z aplikačnej vrstvy. Zjednodušuje sa tým komunikácia s databázou, pretože namiesto spúšťania väčšieho množstva sql dotazov z aplikačnej vrstvy a spracúvania medzivýsledkov rovnako na aplikačnej vrstve je tento kód vo väčšine prípadov možné nahradiť volaním jedinej uloženej procedúry.

Tiež bola vytvorená agregáčna funkcia *MULTIPLY*. Je to vlastne obdoba štandardnej funkcie *SUM* umožňujúca násobenie. Jej vývoj bol potrebný kvôli elegantnému použitiu v niektorých dotazoch a tiež kvôli tomu, že Oracle neposkytuje agregáčnu funkciu vykonávajúcu násobenie.

4.2.2. Triggery

V aplikácii je potrebné v niektorých prípadoch pri vyvolaní nejakej akcie užívateľom vždy vyvolať inú akciu. Keďže v našom prípade sa všetky tieto akcie vykonávajú na úrovni databázovej vrstvy, pre implementáciu požadovanej funkčnosti sú využité tzv. triggery.

Triggery sú databázové procedúry ktoré sa automaticky vykonajú ako reakcia na určenú udalosť nad nejakou databázovou tabuľkou. V mojom programe sú triggery využité na vyhodnotenie tipov na určitý zápas po vložení výsledku tohto zápasu, na zapísanie zmeny v stave bodov na účte po výhre a vsadení tiketu a na počiatočné nastavenie stavu účtu pri vytvorení nového užívateľa.

4.2.3. Periodicky sa opakujúce úkony

Pre lepšiu súťažnú funkciu aplikácie je dobré aby každý súťažný mesiac štartovali všetci užívatelia za rovnakých podmienok. Preto je potrebné všetkým užívateľom každý mesiac nastaviť rovnaký štartovací počet bodov. Ďalej je pre zvýšenie hrateľnosti každý deň pripisovaný určitý počet bonusových bodov všetkým užívateľom. Keďže tieto udalosti sa pravidelne opakujú, je vhodné ich vykonávať automatizovane, bez zásahov administrátora.

Na implementáciu periodicky sa opakujúcich akcií sa využíva funkčnosť poskytovaná balíčkom `dbms_scheduler`. Tento balíček procedúr a funkcií je obsiahnutý v každej inštalácii databázového systému Oracle. Poskytuje nám funkčnosť umožňujúcu pravidelné opakované spúšťanie nejakej procedúry, anonymného kódu alebo dokonca aj binárnych súborov a shell scriptov.

4.2.4. Implementacia aliasov v DB

Názvy teamov a ich aliasy sú implementované dvomi tabuľkami, a to `T_TEAM` a `T_ALIAS`. V tabuľke `T_TEAM` sú zaujímavé iba stĺpce `TEAM_ID` a `TEAM_ALIAS_ID`. `TEAM_ID` určuje jedinečné id každého teamu na ktorý sa odkazujú športové udalosti. `TEAM_ALIAS_ID` je cudzí kľúč do tabuľky `T_ALIAS` označujúci hlavný (v aplikácii využívaný) názov teamu. Z tabuľky `T_ALIAS` nás zaujímajú stĺpce `ALIAS_ID` – id aliasu, `ALIAS_TEAM_ID` – id teamu ktorému patrí alias a `ALIAS_NAZOV_TX` – alias názvu teamu. Vyhľadávanie teamu a jeho používaného názvu prebieha tak, že v tabuľke `T_ALIAS` sa podľa `ALIAS_NAZOV_TX` nájde id teamu ktorému patrí tento alias. Potom sa môže podľa id teamu vybrať a použiť z tabuľky `T_TEAM` id a názov oficiálneho aliasu.

4.3. Zabezpečenie aplikácie

Časť aplikácie je prístupná iba prihláseným užívateľom. Toho som dosiahol jednoduchou metódou, štýlom že na každej verejne neprístupnej stránke overím či ju načíta prihlásený užívateľ. V prípade že nie tak ho okamžite presmerujem na chybovú stránku.

4.3.1. Bezpečnosť hesiel

Keďže som sa rozhodol pre vlastnú implementáciu zabezpečenia, tak mám heslá pod kontrolou vo svojej databázi. Pri ukladaní hesiel využívam hashovanie šifrou SHA-1 produkujúcou 160 bitový hash. Táto šifra je implementovaná v databázovom systéme Oracle (package dbms_crypto).

Pre ochranu pred jednoduchým slovníkovým útokom, pri ktorom útočník má predgenerovanú databázu hashov hesiel sa používa metóda „Salting“. Salting spočíva v tom, že ku každému heslu sa vygeneruje nejaký náhodný reťazec (salt), a hashuje sa výsledok zretazenia salt + heslo. Do databáze sa ukladá až výsledný hash reťazca salt + heslo. Spolu s týmto hashom sa ukladá do iného stĺpca aj samotný salt, aby sme pri prihlasovaní mohli zase porovnať zhodu hashu salt + zadávané heslo s hashom uloženým v databáze.

4.3.2. SQL Injection

SQL injection je hackerská metóda ktorá využíva nedostatky (chyby programátora) v kontrole vstupných dát k tomu aby mohol hacker podstrčiť namiesto očakávaných informácií vlastné a prebrať tak kontrolu nad databázou nad rámec svojich práv. Tým prebratím kontroly je myslená úplná kontrola nad databázou kde hacker môže databázu kopírovať, mazať, pridávať ľubovoľné údaje alebo krahnúť dôležité informácie.

V Jave je možné vyhnúť sa SQL Injection používaním objektu PreparedStatement na prístup k databáze. Podstatné je pri vkladaní vstupu od užívateľa alebo obsahu premenných využívať parametrizované dotazy, a hodnotu jednotlivých parametrov nastavovať volaním metód setXXX objektu PreparedStatement. Takto predávaný vstup je automaticky ošetrovaný (escaped) JDBC ovládačom.

Príklad:

```
String selectStatement = "SELECT * FROM User WHERE userId = ? ";
PreparedStatement prepStmt = con.prepareStatement(selectStatement);
prepStmt.setString(1, userId);
ResultSet rs = prepStmt.executeQuery();
```

V aplikácii používam práve tento spôsob práce s databázou, čo mi zaručuje odolnosť aplikácie proti tomuto typu útoku.

4.4. Implementácia konfiguračných súborov

4.4.1 Konfiguračný súbor stávkovej kancelárie

Konfiguračný súbor stávkovej kancelárie obsahuje popis štruktúry kurzového lístka danej SK a regulárne výrazy využívané pre prechod kurzovým lístkom a získanie dát z neho.

Na popis štruktúry kurzového lístka slúžia tzv. pravidlá. Pravidlá definujú štruktúru kurzového lístka a riadi sa nimi algoritmus pre získavanie dát z neho.

4.4.1.1. Pravidlá

Každé pravidlo je má tvar HLAVICKA_PRAVIDLA { TELO_PRAVIDLA }. HLAVICKA_PRAVIDLA je názvom tohto pravidla, musí byť v rámci tohto konfiguračného súboru jedinečná, a musí začínať reťazcom „RULE_“.

V TELO_PRAVIDLA je zadeklarovaná činnosť ktorú má vykonať program. Táto činnosť sa môže skladať z jedného alebo dvoch krokov.

Prvým krokom činnosti môže byť buď aplikácia regulárneho výrazu zadaného jeho názvom alebo spracovanie pravidla zadaného názvom hlavičky.

Druhým krokom môže byť iba spracovanie pravidla (teda druhú časť tela pravidla tvorí názov hlavičky ďalšieho pravidla).

Jedná sa v podstate o rekurzívne spracúvanie jednotlivých pravidiel ukončované neúspešnou aplikáciou regulárnych výrazov v prvých krokoch pravidiel. Postup spracovania kurzového lístka sa tak vlastne rekurzívne riadi tým ako na seba odkazujú pravidlá.

Pravidlo ktorým sa začína musí byť vždy pomenované RULE_START.

4.4.1.2. Regulárne výrazy

Regulárne výrazy slúžia na získavanie dát z kurzového lístka. Aplikáciou regulárnych výrazov sa tiež vyhodnocuje prechod kurzovým lístkom. V závislosti od názvu regulárneho výrazu rozlišujeme 2 typy.

Výrazy typu RE_UPDATE

Regulárny výraz s názvom začínajúcim na „RE_UPDATE_“. Tento regulárny výraz slúži na zber dát ktoré slúžia na neskoršie doplnenie k ostatným nazberaným dátam z dôvodu skompletizovania informácií o jednotlivých zápasoch.

Výrazy typu RE_CREATE

Regulárny výraz s názvom začínajúcim na „RE_CREATE_“. Po úspešnej aplikácii regulárneho výrazu tohto typu sa vytvára konečný objekt s informáciami o zápase ktorý je ďalej doplnený o dáta nazbierané RE_UPDATE_ výrazmi pri doterajšom prechode pravidlami „RULE_“.

Každý regulárny výraz je zadefinovaný na dvoch riadkoch.

Na začiatku prvého riadku sa nachádza názov výrazu ktorý ho identifikuje. Za ním nasleduje kostra regulárneho výrazu.

Kostrou regulárneho výrazu nazývame reťazec, obsahujúci pre človeka ľahko čitateľné časti (názvy čiastkových výrazov). Čiastkové výrazy sú definované v hlavnom konfiguračnom súbore.

Konečný regulárny výraz sa z kostry získa nahradením názvov čiastkových výrazov ich obsahom podľa hlavného konfiguračného súboru. Samozrejme, kostra regulárneho výrazu môže byť samotný regulárny výraz a nemusí obsahovať žiaden čiastkový výraz. V tomto prípade je kostra zhodná s konečným výrazom.

V regulárnom výraze je nutné miesta ktoré zodpovedajú očakávaným dátam uzavrieť do zátvoriek „(“ a „)“.

Na druhom riadku sa nachádza popis dát. Ako popis dát získavaných aktuálnym regulárnym výrazom slúži postupnosť dvojíc (identifikátor_dát poradie) kde poradie znamená poradie odchyťávajúcich zátvoriek v rámci regulárneho výrazu. Tieto dvojice sú oddelené medzerami.

Keďže u textových kurzových lístkov sa často stáva že riadok obsahujúci v podstate rovnaké informácie má rôzny tvar, je umožnené „preťažovanie“ regulárnych výrazov.

„Preťažovaním“ regulárnych výrazov nazývame možnosť zdefinovať viacero regulárnych výrazov s rovnakým názvom. Pri vyhodnocovaní sa potom aplikujú jednotlivé varianty tohto „preťaženého“ výrazu dovtedy kým nejaký nevyhovie alebo kým sa neskúsia všetky varianty.

4.4.2. Hlavný konfiguračný súbor

Hlavný konfiguračný súbor obsahuje definície čiastkových regulárnych výrazov, povolených športov a zakázaných podreťazcov v názvoch líg. Tento konfiguračný súbor sa využíva pri získavaní dát z kurzových lístkov všetkých stávkových kancelárií.

Čiastkové regulárne výrazy sú vlastne regulárne výrazy s prideleným názvom tvaru RE_PART_NAZOV_VYRAZU. V hlavnom konfiguračnom súbore sú zapísane vo forme RE_PART_NAZOV_VYRAZU obsah_vyrazu. Využívajú sa pri vytváraní konečných regulárnych výrazov používaných na parsovanie kurzového lístka. Toto vytváranie prebieha tak, že sa v kostre výrazu z konfiguračného súboru stávkovej kancelárie nájdu všetky podreťazce tvaru %NAZOV_VYRAZU% kde NAZOV_VYRAZU môže obsahovať znaky a-z, A-Z, 0-9 a „_“. Tieto podreťazce následne zamení za obsah_vyrazu. Každý názov výrazu musí začínať reťazcom „RE_PART_“. Každý čiastkový regulárny výraz musí byť na samostatnom riadku.

V definíciách povolených športov máme vymenované športy na ktoré je možné v našej aplikácii tipovať. Zápasy z iných športov sa do databázy nevkladajú. Tieto definície sa využívajú pri pri záverečnom filtrovaní dát načítaných z kurzového lístka. Zápasy zo športov ktoré neboli vymenované sa neukladajú. Definícia povoleného športu je tvaru „ALLOWED_SPORT názov športu“ . Každá definícia musí byť na samostatnom riadku.

Definovaním zakázaných podreťazcov pre názvy líg dosiahneme to, že po načítaní dát z kurzového lístka a odstránení zápasov z nepodporovaných športov ešte v rámci povolených športov môžeme odfiltrovať celé skupiny zápasov na základe

podreťazcov obsiahnutých v názve „ligy“. Ako názov ligy označujeme celý názov ktorým je liga označená u stávkovej kancelárie.

Napríklad „Italie / 2.liga / 41.kolo“ alebo „EURO 2008 / Lepsi ve skupine / Skupina_A“. Ak nechceme umožniť tipovanie na udalosti typu „Lepsi ve skupine“ tak stačí aby sme pridali do konfiguračného súboru riadok
NOT_ALLOWED_LEAGUE Lepsi ve skupine

Tým dosiahneme odstránenie všetkých zápasov načítaných z kurzového lístka, ktoré obsahujú tento podreťazec v rámci názvu ligy. Definícia nepovoleného reťazca je tvaru „NOT_ALLOWED_LEAGUE nepovoleny_retazec“. Každá definícia musí byť na samostatnom riadku.

4.5. Implementácia získavania údajov z kurzových lístkov

Parser je implementovaný formou samostatného, samostatne vyvinutého balíčka. Tento spôsob implementácie umožnil jednoduchšie testovanie na kurzových lístkoch pred tým než sa začal vývoj webovej časti aplikácie.

4.5.1. Trieda MyParser

Táto trieda má na starosti celkovú réžiu získavania údajov.

Najprv získam údaje z konfiguračných súborov. Potom načítam kurzový lístok do zoznamu riadkov. Z tohto celkového zoznamu si vyfiltrujem do ďalšieho zoznamu iba riadky ktoré zodpovedajú (matchujú) nejakému regulárnemu výrazu definovanému v konfiguračnom súbore. Tento zoznam matchujúcich riadkov ďalej predpracujem a tým odstránim neúplne načítané ligy.

Mohlo sa totiž stať že pri použití rôznych regulárnych výrazov kvôli rôznym zápisom jednotlivých športov sa mi namatchujú aj zápasy nejakej nechcenej ligy bez jej názvu, prípadne iba názov ligy bez jej zápasov. Tieto prípady nekompletne načítaných nechcených líg je potrebné odstrániť. To je dosiahnuté pomocou pravidiel zapísaných v konfiguračnom súbore stávkovej kancelárie. Vychádzam z toho, že pred každým matchujúcim regulárnym výrazom typu RE_CREATE musia byť zmatchované všetky deklarované RE_UPDATE regexy (zmatchovaných môže byť pred aktuálnym záznamom aj viacej RE_CREATE regexov, ale to nemá vplyv na vyhodnocovanie – očakáva sa že u líg bude viacej ako len jeden zápas). Postupne prechádzam zoznam matchujúcich riadkov a odstraňujem riadky narúšajúce nejakým spôsobom očakávané poradie. Takéto riadky sú:

- Ak oba matchujú RE_UPDATE regex s rovnakým názvom tak prvý z nich odstránim.
- Ak prvý matchuje a nie je typu RE_CREATE a súčasne druhý nematchuje tak prvý odstránim.
- Ak riadok matchuje a súčasne matchujúci regex nie je typu RE_CREATE a súčasne je to posledný riadok tak tento riadok tiež odstránim.
- Ak prvý riadok je prázdny reťazec a súčasne aj druhý riadok je prázdny reťazec tak prvý z nich odstránim.

Takto sa zbavím prebytočných prázdnych riadkov v zozname. Pre oddelenie kompletných líg mi stačí jeden prázdny riadok.

Z takto predspracovaného kurzového lístka potom zostavím zoznam objektov, kde každý jeden objekt reprezentuje jednu ligu. Liga je reprezentovaná zoznamom riadkov (regulárnych výrazov) do tejto ligy spadajúcich.

Táto celá činnosť sa vykoná v rámci konštrukcie objektu.

Potom volaním metódy *parseLeagues()* získam zoznam objektov typu *EventInfo* reprezentujúcich získané informácie.

4.5.2. Trieda *MyMatcher*

Vlastné získavanie dát z textových reťazcov má na starosti trieda *MyMatcher* a jeho metóda *parse*. Táto metóda sa zavolá na každú ligu v zozname a podľa pravidiel zapísaných v konfiguračnom súbore stávkovej kancelárie riadi prechod a načítanie dát zo zoznamu reťazcov reprezentujúcich danú ligu. Pri prvom zavolaní sa ako názov spracúvaného pravidla musí zadať „RULE_START“ a pozícia reťazca v lige 0. Návratovou hodnotou metódy *parse* je zoznam objektov reprezentujúcich jednotlivé načítané športové udalosti (*class EventInfo*). Vytváranie týchto objektov sa deje pri spracúvaní regulárnych výrazov typu RE_CREATE.

4.5.3. Získavanie údajov

Podľa názvu pravidla zadaného ako parameter potom načítam telo tohto pravidla (1 až 2 kroky), a vyhodnocujem ho pre jednotlivé kroky.

Ak sa telo pravidla skladá iba z jedného kroku, tak sa postupuje nasledujúcim spôsobom. V prípade že krok je názov pravidla tak sa znovu (rekurzívne) volá metóda *parse* s týmto pravidlom a aktuálnou pozíciou v kurzovom lístku. V prípade že krok je názov regexu sa overí či tento regex je typu RE_CREATE. Regex musí byť tohto typu kvôli tomu že ako jediný v pravidle ukončuje rekurzívne získavanie dát a to sa musí ukončiť vytvorením objektu (*EventInfo*).

Ak telo pravidla tvoria dva kroky, tak je postup nasledovný.

Ak je prvým krokom názov pravidla, tak sa zavolá metóda *parse* s týmto názvom pravidla a aktuálnou pozíciou ako parametrom. Toto volanie metódy *parse* nám vráti zoznam objektov reprezentujúcich športové udalosti (*EventInfo*) a pozíciu v kurzovom lístku kde sa prechod zastavil. Druhým krokom potom musí byť tiež názov pravidla. Obdobným volaním metódy *parse* získame ďalší zoznam objektov vyhovujúcich druhému pravidlu. Tieto dva zoznamy nakoniec zreťazíme a výsledok zreťazenia vrátime ako návratovú hodnotu metódy *parse*.

Ak nám pri tvorbe pravidiel zo štruktúry kurzového lístka nejako vyplýva že potrebujeme telo pravidla v tvare „nazov_pravidla nazov_regexu“, môžeme to jednoducho vyriešiť vytvorením ďalšieho pravidla ktorého telo bude iba tento nazov_regexu. Názov tohto pravidla potom použijeme v tele namiesto „nazov_regexu“.

V prípade keď je prvým krokom je regulárny výraz nastávajú dve možnosti. Avšak u oboch možností sa znovu vyžaduje aby druhým krokom bolo pravidlo.

Prvá možnosť znamená že regulárny výraz je typu RE_CREATE. Ak riadok na aktuálnej pozícii v kurzovom lístku vyhovuje tomuto regulárnemu výrazu, načítajú sa z neho dáta, vytvorí sa objekt (*EventInfo*) na základe práve načítaných dát a pokračuje sa druhým krokom, teda spracovaním pravidla (rovnako ako bolo opísané vyššie).

Druhou možnosťou je že regulárny výraz je typu RE_UPDATE. Tu sa postupuje tak, že o dáta načítané aplikáciou tohto regulárneho výrazu updatujeme všetky objekty v zozname ktorý nám vráti spracovanie pravidla (*parse*) v druhom kroku. Tento zoznam updatovaných objektov je potom návratovou hodnotou.

4.5.4. Príklad získavania dát

4.5.4.1 Časť kurzového lístka

Fotbal / Anglie / Dohravky / 11.tyden							
	12.3.20:45	1	0	2	10	12	02
938	Aston Villa-Middlesbrough	1.62	3.5	4.8	1.1	1.21	2.02
940	Portsmouth-Birmingham	1.72	3.4	4.25	1.14	1.22	1.88
	12.3.21:00						
939	Chelsea-Derby	1.1	8	13	-	1.01	4.95
Fotbal / Ceska rep. / 1.liga / 20.kolo							
	10.3.17:15	1	0	2	10	12	02
107 +	Sparta-Teplice	1.42	3.8	7	1.03	1.18	2.46

4.5.4.2. Konfiguračný súbor

Pozn.: Zobrazená je iba časť konfiguračného súboru nutná pre porozumenie príkladu (pravidlá riadiace spracovanie kurzového lístka). Nie sú zobrazené regulárne výrazy a rovnako nie sú zobrazené ani reťazce ALLOWED_SPORT a NOT_ALLOWED_LEAGUE. Taktiež nie je vyobrazený hlavný konfiguračný súbor s deklaráciami čiastkových regulárnych výrazov.

```
RULE_START { RULE_LEAGUES }
```

```
RULE_LEAGUES { RULE_LEAGUE RULE_LEAGUES }
```

```
RULE_LEAGUE { RE_UPDATE_LIGA RULE_TIME_BLOCKS }
```

```
RULE_TIME_BLOCKS { RULE_TIME_BLOCK RULE_TIME_BLOCKS }
```

```
RULE_TIME_BLOCK { RE_UPDATE_CAS RULE_EVENTS }
```

```
RULE_EVENTS { RE_CREATE_EVENT RULE_EVENTS }
```

4.5.4.3. Postup získavania dát

1. Zavoláme metodu *parse* s parametrami – zoznam riadkov kurzového lístka, „RULE_START“ (pré spracúvané pravidlo), pozícia – prvý riadok
2. Postupne sa spracovaním pravidiel RULE_LEAGUES a RULE_LEAGUE dostaneme k aplikácii regulárneho výrazu RE_UPDATE_LIGA. Aplikácia na prvý riadok prebehne úspešne, načítame názov športu a ligy a posunieme sa v kurzovom lístku na ďalší (druhý) riadok.
3. Sme na druhom riadku kurzového lístka a pokračujeme spracovaním pravidla RULE_TIME_BLOCKS. Cez pravidlo RULE_TIME_BLOCK sa dostaneme k aplikácii regulárneho výrazu RE_UPDATE_CAS na druhý riadok kurzového lístka. Aplikácia je úspešná, máme načítaný dátum a čas zápasov, posunieme sa v kurzovom lístku na tretí riadok a pokračujeme spracovaním pravidla RULE_EVENTS.
4. Prvým krokom pravidla RULE_EVENTS je aplikácia regulárneho výrazu RE_CREATE_EVENT na aktuálny(tretí) riadok kurzového lístka. Aplikácia je úspešná, my načítame názvy teamov, id zápasu a kurzy na tento zápas a posunieme sa v kurzovom lístku na štvrtý riadok. Vytvorí sa objekt typu *EventInfo* a spracúva sa znova RULE_EVENTS.
5. Znovu úspešne aplikujeme regulárny výraz RE_CREATE_EVENT, vytvoríme objekt typu *EventInfo* obsahujúci načítane údaje a posunieme sa v kurzovom lístku na piaty riadok. Pri nasledujúcom pokuse o ďalšie spracovanie pravidla RULE_EVENTS nad piatym riadkom neuspějeme, tak sa vynoríme z rekurzie až na úroveň pravidla RULE_TIME_BLOCKS. Avšak pred vynorením z úrovne pravidla RULE_TIME_BLOCK updatujeme všetky objekty typu *EventInfo* navrátené rekurziou o časové údaje načítané predtým pri aplikácii regexu RE_UPDATE_CAS na druhom riadku. Na úrovni RULE_TIME_BLOCKS sa uchovávajú objekty typu *EventInfo* vrátené z prvého kroku (aplikácii pravidla RULE_TIME_BLOCK) a pokračuje sa druhým krokom, čo je znova spracovanie pravidla RULE_TIME_BLOCKS. Aktuálne sme na piatom riadku kurzového lístka.
6. Obdobne potom spracovaním pravidla RULE_TIME_BLOCKS načítame piaty a šiesty riadok až skončíme na siedmom.
7. Siedmy (prázdny) riadok nepochybuje so žiadnym regulárnym výrazom a prinúti nás vynoriť sa z rekurzie na úroveň nad všetkými update regulárnymi výrazmi. Pri tomto vynorení sa skompletizovali všetky objekty typu *EventInfo* ktoré boli vytvorené pri načítaní prvej ligy. Keď sme už vynorení nad všetky update regexy tak tento prázdny riadok preskočíme, posunieme sa na ďalší (ôsmy) riadok a pokračujeme v spracovaní pravidla na ktorom sme ostali pri vynáraní z rekurzie (prípadne pokračujeme vo vynáraní).
8. V našom príklade sme pri vynáraní zastali na druhom kroku pravidla RULE_LEAGUES, čo je znova pravidlo RULE_LEAGUES. Takže pokračujeme obdobne ako v prípade načítania prvej ligy.

9. Po dočítaní kurzového lístka do konca sa už iba úplne vynoríme z rekurzie a máme vo forme objektov typu *EventInfo* prečítaný celý kurzový lístok.

4.6. Zhodnosť zápasov

Automatické vyhľadávanie zodpovedajúcich zápasov

Pri pridávaní kurzov zo ostatných používaných stávkových kancelárií ku zápasom ktoré už máme načítané z hlavnej stávkovej kancelárie sa ocitáme pred problémom nájdenia zhodných zápasov. Pod zhodnosťou zápasov rozumieme to, že určitý zápas môžu mať rôzne stávkové kancelárie rozlične pomenovaný. Typicky sa líšia názvami teamov, no sú aj prípady keď je rozdielny čas začiatku zápasu.

Vyhľadávanie zápasov z databázy ktoré zodpovedajú zápasu s pridávaným kurzom má 2 stupne presnosti. Prvým stupňom je:

Nájdenie úplne zhodného zápasu:

Nájde v DB zápas so zhodným športom, dátumom zhodným na minútu a so zhodnými domácimi teamami a teamami hostí. V prípade že sa takýto zápas nájde, pridáme mu pridávaný kurz a pokračujeme pridávaním ďalšieho kurzu. Zhodnosť teamov znamená že oba názvy sú aliasy rovnakého teamu. Teda že ALIAS_TEAM_ID pre názov teamu domácich je zhodné s ALIAS_TEAM_ID názvu teamu hostí.

Ak sa takýto zápas nenájde, tak nasleduje druhý stupeň:

Vyhľadanie čiastočne zhodných zápasov

Za čiastočne zhodný je považovaný zápas so zhodným športom, dátumom zhodným na minútu a zhodným teamom domácich ALEBO teamom hostí. U teamu u ktorého nebola nájdená zhodnosť sa najprv pridá názov teamu zo spracúvaného zápasu ako alias teamu ktorý máme v databáze. Týmto dosiahneme úplnú zhodnosť. Potom sa nájdenému zápasu pridajú kurzy spracúvaného zápasu a pokračuje sa spracúvaním ďalšieho zápasu.

Ak sa ani takýto zápas nenájde, spracúvaná udalosť sa uchová ako udalosť s nepriradeným kurzom a o priradení k zápasu v databáze rozhodne administrátor.

4.7. Automatizované získavanie údajov

Automatizované získavanie športových udalostí, kurzov a výsledkov je vyriešené pomocou volania adresy `adminGetOnlineKurzy.jsp`. Pri načítaní tejto stránky sa najprv spustí proces získavania výsledkov. Potom sa spustí proces pri ktorom sa najprv získajú športové udalosti a kurzy z hlavnej stávkovej kancelária a potom sa doplnia kurzy ostatných stávkových kancelárií. Tento spôsob získavania dát je implementovaný iba pre stávkové kancelárie ktoré majú jednoducho prístupný textový súbor s kurzovým lístkom (napr. `www.adresa.cz/nabidka.txt`). V prípade získavania výsledkov musí byť u hlavnej stávkovej kancelárie jednoducho prístupný textový súbor s výsledkami.

5. Záver

Cieľom práce na zadanú tému Virtuální sázková kancelář bolo vytvoriť program Tipserver, ktorý by umožňoval užívateľom tipovať výsledky športových udalostí za virtuálne body bez potreby finančných investícií.

Ďalej je užívateľom umožnené zobrazit' si historické štatistiky úspešnosti u seba, alebo aj u iných užívateľov. Každý mesiac sú vyhodnocované poradia úspešnosti vo viacerých kategóriách, čo vnáša do aplikácie pre časť užívateľov prít'azlivý súťažný prvok.

Jednou z dôležitých vlastností programu je jednoduché pridávanie stávkových kancelárií a ich konfigurácia pre získavanie dát.

Vďaka jednotnému systému pre získavanie údajov z kurzových lístkov je potrebné iba napísať konfiguračný súbor ktorým sa má bude získavanie dát riadiť.

Program poskytuje možnosť automatizovaného získavania dát z textových súborov umiestnených na internete, a tým znižuje množstvo potrebnej administrátorskej práce. Za pomoci správcu aplikácie sa program dokáže naučiť rozpoznávať väčšinu zhodných športových udalostí získavaných z iných stávkových kancelárií.

Pokračovaním vývoja tohto programu by bolo možné rozšíriť ho o radu ďalších funkcií:

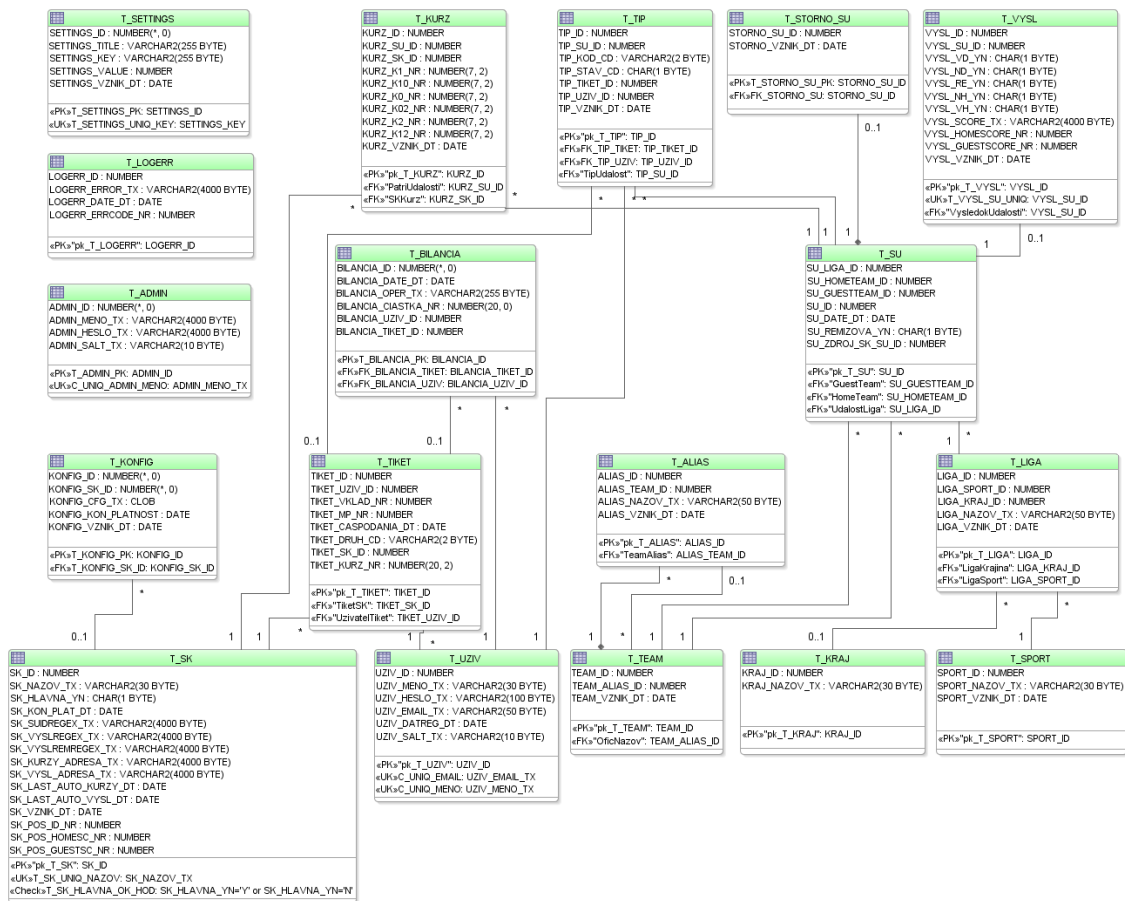
- Zmena hlavnej stávkovej kancelárie
- Podpora parsera pre prácu s viacriadkovými reťazcami (čo by umožnilo získavať dáta z HTML súborov)
- Aliasy pre názvy športov a líg
- Podrobné užívateľské štatistiky

Literatúra

- [1] Bill Dudley, Jonathan Lehr, Bill Willis, LeRoy Mattingly: Mastering JavaServer Faces, Wiley Publishing Inc., Město, 2004
- [2] Luboslav Lacko: Oracle – Správa, programování a použití databázového systému, Computer Press, Brno, 2007
- [3] Brett Spell: Java, programujeme profesionálně, Computer Press, 2002

Prílohy

Štruktúra databázy použitej v programe



Obr. 3.1