

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Šorn

**Sistem za prikaz rangiranja športnih  
klubov**

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM  
ŠTUDIJU

MENTOR: doc. dr. Erik Štrumbelj

Ljubljana, 2016



Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Sistem za prikaz rangiranja športnih klubov

Tematika naloge:

Študent naj izdelava programsko opremo za zbiranje in hranjenje rezultatov športnih tekem, uporabo teh podatkov v obstoječem algoritmu za rangiranje ter prikaz izračunanih rangov na spletni strani. Za razvoj naj izbere najbolj primerno programsko opremo, pri čemer so zaželeno brezplačne oziroma odprtokodne rešitve.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Jure Šorn, vpisna številka 63010279, avtor zaključnega dela z naslovom:

*Sistem za prikaz rangiranja športnih klubov*

### IZJAVLJAM

1. da sem pisno zaključno delo študija izdelal samostojno pod mentorstvom doc. dr. Erika Štrumbelja;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

V Ljubljani, 30. avgust 2016

Podpis študenta:





*Zahvaljujem se doc. dr. Eriku Štrumbelju za vso pomoč in izredno hitro odzivnost tudi v času počitnic.*

*Rad bi se zahvalil tudi družini za vso podporo pri zaključevanju študija.*

*Hvala tudi vsem ostalim za spodbudo.*



Vsem ki jim ni težko,  
čeprav bi jim bilo lahko.



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Sorodna dela . . . . .	2
1.2	Kaj smo želeli izboljšati . . . . .	4
1.3	Kratek povzetek preostanka vsebine . . . . .	5
<b>2</b>	<b>Zahteve in zaledni del sistema</b>	<b>7</b>
2.1	Zaledni del sistema . . . . .	8
2.2	Algoritem za rangiranje . . . . .	18
<b>3</b>	<b>Čelni del sistema</b>	<b>21</b>
3.1	Zemljevidi . . . . .	21
3.2	Predstavitve strani . . . . .	24
<b>4</b>	<b>Zaključek</b>	<b>29</b>
	<b>Literatura</b>	<b>31</b>



# Povzetek

**Naslov:** Sistem za prikaz rangiranja športnih klubov

**Povzetek:**

V diplomski nalogi je predstavljen sistem za rangiranje športnih klubov, ki temelji na kvotah stavnic. Sistem sestavljajo štirje deli, in sicer podatkovna baza, skripta za pobiranje stavnih kvot in rezultatov iz spleta, skripta za izračunavanje rangov ter spletna stran za prikaz rangov. Ker smo za izračunavanje rangov uporabili obstoječi algoritem, smo se pri razvoju osredotočili na robustnost pobiranja podatkov iz spleta in na preglednost spletne strani. Za razliko od obstoječih spletnih strani, ki prikazujejo range ekip, smo uporabili zemljevide in klube predstavili skupaj z njihovimi geografskimi lokacijami, z barvami pa tudi moči posameznih držav. Za razvoj smo izbrali odprtokodne tehnologije, ki so nam omogočile hiter razvoj razširljivega sistema.

**Ključne besede:** spletne strani, podatkovna baza, strganje, SQLite, Python





# Abstract

**Title:** A system for displaying sports club rankings

**Abstract:**

In this thesis we present a system for displaying sports club rankings that are based on betting odds. The system consists of four main parts: a database, a script for scraping betting odds and game results from websites, a script for calculating the ranks and a website that displays them. We used an existing algorithm for rank calculation and focused our efforts on the development of a robust scraping algorithm and a transparent website. The main characteristic that distinguishes this project from other similar ones, is our use of maps, which enable us to display geolocations of the clubs and also, with the use of colors, the strengths of individual countries. For development we chose open source technologies that enabled us quick development of an extensible system.

**Keywords:** web pages, database, scraping, SQLite, Python.



# Poglavje 1

## Uvod

V diplomski nalogi smo izdelali sistem za prikazovanje rangiranja športnih klubov. Sistem temelji na obstoječem algoritmu za računanje rangov ekip na podlagi rezultatov odigranih tekem in kvot, ki so jih določile stavnice, in mora poskrbeti za zajem in hranjenje teh podatkov ter prikaz rangiranj, ki jih izračuna algoritem. Zato je sistem sestavljen iz dveh delov: strgalnika (angl. scraper), ki zajema podatke o tekmah s spleta, in spletne strani, ki služi prikazu izračunanih rangov ekip.

Sistem je v osnovi namenjen širši športni javnosti, saj športni navdušenci radi spremljajo svoje klube in jih primerjajo z drugimi klubi. Sistem pa je potencialno zanimiv tudi za rangiranje klubov, ki je potrebno za postavljanje nosilcev v tekmovanjih ter kot pomožno orodje za menedžerje klubov in skavte, ki iščejo manj znane in poceni igralce v nižjih ligah. Razporejanje ekip po rangih omogoča lažje prepoznavanje ekip z velikim potencialom kot samo analiza rezultatov in položaja kluba znotraj lige. Takšne ekipe so ponavadi sestavljene iz mladih in talentiranih igralcev, ki so zanimivi za klube v višjih ligah.

ENGLISH ▾

Football Club World Ranking®

Home Ranking Results News FAQ Club Directory

WEEK 30 RANKING PER COUNTRY CLUB FINDER

07/25/2016 ▾ All countries ▾  GO >

f t g+

1	2	3	4	5
<b>Barcelona</b> Barcelona 18,174 pts	<b>Real Madrid</b> Madrid 15,076 pts	<b>Bayern München</b> München 14,100 pts	<b>Atlético Madrid</b> Madrid 13,764 pts	<b>Borussia Dortmund</b> Dortmund 12,299 pts
6	7	8	9	10
<b>Sevilla</b> Sevilla 11,603 pts	<b>Athletic Club</b> Bilbao 10,818 pts	<b>Villarreal</b> Villarreal 10,774 pts	<b>PSG</b> Paris 10,559 pts	<b>Santos</b> Santos, São Paulo 10,173 pts

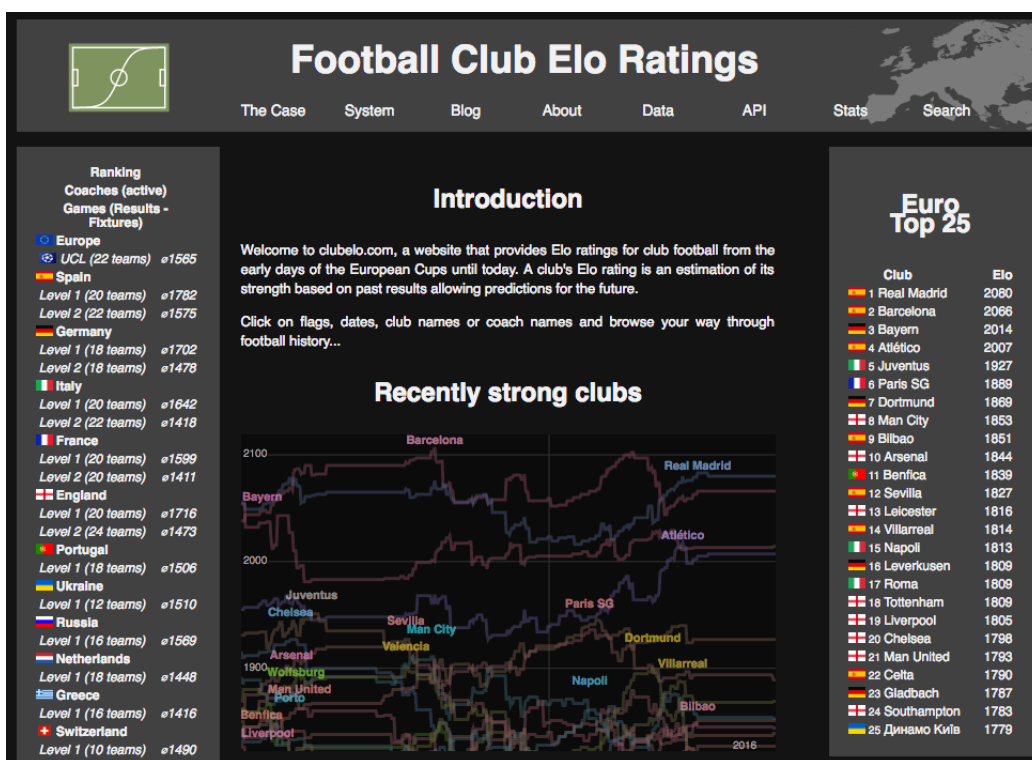
FULL CLUB WORLD RANKING +

Slika 1.1: Domača stran spletne strani [www.clubworldranking.com](http://www.clubworldranking.com).

## 1.1 Sorodna dela

Nekaj podobnih sistemov že obstaja, vendar jih ima večina nepregledne ter zastarele uporabniške vmesnike. Najbolj podobni našemu projektu sta strani Club world ranking (slika 1.1) in Club ELO (slika 1.2), ki sta namenjeni rangiranju nogometnih klubov.

Spletna stran Club world ranking je namenjena rangiranju nogometnih klubov, ki nastopajo v prvih državnih ligah. Rangirani so številski (maksimalni je trenutno 18174) in se računajo z uporabo enostavnih formul [2]. Stran



Slika 1.2: Domača stran spletne strani www.clubelo.com.

ima pregledno in estetsko vizualno podobo. Domača stran prikazuje lestvico trenutno najboljših klubov, prikaz pa lahko osredotočimo tudi na posamezne države. Na klubski strani so osnovni podatki o klubu, trenuten ter najvišji dosežen rang in seznam lovorik. Pod navedenimi podatki so še trije zavihki, ki prikazujejo zadnje rezultate, lestvico državne lige, v kateri nastopa klub, in graf, ki prikazuje gibanje klubskega ranga v zadnjem letu.

Spletna stran Club ELO rangira nogometne klube z uporabo sistema ELO, poimenovanem po njegovem izumitelju, madžarskem fiziku Arpadu Elu [4]. Sistem je bil na začetku namenjen rangiranju šahistov, vendar se uporablja tudi za številna druga tekmovanja in športe. Na domači strani se nahaja lestvica petindvajsetih najboljših evropskih klubov, graf, ki prikazuje gibanje ratingov najboljših klubov v zadnjem letu, ter tabela tekem, ki so bile odigrane v zadnjih dveh dneh. Vsaka vrstica tabele vsebuje poleg rezultata

tekme tudi verjetnosti posameznih izidov, skupaj s številom točk ELO, ki jih je prejela zmagovalna in izgubila poražena ekipa. Prisoten je tudi graf, ki prikazuje ratinge najboljših klubov od leta 1960 naprej, in lestvica trenerjev, v kateri so le-ti razporejeni glede na lastne ratinge. Na vseh straneh je na levem robu prisotna lestvica držav, ki prikazuje ratinge le-teh. Prikaz podatkov na podstrani države je podoben kot na domači strani, le da prikazuje samo podatke o klubih izbrane države. Na klubske strani je prikazan graf ratingov kluba za zadnje štiri leta, pod katerim se nahaja tabela na zadnje odigranih tekem, v kateri so navedeni predvideni rezultati in spremembe ratinga ELO. Zabeležena je tudi prognoza, s koliko točkami in na katerem mestu bo klub končal klubske tekmovanje, manjkajo pa osnovni podatki o klubu in logotip kluba.

## 1.2 Kaj smo želeli izboljšati

Glavna motivacija pri razvoju sistema je bilo dejstvo, da nobeden od sorodnih sistemov ne uporablja algoritma, ki bi range izračunaval iz kvot stavnic [2, 3]. Predvidevamo, da uporaba le-tega omogoča boljšo oceno moči klubov, predvsem pri primerjavi klubov, ki ne igrajo v istih tekmovanjih. Poleg drugačnega algoritma za rangiranje, naša stran omogoča tudi rangiranje ekip in tekmovalcev iz poljubnih športov in ne samo nogometa.

Osredotočili smo se predvsem na čelni del sistema. Želeli smo ustvariti spletno stran s preglednejšim in bolj intuitivnim vmesnikom, kot ga ponujajo že obstoječe strani. To smo poskušali doseči tako, da smo range prikazali z uporabo zemljevida, na katerem so označene geografske lokacije klubov. Poleg lokacij klubov smo, z uporabo različnih svetlosti barv, na zemljevidu označili tudi moči posameznih držav. Za hitrejše prepoznavanje klubov smo tabelam in oznakam lokacij, dodali logotipe klubov in ikone zastav, ki sporočajo, iz katere države izhaja klub.

### **1.3 Kratek povzetek preostanka vsebine**

V 2. poglavju bomo najprej podrobno predstavili zahteve in nato opisali postopek realizacije zalednega dela sistema, predvsem posebnosti in robne primere, s katerimi smo se srečali pri realizaciji. Sledilo bo 3. poglavje, kjer bomo opisali postopek realizacije čelnega dela sistema in predstavili spletne strani. Sledi še zaključek, kjer so izpostavljena glavna dognanja, do katerih smo prišli pri realizaciji projekta, in zaključne misli, ki bi lahko bile v pomoč pri realizaciji podobnih projektov.





## Poglavje 2

# Zahteve in zaledni del sistema

Pri izdelavi sistema smo izhajali iz zahtev:

- Da sta strgalnik in spletna stran razvita v istem programskem okolju.
- Da je programsko okolje brezplačno.
- Da je program združljiv z operacijskima sistemoma Microsoft Windows in Linux.
- Da programsko okolje in podatkovna baza omogočata sprejemljive odzivne čase spletne strani.
- Da želimo kratek čas izgradnje prve različice delujočega sistema, ki ga bo možno kasneje razširiti in popraviti. Zato je zaželen uporaba razširjenega programskega okolja, za katerega obstaja zadostno število knjižnic, gradiv in spletnih strani z vprašanji in odgovori (angl. Q&A). Predvsem je pomemben obstoj zanesljivih in enostavnih knjižnic za razčlenjevanje HTML dokumentov in za izgradnjo spletnih strani.
- Da se za prikaz lokacij klubov uporabijo zemljevidi.
- Da se na smiselnih mestih ob imenih klubov prikaže tudi njihove logotipe.

- Da se kvote odigranih tekem pobirajo iz spletne strani BetExplorer ([www.betexplorer.com](http://www.betexplorer.com)).
- Da spletna stran vsebuje podstrani za prikaz podatkov o športih, državah ter klubih.

Zaradi razvejanih relacij med vhodnimi podatki smo za njihovo hranjenje uporabili podatkovno bazo, in sicer SQLite, ker je "lahka" (angl. lightweight), a vseeno dovolj zmogljiva za naše potrebe. V bazi se za vsak klub hrani odigrane tekme ter kvote. Hrani se tudi logotip kluba ter koordinate stadiona ali dvorane, ki se uporabijo za prikaz lokacije kluba na zemljevidu.

Za vir kvot je bila zahtevana spletna stran BetExplorer, več svobode pa smo imeli pri izboru vira logotipov in lokacij klubov. Na koncu smo izbrali spletno enciklopedijo Wikipedia ([www.wikipedia.org](http://www.wikipedia.org)), ker vsebuje podatke za zadostno število klubov in ker so logotipi in lokacije enostavno dostopni preko klubske strani.

Za programsko okolje smo izbrali Python ker je kompatibilno tako z operacijskima sistemoma Microsoft Windows in Linux, ponuja ogrodja za dinamično izgradnjo spletnih strani s sprejemljivimi odzivnimi časi, omogoča hiter razvoj in ker je brezplačno, ter zato ustreza vsem zadanim zahtevam.

V preostanku poglavja bomo bolj podrobno opisali realizacijo posameznih komponent sistema, nekatere posebne primere, izbrane tehnologije in na kratko predstavili možne alternative.

## 2.1 Zaledni del sistema

Zaledje je sestavljeno iz treh delov, in sicer strgalnika, ki prenese potrebne podatke iz spleta, algoritma za računanje rangov in generatorja dinamičnih spletnih strani.

Strgalnik je sestavljen iz dveh delov: skripte, ki prenese stalne podatke o klubih in jo je potrebno zagnati samo enkrat, ter skripte, ki prenaša kvote in se jo zaganja v ciklih poljubne dolžine. Med stalne podatke spadajo logotipi klubov in njihove lokacije v obliki koordinat. Vsi pobrani podatki

razen logotipov se zapisujejo v podatkovno bazo. Na podlagi teh podatkov nato algoritem za rangiranje izračuna range, ki jih prav tako shrani v bazo. Uporabnik dostopa do teh podatkov preko spletne strani.

Za realizacijo zaledja smo uporabili programski jezik Python, glavne prednosti katerega so kompatibilnost z vsemi glavnimi operacijskimi sistemi in množica obstoječih knjižnic ter ogrodij za izgradnjo dinamičnih spletnih strani in pobiranje podatkov iz spleta.

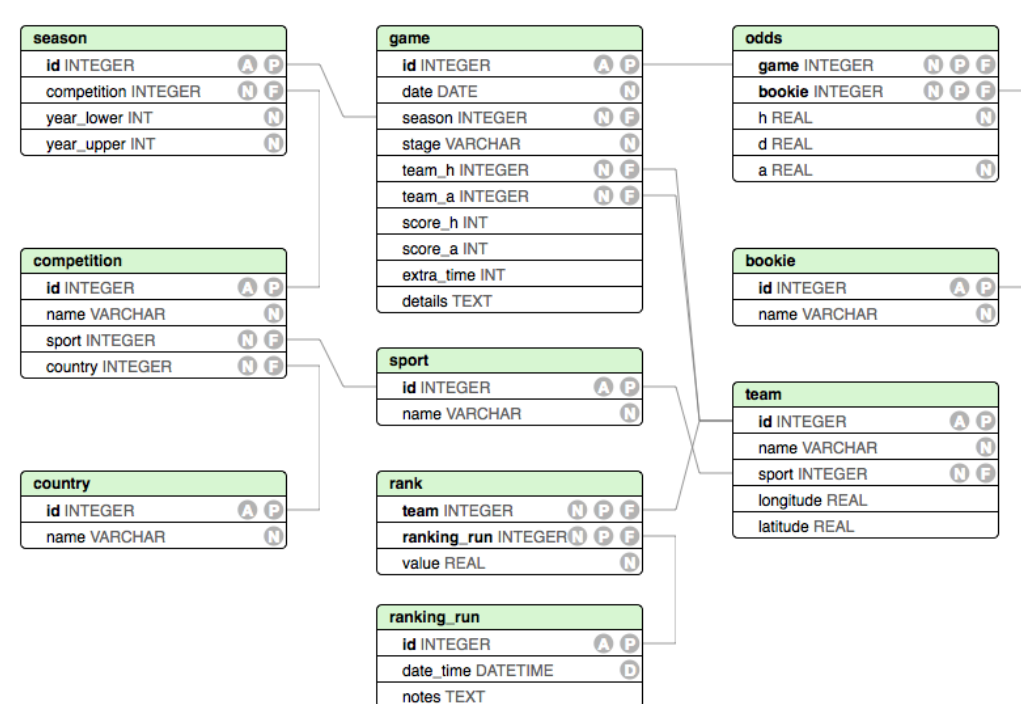
### 2.1.1 Podatkovna baza

Za hranjenje podatkov smo uporabili podatkovno bazo SQLite. Baza SQLite je najbolj razširjena "lahka" podatkovna baza [10], kar pomeni, da jo lahko namestimo in zaženemo brez urejanja konfiguracijske datoteke, hkrati pa hrani vsako podatkovno bazo v svoji datoteki, kar omogoča enostavno migracijo iz enega okolja v drugo. Možna alternativa tej izbiri bi bila podatkovna baza MySQL, ki je trenutno najbolj razširjena prosto dostopna podatkovna baza [10], vendar ni tako enostavna za uporabo.

Vsi deli sistema so povezani preko podatkovne baze. Do nje dostopajo preko knjižnice SQLite, ki je del standardne knjižnice jezika Python, kar pomeni, da je ni potrebno posebej namestiti. Osnovne funkcije, ki jih ta knjižnica ponuja, so funkcija "connect", s katero se povežemo z zeleno bazo SQLite, metoda "execute", s katero pošiljamo SQL ukaze bazi, ter metoda "close", s katero prekinemo povezavo.

#### Struktura baze

Podatkovna baza vsebuje 11 tabel. Tabele hranijo tekme, sezone, tekmovanja, športe, države, ekipe, kvote, stavnice, range ter podatke o posameznih rangiranjih. Večina tabel v prvi vrstici vsebuje celoštevilčni razločevalec, ki služi kot primarni ključ. Podroben prikaz vseh relacij, atributov in podatkovnih tipov podatkovne baze je podan na sliki 2.1.



Slika 2.1: Shema podatkovne baze.

### 2.1.2 Strgalnik

Za pobiranje podatkov iz spletnih strani smo uporabili knjižnico Beautiful soup, ker je najbolj razširjena Python knjižnica za ta namen, kar pomeni tudi, da za njo obstaja veliko priročnikov ter gradiv. V primeru, da strani, iz katerih se pobira podatke, vsebujejo vsebino, ki se dinamično ustvarja na uporabnikovi strani s pomočjo JavaScript kode, je potrebno poleg knjižnice Beautiful soup uporabiti še knjižnico Selenium, s pomočjo katere se ustvarijo elementi HTML. V našem primeru ni bilo te potrebe, ker so bili vsi potrebni podatki na strani BetExplorer že v formatu HTML.

#### Nastavitvena datoteka

Delovanje strgalnika upravljamo s pomočjo nastavitvene datoteke. Iz nje strgalnik prebere, iz katerih držav, za katere športe ter za katero obdobje

The screenshot displays the BetExplorer.com interface. At the top, there are navigation tabs for various sports: HOME, SOCCER, HOCKEY, BASKETBALL, HANDBALL, VOLLEYBALL, and BASEBALL. Below these, a breadcrumb trail shows 'HOME » Soccer » Austria'. A sidebar on the left contains navigation options like RESULTS, NEXT MATCHES, ODDS FILTER, etc. The main content area is divided into sections for different seasons (2016/2017 and 2015/2016) and a 'SELECT COUNTRY' dropdown menu. A match page for 'Admira - Rapid Vienna' is shown, including a score of 1:3, a list of players, and a table of 1X2 odds from various bookmakers.

Slika 2.2: Hierarhija spletne strani www.betexplorer.com.

želi uporabnik pobrati rezultate in kvote tekem iz spletne strani BetExplorer. Poleg navedenih podatkov se v nastavitveni datoteki lahko določi tudi, ali želimo pobrati samo podatke, ki še niso bili preneseni, in, ali želimo prepisati že prenesene vrednosti.

V izseku 2.1 je prikazan primer vrstice iz nastavitvene datoteke, ki določi, da naj se poberejo vsi rezultati in kvote iz obdobja od leta 2011 do 2012 za nogometne tekme odigrane v avstrijskih prvenstvih.

Strgalnik nato pobere zelene podatke iz spletne strani BetExplorer, tako da se pomika po njeni hierarhiji (slika 2.2). Na vrhu te hierarhije so različni športi, na strani, ki prikazuje določen šport, je lista vseh držav, v katerih se igra izbrani šport, na strani države so po letih razporejene sezone vseh lig,

```
1 soccer; year-from: 2011, year-to: 2012, replace-old-entries:  
   false, scrape-only-new-matches: true, countries: Austria
```

Izsek 2.1: Primer vrstice v nastavitveni datoteki.

za katere stran hrani podatke, in na strani sezone je spisek vseh odigranih tekem. Na strani posamezne tekme se nahajajo podatki, ki se nato zapišejo v bazo: končni rezultat, rezultat med polčasom in vrednosti kvot različnih stavnic za vse možne izide (zmaga domače ekipe, neodločen izid ali zmaga gostov).

Za periodično zaganjanje skripte v zelenih intervalih mora uporabnik poskrbeti sam. Na sistemu Linux se to doseže tako, da se v nastavitveno datoteko "crontab" doda vrstico z datotečno potjo strgalnika, poleg katere se določi interval, v katerem se bo zaganjala.

## Beautiful soup

Beautiful soup je paket za programsko okolje Python, ki omogoča razčlenjevanje dokumentov HTML in XML. Iz razčlenjenega dokumenta ustvari podatkovno strukturo drevesa, s pomočjo katere lahko nato dostopamo do zelenih podatkov HTML [5]. Paket je združljiv s programskim okoljem Python 3, in sicer od verzije 2.6 dalje.

V izseku 2.2 je prikazan primer uporabe knjižnice, ki prenese vsebino domače strani spletne enciklopedije Wikipedia in izpiše spletne naslove vseh povezav.

### 2.1.3 Strgalnik logotipov

Poleg skripte za pobiranje kvot smo implementirali še skripto, ki pobira logotipe in koordinate klubov. Teh podatkov algoritem za rangiranje ne potrebuje, tako da so namenjeni izključno večji preglednosti spletnih strani.

Načeloma je težko dobiti logotipe klubov, še posebej pa geolokacije, zato se je Wikipedia izkazala za najboljši vir, saj na enem mestu ponuja oboje.

```
1 from bs4 import BeautifulSoup
2 import urllib2
3
4 webpage = urllib2.urlopen('http://en.wikipedia.org')
5 soup = BeautifulSoup(webpage, 'html.parser')
6 for anchor in soup.find_all('a'):
7     print(anchor.get('href'))
```

Izsek 2.2: Primer uporabe knjižnice Beautiful soup.

Potrebno pa je omeniti, da, ker stran BetExplorer beleži tudi tekme ekip iz amaterskih lig, jih veliko nima svoje podstrani na strani Wikipedia, ali pa na njej ni logotipa in/ali geolokacije. Ker se delež takšnih ekip giblje okrog petdesetih odstotkov, smo implementirali še skripto, ki zgenerira nastavitveno datoteko za ročni vnos manjkajočih koordinat.

Kot vse ostale skripte, je tudi strgalnik logotipov napisan v programskem jeziku Python in prav tako dostopa do podatkovne baze s pomočjo knjižnice SQLite. S strgalnikom kvot si deli datoteko "db.py", v kateri so definirane pomožne funkcije za dostop do podatkovne baze.

### Pregled delovanja

Najprej skripta iz podatkovne baze pobere seznam vseh obstoječih klubov in za vsakega preveri, ali ima že shranjen logotip in/ali koordinate. V primeru, da kateri od podatkov manjka, ga poskuša prenesti s spleta. Za iskanje naslova klubske podstrani na strani Wikipedia uporabi spletni iskalnik Google, ker je pri iskanju strani veliko bolj uspešen kot privzeti iskalnik. Iskalni niz se sestavi iz besede "wiki", imena kluba in športa. Nato se z uporabo knjižnice Urllib prenese stran z zadetki, od koder se s pomočjo knjižnice Beautiful soup izlušči spletni naslov prvega zadetka. Če je bilo iskanje uspešno, vsebuje prvi zadetek naslov podstrani spletne enciklopedije Wikipedia, na kateri se nahajajo podatki o klubu. Po prenosu te strani se izvajanje programa loči na dve veji, in sicer funkcijo za pobiranje logotipa in funkcijo za pobiranje koordinat.

## Funkcija za pobiranje logotipa



Slika 2.3: Okvir z ključnimi podatki kluba.

Pobiranje podatkov iz spletne strani Wikipedia znatno olajša dejstvo, da večina klubskih strani uporablja isto predlogo (slika 2.3). Predloga vsebuje okvir s ključnimi podatki kluba na desnem robu strani. V tem okvirju se nahaja tabela razreda "infobox vcard", ki med drugim vsebuje naslov slike s klubskim logotipom in ime štabiona ali arene, ki je hkrati tudi povezava na njegovo podstran na strani Wikipedia. Če slika ne obstaja, jo stran ne vsebuje ali pa, je bil prvi zadetek spletnega iskalnika Google napačen. Če slika obstaja, se njen naslov pošlje funkciji "urlretrieve" knjižnice Urllib. Ta funkcija prenese vsebino, ki se nahaja na podanem naslovu, in jo shrani v



lokalno datoteko, datotečna pot katere se poda preko drugega argumenta funkcije. Ime datoteke se sestavi iz športa in imena kluba.

### **Funkcija za pobiranje koordinat**

Koordinate se za razliko od logotipov zapišejo v podatkovno bazo, in sicer v tabelo Team. Nahajajo se na strani klubskega štadiona ali arene, v desnem zgornjem kotu, znotraj elementa "span", ki pripada razredu "geo-dms". Preden se koordinate zapiše v podatkovno bazo, jih je potrebno še pretvoriti, in sicer iz seksagezimalnega sistema v decimalni.

### **Skripta za prikaz manjkajočih logotipov in koordinat**

Poleg skripte za prenos logotipov in koordinat obstaja še skripta, s katero preverimo katerih logotipov in koordinat nam ni uspelo prenesti. Ta skripta ustvari datoteko "missing-data.txt", v kateri so navedeni iskalni nizi, s katerimi je skripta poskušala najti naslov spletne strani kluba, kar nam omogoča, da enostavno kopiramo iskalni niz in ga prilepimo v iskalno vrstico brskalnika. Datoteka poleg iskalnega niza vsebuje še ključne za vnos koordinat, ki so oblike "<šport>-<ime kluba>-[latituda|longituda]:", kar nam omogoča, da enostavno prilepimo manjkajočo koordinato za dvopičjem. Iz te datoteke nato skripta "read\_coordinates.py" prebere manjkajoče koordinate in jih vnese v podatkovno bazo.

### **2.1.4 Generator spletnih strani**

Generator spletnih strani je realiziran s pomočjo mikro programskega ogrodja Bottle. Ogrodje Bottle omogoča izgradnjo dinamičnih spletnih strani z minimalno količino splošne (angl. boilerplate) kode. Za programski jezik Python obstaja veliko število ogrodij za ustvarjanje spletnih strani, vendar jih večina vsebuje preobširen nabor konstruktov, zaradi česar njihova uporaba ni primerna za enostavnejše spletne strani, ker upočasnjuje razvoj in privede do preveč zapletene kode. Ogrodje Bottle smo izbrali, ker je eno najmanj obsežnih in

je bilo zato najbolj primerno za našo dokaj enostavno spletno stran, hkrati pa je dovolj razširjeno, kar pomeni da zanj obstaja dovolj podpore in dokumentacije.

## Opis delovanja

Strežnik s spletno stranjo zaženemo s skripto "start\_server.py". Če skripto zaženemo brez argumentov, le-ta zažene strežnik na naslovu "localhost" in vratih številka 8080, kar pomeni da je stran dostopna samo lokalno. Če želimo, da bo stran dostopna preko spleta, moramo skripti podati argument "production".

Za zagon strežnika se uporabi funkcija ogrodja Bottle imenovana "run". Dotično ogrodje vsebuje enostavni strežnik, ki zadošča za lokalno testiranje kode ter za spletne strani z malo prometa. Če se s časom promet na strani poveča do te mere, da privzeti strežnik ne zmore več nuditi zadostnega pretoka podatkov, omogoča ogrodje Bottle enostaven prekop na ustreznejši strežnik drugega ponudnika, na primer strežnik organizacije Apache.

Ko je enkrat strežnik zagnan, posreduje ogrodje Bottle zahteve po spletnih straneh skripti, tako da s pomočjo označbe "@route" najde funkcijo, ki je zadolžena za želeni naslov spletne strani. To pomeni, da je funkcija, ki je označena z nizom "@route (' / ' )", zadolžena za ustvarjanje korenske (domače) strani, funkcija z nizom "@route (' / <sport> ' )", za ustvarjanje podstrani, ki v našem primeru predstavljajo pregled posameznih športov, itd. Naloga vsake označene funkcije je, da vrne niz, ki vsebuje želeno stran v obliki HTML. Vsa koda znotraj funkcij je neodvisna od ogrodja, kar bistveno izboljša preglednost programa.

### 2.1.5 Alternative

V tem podpoglavju bomo na kratko povzeli možne alternative izbranim tehnologijam ter zakaj se zanje nismo odločili.

## Programsko okolje

Glavne alternative programskemu okolju Python 3, ki zadostujejo zastavljenim zahtevam so:

- Okolje **Java**, ki je trenutno najbolj razširjeno programsko okolje [6]. Poleg tega se najpogosteje uporablja prav za ustvarjanje spletnih strani ter drugih vsebin, ki so dostopne preko omrežja. Razlog, zakaj nismo izbrali Jave, je jezik sam, saj je preveč gostobeseden (angl. verbose) [7] in preveč objektno usmerjen, kar upočasnjuje in otežuje razvoj enostavnejših projektov.
- Okolje **C#**, ki je zelo podobno Javi, z malo boljšo sintakso in dobrim razvojnim orodjem. Njegova glavna slabost je, da ne nudi uradno podprte verzije navideznega stroja za operacijski sistem Linux.
- Okolje **Scala**, ki uporablja navidezni stroj okolja Java in omogoča uporabo vseh javanskih knjižnic. Sintaksa jezika je veliko bolj jedrnata od sintakse programskega jezika Java in omogoča programiranje v funkcijski paradigmi. Slabost okolja je njegova nerazširjenost ter počasnost orodja za izgradnjo programov.
- Okolje **D**, ki se razlikuje od zgoraj navedenih po tem, da se koda prevede v izvršljivo datoteko, ki ne potrebuje virtualnega stroja za zagon, kar omogoča hitrejše izvajanje programa. Glavna slabost okolja je pomanjkanje razvojnih orodij.
- Okolje **Python 2**. Edini tehten razlog za razvoj projekta v drugi verziji programskega okolja Python bi bil, da katera od knjižnic ali katero od ogrodij ne bi bilo združljivo s tretjo verzijo. Ker vse knjižnice, za uporabo katerih smo se odločili, podpirajo tretjo verzijo okolja, smo izbrali le-to. Programski okolji sta sicer dokaj podobni, glavna slabost druge verzije pa je, da ne podpira standarda za kodiranje znakov UTF-8 v celoti [8], kar bi pri našem projektu lahko privedlo do težav pri zapisu in prikazu imen klubov.

## Ogrodja za generiranje spletnih strani

Kot smo že omenili obstaja veliko ogrodij, ki omogočajo ustvarjanje spletnih strani v jeziku Python. Najpopularnejša med njimi so Django, TurboGears in Web2py [9]. Glavna pomanjkljivost teh ogrodij je, da so preobširna in zato neprimerna za enostavnejše spletne strani, saj zaradi števila konstruktov in kompleksnosti uporabe preveč upočasnijo razvoj in otežujejo vzdrževanje.

## 2.2 Algoritem za rangiranje

Sistem, ki smo ga razvili v diplomskem delu je zgrajen okoli algoritma za rangiranje, ki je bil predstavljen v članku "Utilizing betting odds for rating basketball teams across different competitions" [1]. Z vidika diplomskega dela je algoritem za rangiranje obravnavan kot črna škatla. Sistem poskrbi, da se ustrezni podatki prenesejo iz podatkovne baze do algoritma za rangiranje. Slednji za rangiranje množice klubov potrebuje le ustrezno množico tekem teh klubov (o tem, kaj je ustrezna množica, v naslednjem odstavku), pri čemer je vsako tekmo potrebno opisati z nazivom domačega in gostujočega kluba ter kvotami za zmago domačinov, neodločen izid in zmago gostujočega kluba. Ko se izvede, algoritem vrne enoštevilski rang za vsak klub v množici tekem, sistem pa izračunane range zapiše v podatkovno bazo.

Kot smo že omenili, je prednost uporabljenega algoritma za rangiranje ravno upoštevanje kvot, ki nudijo več informacije o moči klubov (v primerjavi z izidi). Tako lahko, v primerjavi z obstoječimi algoritmi za rangiranje, veliko bolj natančno ocenimo razmerja moči klubov, ki ne igrajo v isti ligi, čeprav imamo na voljo le manjše število tekem, ki povezujejo klube ene in druge lige [1]. Seveda pa je bilo potrebno zgraditi nekoliko bolj zapleten sistem zbiranja potrebnih podatkov, kot bi ga, če bi potrebovali izide tekem. Čeprav sistem zbira kvote vseh stavnic, ki so na voljo za tekmo, in algoritem podpira uporabo več stavnic, trenutno uporabljamo samo povprečne kvote.

Da bi lahko ustrezno rangirali množico klubov, mora v naši množici tekem obstajati pot med poljubnima dvema kluboma. Z drugimi besedami,

---

če množico tekem predstavimo kot graf, kjer so ekipe vozlišča, povezave pa predstavljajo, da sta ekipi odigrali vsaj eno tekmo, mora obstajati pot med vsakim parom vozlišč. V primeru, da graf ni povezan, so rangi smiselni le znotraj povezanih komponent, ekip iz različnih komponent pa ne moremo primerjati med seboj. Trenutno v sistemu prikazujemo le range večjih evropskih lig, za katere vemo, da so vsako leto povezane preko evropskih klubskih tekmovanj.



## Poglavje 3

# Čelni del sistema

Čelni del sistema je sestavljen iz spletnih strani, ki omogočajo dostop do zelenih podatkov v obliki tabel, ki vsebujejo po rangih razporejene klube ali države in zemljevidov (slika 3.1). Za vsak šport imamo skupno tabelo klubov, v kateri so razporejeni vsi klubi izbranega športa. Poleg te skupne tabele imamo še tabele, ki vsebujejo klube samo iz določene države in tabele držav, v katerih so razporejene države glede na povprečen rang klubov, ki sodelujejo v tekmovanjih te države. Nad tabelo je vedno prisoten še zemljevid, ki prikazuje lokacije klubov z najvišjimi rangi in na katerem so države obarvane glede na povprečen rang klubov.

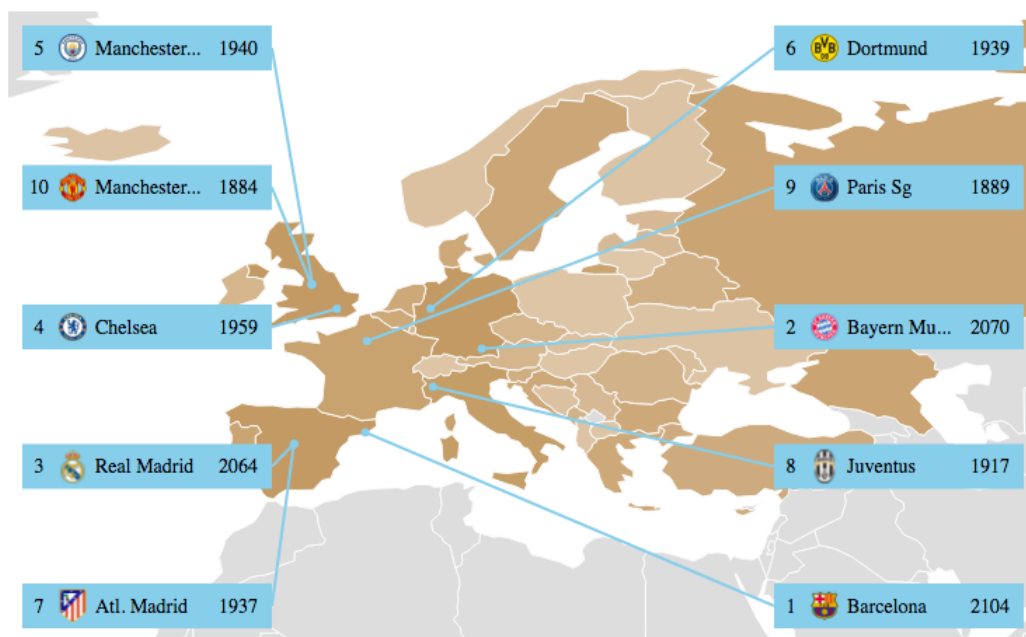
Vsi deli spletnih strani, razen zemljevidov, so definirani v jeziku HTML. Zemljevidi so definirani s pomočjo skripte JavaScript, ki se zažene v spletnem brskalniku. Skripta z uporabo knjižnic izriše zemljevid znotraj elementa SVG spletne strani.

### 3.1 Zemljevidi



Za ustvarjanje zemljevidov smo uporabili JavaScript knjižnico Datamaps, ki temelji na knjižnici D3. Vse strani, razen strani z listo športov, vsebujejo svojo različico zemljevida.

Stran določenega športa vsebuje zemljevid Evrope, na katerem so izrisane

## Soccer



## Countries

1		<a href="#">Spain</a>	1742
2		<a href="#">Germany</a>	1737
3		<a href="#">England</a>	1709
4		<a href="#">Italy</a>	1667
5		<a href="#">France</a>	1587
6		<a href="#">Russia</a>	1502
7		<a href="#">Portugal</a>	1500

Slika 3.1: Spletna stran športa.



lokacije najboljše rangiranih klubov. Zemljevid poleg lokacij klubov prikazuje tudi rang držav, in sicer z uporabo različnih barv. Boljši kot je povprečni rang ekip določene države, s temnejšim odtenkom je le-ta obarvana. Stran za prikaz športa v določeni državi vsebuje zemljevid z obrisom te države, na kateremu so označene lokacije najboljših klubov v državi. Klubska stran prav tako vsebuje obris države, iz katere izhaja klub, vendar je na njemu označena samo lokacija izbranega kluba.

### 3.1.1 Algoritem za razporejanje oznak z imeni klubov

Glavna težava pri izrisovanju zemljevidov je bila, kako razporediti oznake z imeni klubov (slika 3.1) na takšen način, da se ne prekrivajo. Ker knjižnica *Datamaps* ne nudi privzete rešitve in ker nismo našli že obstoječe rešitve, ki bi jo bilo lahko vgraditi v naš sistem, smo se odločili da sami razvijemo algoritem, ki bo omogočal dovolj pregledno razporejanje oznak.

Pri načrtovanju algoritma smo iskali najenostavnejšo rešitev, ki bi dala zadovoljivo vizualno podobo, zato smo se odločili, da bodo oznake enakomerno razporejene ob levi in desni rob zemljevida. Naloga algoritma je, da izbere katere lokacije bo povezal s katerimi oznakami, in sicer na način, da bo čim manj prekrizanih črt in da bo vsota dolžin vseh črt čim manjša. Odločili smo se, da se bomo zadovoljili z približno rešitvijo. To smo implementirali tako, da algoritem najprej razdeli zemljevid na dva dela po vertikalni osi, nato pa levi strani priredi polovico najbolj zahodno ležečih klubov, preostanek klubov pa desni. Nato algoritem poveže lokacije klubov z oznakami glede na njihovo latitudo, se pravi da se klubu ki je najvišje na zemljevidu dodeli prva oznaka od zgoraj, itd.

### 3.1.2 Uporabljene knjižnice JavaScript

#### D3

Knjižnica *D3* je namenjena izrisovanju interaktivnih diagramov, grafikonov, infografik ter zemljevidov na spletnih straneh. Za izrisovanje elementov upo-

```
1 d3.selectAll("p")
2   .style("color", "red")
3   .attr("class", "squares")
4   .attr("x", 50);
```

Izsek 3.1: Primer uporabe knjižnice D3.

```
1 var map = new Datamap({
2   scope: 'world',
3   element: document.getElementById('container1'),
4   projection: 'mercator',
5   height: 500
6   })
```

Izsek 3.2: Primer uporabe knjižnice Datamaps.

rablja konstrukte označevalnega jezika SVG (Scalable Vector Graphics), ki je del standarda HTML5. Je ena najbolj razširjenih knjižnic za ta namen, njena največja prednost pa je, da temelji na funkcijski paradigmi [11], kar omogoča veliko fleksibilnosti pri ustvarjanju in spreminjanju konstruktorov.

V izseku 3.1 je prikazan primer uporabe knjižnice, ki izbere vse HTML elemente tipa "p", jim spremeni barvo v rdečo ter jim nastavi ali spremeni vrednosti atributov "class" in "x".

## Datamaps

Knjižnica Datamaps za izrisovanje zemljevidov uporablja knjižnico D3. Njena prednost je, da nudi enostavnejši vmesnik za ustvarjanje zemljevidov, kot ga ponuja knjižnica D3.

V izseku 3.2 je prikazan primer uporabe knjižnice, ki nariše zemljevid sveta znotraj HTML elementa "container1".

## 3.2 Predstavitve strani

Hierarhija spletne strani je sestavljena iz štirih nivojev, ki zaobjemajo:

- domačo stran, na kateri je lista vseh športov,
- stran s pregledom določenega športa, ki vsebuje tabeli držav in klubov,
- stran z pregledom športa v določeni državi, ki vsebuje tabelo klubov, in
- stran določenega kluba.

### 3.2.1 Domača stran

Domača stran prikazuje tabelo športov, za katere so bili izračunani rangi. Vsak šport v tabeli je povezava na stran športa.

### 3.2.2 Stran športa

Vsak šport za katerega so bili izračunani rangi ima svojo stran (slika 3.1). Prvi element na tej strani je zemljevid Evrope, na katerem so države obarvane glede na povprečen rang klubov, ki igrajo v ligah te države. Poleg obarvanosti držav zemljevid prikazuje še lokacije desetih najbolj rangiranih klubov. Lokacije so označene z oznakami, na katerih se nahajajo logotipi, imena ter rangi klubov. Oznake služijo tudi kot povezave na klubske strani.

Pod zemljevidom se nahajata tabeli, in sicer tabela držav in tabela klubov. Tabela držav vsebuje države v katerih nastopajo rangirani klubi, in so razporejene glede na povprečni rang klubov. V tabeli klubov so razvrščeni vsi klubi glede na njihov rang. Imena držav in klubov služijo kot povezave na podstrani le-teh.

### 3.2.3 Stran države

Na spletni podstrani države (slika 3.2) se nahaja zemljevid le-te, na katerem so označene lokacije desetih najboljših klubov, in tabela klubov, ki igrajo v tej državi.

## Soccer - Italy



## Teams

1		<a href="#">Juventus</a>	1917
2		<a href="#">Napoli</a>	1839
3		<a href="#">As Roma</a>	1831
4		<a href="#">Lazio</a>	1787
5		<a href="#">Fiorentina</a>	1769
6		<a href="#">Inter</a>	1766
7		<a href="#">Ac Milan</a>	1724

Slika 3.2: Stran države.

## **Soccer - Spain - Barcelona**



### **Rankings**

Teams rating is 2104.

Team is no 1 in Europe and  
no 1 in Spain.

### **Location**



Slika 3.3: Stran kluba.

### **3.2.4 Stran kluba**

Na strani kluba (slika 3.3) se nahajajo: zemljevid države, v kateri klub nastopa (z označeno lokacija kluba), logotip in rang kluba. Poleg tega je prikazano, katero mesto zaseda klub na evropski in državni ravni.



# Poglavje 4

## Zaključek

V skladu z zahtevami podanimi v 2. poglavju smo hitro razvili sistem za prikaz rangiranja športnih klubov na podoben način, kot na primer ratingi ELO v šahu ali lestvica ATP v tenisu. Glavna prednost našega sistema pred podobnimi, že obstoječimi sistemi, je, da range ne izračunava iz rezultatov tekem, temveč iz kvot, ki so jih stavnice dodelile klubu. Takšen način omogoča natančnejše rangiranje v primeru, ko je bilo odigrano malo tekem med podnožicami klubov. Sistem sestavljajo štirje deli, in sicer: podatkovna baza, skripta za pobiranje kvot s spleta, skripta za izračunavanje rangov ter spletna stran za prikaz rangov.

Zaradi izbora programskega jezika Python, ki omogoča pisanje izredno kompaktne kode, nam je uspelo sistem napisati v približno dva tisoč vrsticah programske kode. Če bi za programski jezik izbrali katero od alternativ, ki smo jih navedli v poglavju 2.1.5, bi koda znašala vsaj dvakrat toliko vrstic. Izbrane knjižnice in ogrodja so zadovoljili naše potrebe po hitrem razvoju zanesljivega in razširljivega sistema. Edina knjižnica, ki je malo razočarala, je bila knjižnica za izrisovanje zemljevidov Datamaps, ker ne ponuja razširljive rešitve za oblikovanje in razporejanje oznak lokacij. V prid omenjeni knjižnici pa lahko štejemo, da temelji na knjižnici za izrisovanje interaktivnih vizualizacij D3, kar nam je omogočilo, da smo implementirali lasten algoritem za izrisovanje oznak, ki je uporabljal konstrukte le-te.

Edini resna pomanjkljivost, ki povzroča povečan odzivni čas spletne strani, je struktura podatkovne baze, in sicer dejstvo, da klub v svoji tabeli nima zabeležene države, v kateri nastopa. Edini način, da poizvedemo, iz katere države izhaja, je preko tabel tekma, sezona in tekmovanje. Posledica tega je, da je potrebno ustvariti SQL poizvedbo, ki vsebuje veliko ukazov "JOIN", kar znatno upočasni ustvarjanje spletnih strani. Problem bi se dalo rešiti tako, da bi v tabelo "team" dodali celoštevilski označevalec tabele "country". Poleg tega bi morali v podatkovni bazi implementirati še prožilec, ki bi ob vnosu vsake tekme preveril, ali imata nastopajoči ekipi že dodeljen označevalec države, in če ga katera od ekip ne bi imela, bi se ji le-ta dodelil z uporabo prej omenjenih ukazov "JOIN".

Druge pomanjkljivosti in možnosti za izboljšave:

- Na klubske strani bi lahko prikazali več podatkov o klubu: seznam osvojenih lovorik, rezultate zadnjih tekem, lestvico državne lige, v kateri nastopa klub, in graf za prikaz gibanja klubskega ranga. Na strani športa in na državnih straneh bi lahko dodali graf za prikaz gibanja ratingov najboljših klubov.
- Spletna stran bi lahko imela boljšo zasnovano in izgled, ki bi med drugim omogočal tudi avtomatsko prilagajanje strani širini zaslona ali brskalnika. Lahko bi tudi zamenjali pisavo, uporabili drugačen slog za tabele in prerazporedili elemente na strani kluba. Manjkata tudi glava in noga strani, ki bi lahko vsebovali logotip strani, povezave na strani športov, izjavo o avtorskih pravicah in povezavo na stran z osnovnimi podatki o projektu.
- Ker se delež klubske logotipov, ki so dostopni na strani Wikipedia, giblje okoli petdesetih odstotkov, bi za strganje logotipov lahko uporabili še kakšen drugi vir.



# Literatura

- [1] E. Štrumbelj. “Utilizing betting odds for rating basketball teams across different competitions”, v zborniku: International Conference on Sport Sciences Research and Technology Support - icSPORTS 2014.
- [2] How is the football club world ranking calculated? [Online]. Dosegljivo: <http://www.clubworldranking.com/faq/how-is-the-football-club-world-ranking-calculated.aspx>. [Dostopano 14. 8. 2016].
- [3] About [Online]. Dosegljivo: <http://clubelo.com/About> [Dostopano 14. 8. 2016].
- [4] A. Elo. *The Rating of Chessplayers, Past and Present*. Arco, 1978.
- [5] Beautiful Soup Documentation [Online]. Dosegljivo: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Dostopano 14. 8. 2016].
- [6] TIOBE Index for August 2016 [Online]. Dosegljivo: <http://www.tiobe.com/tiobe-index/>. [Dostopano 14. 8. 2016].
- [7] Languages, Verbosity, and Java [Online]. Dosegljivo: <http://www.informit.com/articles/article.aspx?p=1824790>. [Dostopano 14. 8. 2016].
- [8] Unicode HOWTO [Online]. Dosegljivo: <https://docs.python.org/2/howto/unicode.html>. [Dostopano 14. 8. 2016].

- [9] Web Frameworks for Python [Online]. Dosegljivo:  
<https://wiki.python.org/moin/WebFrameworks>. [Dostopano 14. 8. 2016].
- [10] DB-Engines Ranking [Online]. Dosegljivo:  
<http://db-engines.com/en/ranking>. [Dostopano 14. 8. 2016].
- [11] D3 Data-Driven Documents [Online]. Dosegljivo:  
<https://d3js.org>. [Dostopano 14. 8. 2016].