

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Martin Lozar

**Odprtokodni sistem za vodenje
turnirjev**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matjaž Kukar

Ljubljana 2012

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00192/2012

Datum: 09.01.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MARTIN LOZAR**

Naslov: **ODPR TOKODNI SISTEM ZA VODENJE TURNIRJEV
AN OPEN-SOURCE SYSTEM FOR TOURNAMENT MANAGEMENT**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj pregleda stanje obstoječih namiznih in spletnih aplikacij, ki se uporabljajo za vizualizacijo in vodenje turnirjev v raznih športnih in drugih igrah. Na osnovi pregleda naj zasnuje ustrezen podatkovni model in razvije lastno spletno aplikacijo za vodenje turnirjev. Razvita spletna aplikacija naj omogoča izvedbo vodenja turnirjev na katerih se uporablja vsaj en popularen turnirski sistem in točkovanje igralcev.

Mentor:

Dekan:

doc. dr. Matjaž Kukar

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Martin Lozar, z vpisno številko **63080032**, sem avtor diplomskega dela z naslovom:

Odprtokodni sistem za vodenje turnirjev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matjaža Kukarja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. januarja 2012

Podpis avtorja:

Zahvaljujem se mentorju doc. dr. Matjažu Kukarju za uso pomoč in nasvete, ki mi jih je nudil pri nastanku te diplomske naloge in svojim staršem za moralno in finančno podporo skozi vsa študijska leta.

Kazalo

Seznam uporabljenih kratic in simbolov

Povzetek

Abstract

1	Uvod	1
2	Turnirski razporeditveni sistemi	3
2.1	Pokalni sistem	3
2.2	Krožni sistem	4
2.3	Švicarski sistem	6
2.4	Svetovno prvenstvo	8
2.5	Turnirski sistemi uporabljeni pri zimskih športih	8
3	Pregled obstoječih brezplačnih aplikacij za vodenje turnirjev	9
3.1	Tournament bracket builder	9
3.2	Bracket maker	13
3.3	Challonge	15
4	Sistemi za ocenjevanje kvalitete tekmovalcev	17
4.1	Sistem ELO	17
4.2	Rangirni sistem Glicko	19
4.3	Rangirni sistem Glicko 2	22
4.4	TrueSkill	22

KAZALO

5	Tehnologije uporabljene pri izdelavi spletne aplikacije	25
5.1	HTML	25
5.2	CSS	26
5.3	JavaScript	26
5.4	MySQL	27
5.5	PHP	28
6	Razvoj spletne aplikacije	31
7	Opis funkcionalnosti spletne aplikacije za vodenje turnirjev	39
8	Zaključek	47
	Literatura	53

Seznam uporabljenih kratic in simbolov

- CSS - Cascading Style Sheets - stilna predloga na spletni strani, v kateri je zapisana oblika spletne strani
- HTML - Hyper Text Markup Language - stilna predloga na spletni strani, v kateri je zapisana oblika spletne strani
- PHP - PHP Hypertext Processor, izvirno Personal Home Page tools - skriptni programski jezik namenjen izdelavi dinamičnih spletnih strani
- SQL - Structured Query Language - strukturiran povpraševalni jezik
- API - Application Program Interface - programski vmesnik
- XML - Extensible Markup Language - razširljivi označevalni jezik
- IDE - Integrated Development Environment - integrirano razvojno okolje
- JVM - Java Virtual Machine - javanski navidezni stroj
- URL - Uniform Resource Locator - internetni naslov, na katerem se nahaja vsebina
- DOM - Document Object Model - objekt z vsebino spletnega dokumenta

KAZALO

- EER - Extended Entity-Relationship Diagram - razširjeni entitetno relacijski diagram

Povzetek

Glavna cilja diplomskega dela sta pregled obstoječih namiznih in spletnih aplikacij, ki se uporabljajo za vizualizacijo in vodenje turnirjev v raznih športih in izdelava lastne spletne aplikacije za vodenje turnirjev. Za potrebe spletne aplikacije smo izdelali podatkovni model, ki smo ga kasneje pretvorili v bazo podatkov. Razvita spletna aplikacija omogoča izvedbo vodenja turnirjev na katerih se uporablja pokalni turnirski sistem. Uporabniki, ki želijo uporabljati razvito aplikacijo, se morajo pred tem registrirati. Aplikacija vodi za vsakega tekmovalca evidenco zmag in porazov v posameznih športnih zvrsteh ter njegovo trenutno relativno moč v le teh. Uporabniku aplikacije je omogočen prikaz nekaterih statističnih podatkov, npr. število zmag, starost, država bivanja sodelujočih igralcev turnirjev itd.

Seznam ključnih besed: Spletna aplikacija, aplikacija za vodenje turnirjev, PHP, MySQL, XAMPP.

Abstract

Main objectives of the diploma thesis are an overview of existing desktop and web applications, which are used for visualization and management of tournaments in different sports and development of web application, which can be used for management of tournaments. For purposes of web application we developed data model which we later transform into database. Developed web application can be used for management of tournaments using single-elimination system. Users who want to use this web application must register. Application keeps records of victories, defeats and ratings for every user in every sport he/she has ever played on all tournaments. Users can also view some statistics data like, number of victories, age, country of players who are subscribed to tournaments, etc.

List of keywords: Web application, application for management of tournaments, PHP, MySQL, XAMPP.

Poglavje 1

Uvod

S hitrim razvojem velikega št. večigralskih spletnih iger (npr. Dota 2, Starcraft 2, Guild wars, League of legends ...) se je v zadnjih letih močno povečalo tudi število turnirjev na katerih se te igre igrajo. Večina spletnih uporabnikov raje pregleduje podatke predstavljene na zanimiv način. Zgornja ugotovitev je bila bistvena za izdelavo diplomskega dela z namenom izdelave spletne aplikacije, ki omogoča pregledovanje in vodenje raznih turnirjev. Poleg tega smo se za izdelavo tega diplomskega dela odločili tudi zaradi želje po nadgradnji znanja iz spletnega programiranja, ki ga bomo potrebovali v podjetju, ki nam je skozi študijska leta pomagalo s kadrovske štipendijo.

V naslednjem poglavju (t.j. poglavje 2) bomo predstavili nekaj najpogosteje uporabljenih in najbolj poznanih sistemov za določanje zmagovalca na različnih turnirjih. Temu poglavju bo sledilo poglavje v katerem bomo predstavili pregled že obstoječih tovrstnih aplikacij. Pregled teh aplikacij bo opisan v poglavju 3. V poglavju 4 bomo predstavili nekaj najprepoznavnejših sistemov za določanje kvalitete posameznega igralca. Poglavje 5 bo na kratko predstavilo tehnologije, ki smo jih uporabili pri razvoju spletne aplikacije. Opis orodji potrebnih za razvoj in sam razvoj spletne aplikacije bo opisan v poglavju 6. Razvito spletno aplikacijo in njene funkcionalnosti bomo predstavili v poglavju 7. V zaključku (t.j. poglavje 8) bomo podali nekaj idej, ki bi ob realizaciji pripomogle k izboljšavi uporabniške izkušnje

ob uporabi te spletne aplikacije.

Poglavje 2

Turnirski razporeditveni sistemi

Če želimo organizirati turnir za določeno športno zvrst oziroma igro, je ena od pomembnih odločitev tudi, kateri turnirski sistem določitve zmagovalca bomo na našem turnirju uporabili. Danes je v uporabi veliko število različnih turnirskih sistemov, ki pa so v osnovi mešanice ali priredbe pokalnega, krožnega in švicarskega turnirskega sistema. V nadaljevanju bomo opisali nekaj najpomembnejših in najpogosteje uporabljenih turnirskih sistemov.

2.1 Pokalni sistem

Pokalni sistem (angl. Single-elimination, uporabljajo se tudi naslednji angl. izrazi: knockout, cup ali sudden death tournament), imenovan tudi izločilni sistem, se uporablja za določitev zmagovalca na raznih turnirjih. Dobra lastnost turnirja je neomejenost števila prijavljenih tekmovalcev oziroma ekip. Če je število prijavljenih različno od 2^x , se najprej izvedejo kvalifikacije v katerih je izločeno toliko tekmovalcev, da jih v turnirju preostane le 2^x . Po kvalifikacijah se določijo dvojice, ki se bodo pomerile v naslednjem krogu turnirja. Te so lahko naključno izbrane, vendar so pri večini turnirjev določene tako, da se prvi nosilec pomeri z najnižje uvrščenim igralcem, drugi nosilec se pomeri z drugim najnižje uvrščenim tekmovalcem itd. Poleg tega so dvojice ponavadi razporejene tako, da se prvi in drugi nosilec turnirja lahko

pomerita komaj v finalnem obračunu, če se uspeta vanj kvalificirati. Do zmagovalca turnirja pridemo dokaj hitro. Igralec oziroma ekipa, ki izgubi tekmo je takoj izločena. Zmagovalec turnirja je tekmovalec oziroma ekipa, ki je zmagala na vseh tekmah. Sistem je pogosto uporabljen na turnirjih, na katerih je prijavljenih veliko tekmovalcev ali pa organizator želi turnir zaključiti v kratkem času (npr. v 1 dnevu). Kot je v viru [15] zapisano je ena od slabosti sistema, da prikaže le zmagovalce turnirja (obstajajo tudi različice, kjer se odigra tekma za 3. in 4. mesto). Pokalni sistem se ponavadi uporablja na šahovskih, teniških, poker in turnirjih raznih večigralskih spletnih videoiger. V razviti spletni aplikaciji za vodenje turnirjev smo izbrali ta turnirski razporeditveni sistem.

Obstaja tudi različica (angl. Double-elimination) pri kateri je tekmovalec izločen po 2 izgubljenih tekmah. Ko torej tekmovalec izgubi prvo tekmo, je avtomatsko uvrščen v “nižjo” vejo, kjer se v naslednjem krogu pomeri proti drugemu tekmovalcu, ki je prav tako prejšnjo tekmo izgubil. Primer takega turnirskega drevesa je prikazan na sliki 3.3 v poglavju “Tournament bracket builder”. Pri tej različici ima tekmovalec, ki je izgubil eno tekmo še vedno možnost zmage na turnirju.

V razviti spletni aplikaciji za vodenje turnirjev smo za razporeditev in določitev zmagovalca izbrali ta sistem. Razlog je v tem, da večina turnirjev v raznih spletnih večigralskih igrah uporablja ta razporeditveni sistem. Poleg tega je ta sistem tudi eden najrazumljivejših in najzanimivejših za vizualizacijo.

2.2 Krožni sistem

Pri krožnem sistemu (angl. Round robin) vsak tekmovalec igra z vsakim. Obstaja tudi dvojni krožni sistem (angl. Double round robin), kjer vsak igralec igra dvakrat z vsakim nasprotnikom. Ta sistema veljata za najpravičnejša. Za določanje dvojic se največkrat uporablja Bergerjev (imenovan po avstrijskem šahistu Johannu Bergerju) sistem določanja parov. Z žrebom se tek-

	1.Dvojica	2.Dvojica	3.Dvojica
1. Krog	2-5	3-4	1-(6)
2. Krog	1-2	5-3	(6)-4
3. Krog	3-1	4-5	2-(6)
4. Krog	1-4	2-3	(6)-5
5. Krog	5-1	4-2	3-(6)

Tabela 2.1: Primer Bergerjeve razporeditvene tabele za krožni turnir s 5 ali 6 tekmovalci.

movalcem določi številke. Pari se določijo tako, da igralec, ki ima največjo številko igra z igralcem z najmanjšo. Če število igralcev ni parno, igralec z najnižjo številko v prvem krogu ne igra. Tabela 2.1 prikazuje primer Bergerjeve razporeditvene tabele parov za krožni turnirski sistemi s 5 ali 6 sodelujočimi. V primeru, ko na turnirju sodeluje 5 tekmovalcev se v posameznem krogu igrata le 2 igri; tekmovalec, čigar številka ni v oklepaju, ta krog ne sodeluje. Slika 2.1 prikazuje turnir (s štirimi tekmovalci) na katerem se uporablja krožni razporeditveni sistem.

Pri šahu se s pomočjo Bergerjevih razporeditvenih tabel določijo tudi barve figur posameznega igralca. Ko imata oba igralca v paru sodi številki, pripadajo bele figure tistemu z višjo številko. V primeru, ko ima en igralec iz para sodo, drugi pa liho številko dobi bele figure tisti z nižjo številko. Pri parnem številu tekmovalcev ima zgornja polovica igralcev bele, spodnja pa črne figure. Slabost tega sistema je le v tem, da je potrebno za določitev zmagovalca odigrati veliko število dvobojev, za kar pa je potrebno veliko časa. Če na takem turnirju (krožni sistem) sodeluje n tekmovalcev, je potrebno za določitev zmagovalca odigrati $(n/2)*(n-1)$ iger. Vsebina poglavja je povzeta po viru [11].

2.3 Švicarski sistem

Razporeditveni sistem, ki je prikazan na sliki 2.1 je bil prvič uporabljen leta 1895 na šahovskem turnirju v Švici (natančneje v Zürichu). Izumil ga je J. Muller. Švicarski sistem (angl. Swiss-system tournament) se ponavadi uporablja, ko se na določen turnir prijavi večje število tekmovalcev ali ekip in bi bil turnir z uporabo krožnega sistema časovno predolg. Iz vira [13] je razvidno, da za razliko od pokalnega sistema tekmovalci v švicarskem sistemu ne izpadajo postopno, ampak vsi odigrajo določeno število iger. Zmagovalca turnirja se določi na podlagi končnih zbranih točk. Vsak igralec lahko dobi določeno število točk ob koncu vsake igre glede na doseženi rezultat. Število točk ni točno določeno. Na nekaterih turnirjih dobijo točke le igralci, ki so igro v določenem krogu zmagali, na drugih pa dobijo nekaj točk tudi tekmovalci, ki so tekmo v določenem krogu remizirali. Dvojice se v prvem krogu lahko določijo naključno, lahko s pomočjo uporabe Bergerjevih razporeditvenih tabel (tekmovalci lahko žrebajo lastne številke ali pa se namesto teh števil upošteva njihova ELO moč - npr. pri šahu). Organizator lahko določi še kakšen drug sistem (npr. prvi nosilec igra z drugim, tretji s četrtem itd.). Po zaključenem krogu so tekmovalci razporejeni v skupine glede na dosežen rezultat. Npr. zmagovalci so uvrščeni v skupino 1, igralci, ki so tekmo remizirali v skupino 1/2, poraženci pa v skupino 0. Torej v vsakem naslednjem krogu igrajo igralci proti tekmovalcem, ki so dosegli podoben rezultat (igralci tekmujejo proti tekmovalcem iz iste skupine). Če je število igralcev znotraj skupine 1 liho, igra najnižje uvrščeni tekmovalec iz skupine 1 z najbolje uvrščenim igralcem iz skupine 1/2, če le-ta obstaja, sicer pa igra z najbolje uvrščenim tekmovalcem iz skupine 0. Na turnirjih, kjer se uporablja Švicarski sistem se tekmovalci pogosto poslužujejo t.i. švicarskega gambita, kjer (ponavadi slabše rangirani igralci) nameroma žrtvujejo zmago nasprotniku (v začetnih krogih turnirja). Na ta način so v nadaljevanju turnirja predvidoma uvrščeni v nekoliko šibkejše skupine (tekmovalci si s tem poskušajo priboriti nekoliko boljšo končno uvrstitev).

CHALLENGE!
Facepunch EU Group 2

Round 1

1	Strom	0
4	Firestarter	0
2	Xera	0
3	Lukepker	1

Round 2

1	Strom	1
3	Lukepker	0
4	Firestarter	0
2	Xera	1

Round 3

1	Strom	1
2	Xera	0
3	Lukepker	0
4	Firestarter	1

	Played	Remaining	Match Wins	Match Ties	Games Won	Total Points in Games
Strom	2	1	2	0	2	2
Xera	3	0	1	0	1	1
Lukepker	3	0	1	0	1	1
Firestarter	2	1	1	0	1	1

Slika 2.1: Primer izgleda turnirja na katerem se uporablja krožni razporeditveni sistem. Slika je generirana s spletno aplikacijo Challenge, ki je natančneje opisana v poglavju 3.3.

2.4 Svetovno prvenstvo

Pri tej vrsti turnirja (angl. Group stage ali pool play ali pool stage) so udeleženci najprej razporejeni v skupine. Znotraj teh skupin igra vsak tekmovalec z vsakim nasprotnikom posebej (krožni sistem). V nadaljno tekmovanje ponavadi napredujeta prva dva najbolje uvrščena tekmovalca iz posamezne skupine (tekmovalci, ki so se v skupinskem delu najbolj odrezali). Po končanem skupinskem delu se zmagovalec določi z uporabo pokalnega sistema. Ta sistem uporabljajo organizatorji svetovnih prvenstev v nogometu, hokeju, košarki, rokometu in verjetno še v kakšnem drugem športu. Vsebina poglavja je povzeta po viru [6].

2.5 Turnirski sistemi uporabljeni pri zimskih športih

Ta sistem (angl. Group tournament ranking system) deloma spominja na Švicarski sistem (ponavadi imajo turnirji te vrste določeno število tekem ali krogov ter na teh turnirjih ni izpadanja). Turnirji te vrste so ponavadi dolgotrajnejši (trajajo tudi več mesecev, npr. celo zimo). Tekmovalci imajo pravico nastopati na vseh tekmah. Na vsaki tekmi tekmovalci dobijo določeno število točk glede na svojo uvrstitev. Zmagovalca turnirja določi skupni seštevek točk, ki jih je posamezen tekmovalec zbral v celotni sezoni (na vseh tekmah). Ta sistem je uporabljen pri alpskih in nordijskih smučarskih disciplinah.

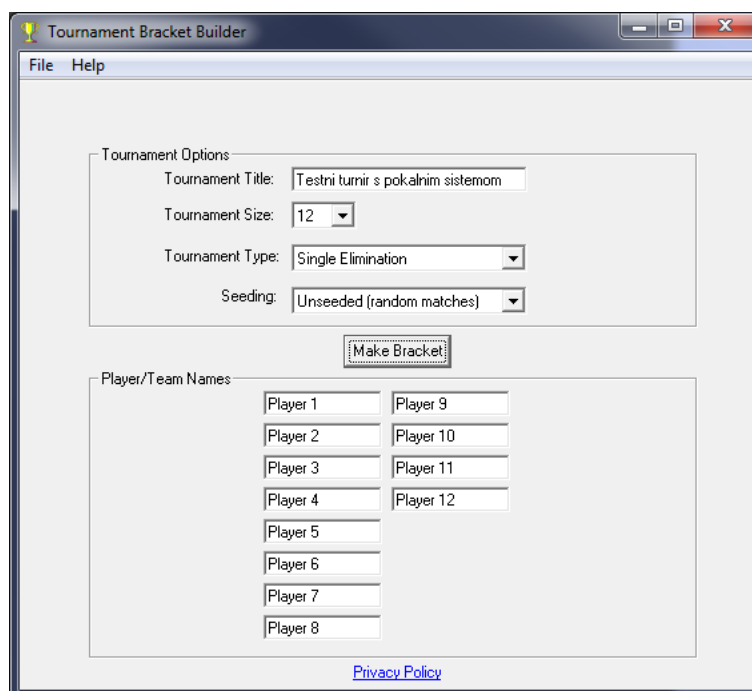
Poglavje 3

Pregled obstoječih brezplačnih aplikacij za vodenje turnirjev

V tem poglavju bomo predstavili nekaj obstoječih aplikacij, ki služijo vizualizaciji (drevesni prikaz uvrstitev posameznih tekmovalcev na določenem turnirju) in/ali administraciji turnirjev. V poglavju se bomo omejili na aplikacije, ki so brezplačne in vsem prosto dostopne.

3.1 Tournament bracket builder

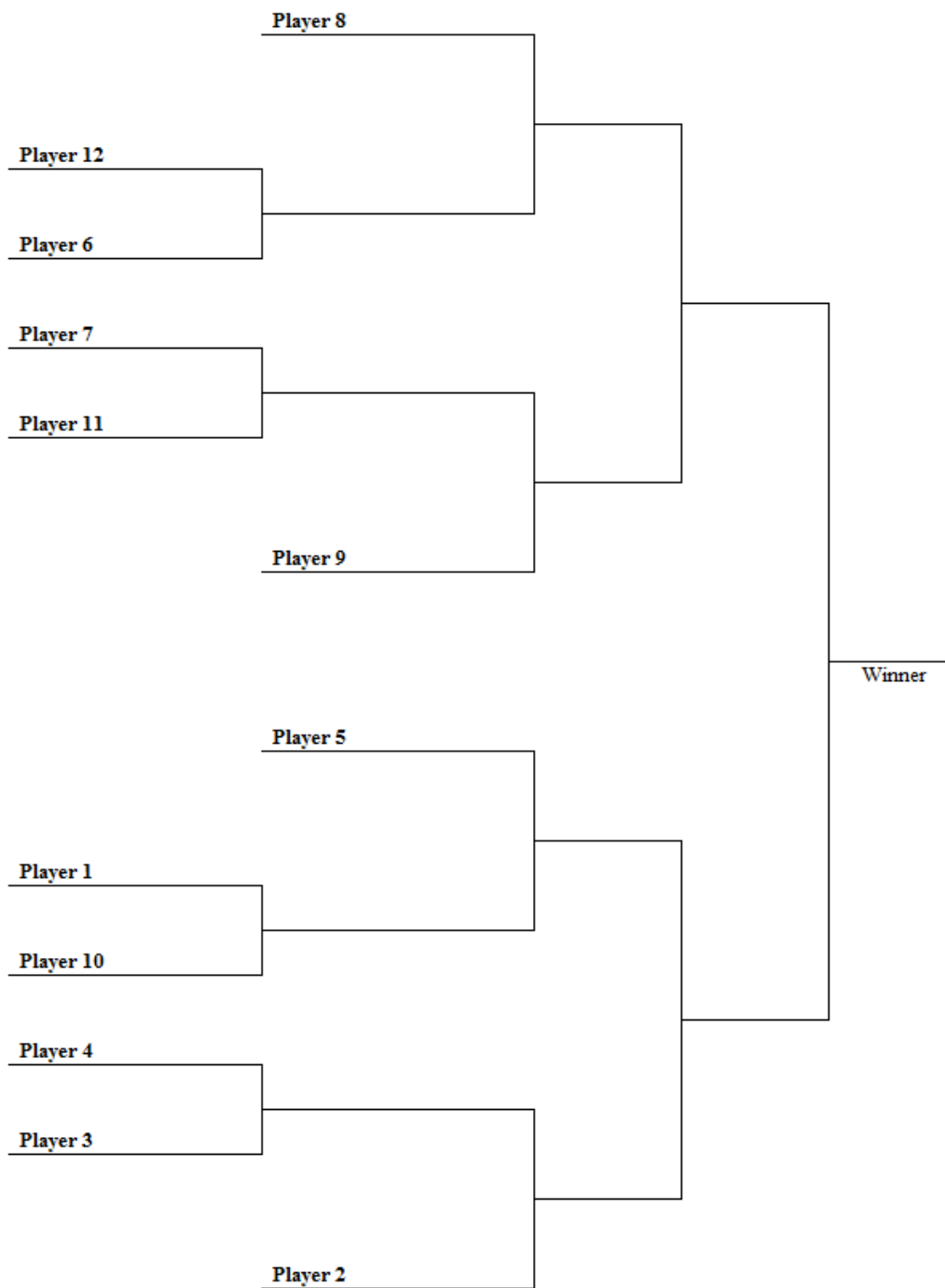
Tournament bracket builder je brezplačna namizna aplikacija, s katero lahko ustvarimo vizualizacijo drevesa, ki predstavlja potek turnirja (v nadaljevanju “PDT” ali “prikazno drevo turnirja”). V meniju, kjer kreiramo PDT lahko določimo maksimalno število udeležencev turnirja (minimalno število udeležencev je 3, maksimalno pa 32). Izbiramo lahko med klasičnim pokalnim sistemom in različico pokalnega sistema pri kateri mora vsak tekmovallec izgubiti 2 tekmi preden je izločen iz tekmovanja (angl. Double elimination - primer takega turnirja je prikazan na sliki 3.3). Vnesemo lahko tudi imena udeležencev, ki bodo na turnirju sodelovali. Poleg tega lahko izbiramo med možnostjo naključnega generiranja dvojic, ki se bodo pomerile v prvem krogu ali pa dvojice določimo sami. Ob pritisku gumba “Make



Slika 3.1: Grafični vmesnik aplikacije Tournament bracket builder z vnešenimi nastavitvami, ki generirajo prikazno drevo turnirja na sliki 3.2.

Bracket” aplikacija generira HTML dokument s HTML kodo, ki predstavlja PDT. HTML dokument se shrani v mapo, kjer smo namestili aplikacijo. Slika 3.1 prikazuje aplikacijski grafični vmesnik. Slika 3.2 prikazuje PDT, ki ga Tournament bracket builder generira. Aplikacija je dostopna na naslovu: <http://tournamentbracketbuilder.uuuq.com/>.

Glavna slabost aplikacije je neobstoj grafičnega vmesnika s pomočjo katerega bi lahko uporabnik sprotno vnašal rezultate tekem. Poleg tega je omejitev velikosti turnirja nekoliko preskromna (maksimalno število tekmovalcev je omejeno na 32).



Slika 3.2: PDT generirano z aplikacijo Tournamnet bracket builder.



Slika 3.3: Primer turnirja na katerem se uporablja različica pokalnega sistema z dvojnimi izpadanjem (angl. Double elimination).

3.2 Bracket maker

Bracket maker je brezplačna spletna aplikacija namenjena kreiranju PDT. Za uporabo aplikacije se je potrebno najprej registrirati in registracijo potrditi z obiskom naslova, ki nam ga aplikacija pošlje na vnešeni elektronski naslov. Aplikacija omogoča pregled nad vsemi turnirji, ki smo jih sami kreirali v določenem preteklem časovnem obdobju (privzeta vrednost je 3 mesece). Pomankljivost pri tem delu aplikacije je nekoliko preokorno (pomankanje natančnosti) sortiranje lastnih podatkov. Lastništvo določenega lastnega turnirja lahko tudi predamo nekemu drugemu uporabniku. Vsak ustvarjen turnir lahko tudi urejamo (npr. spremenimo lahko kategorijo, napise nad posameznim krogom turnirja - to so na sliki 3.4(a) vnosna polja nad katerimi je odebeljen napis Header #, itd.). Slika 3.4(a) prikazuje grafični vmesnik za vnos podatkov o izbranem turnirju. Aplikacija omogoča dokaj široko izbiro različnih tipov eliminacij (enojno in dvojno izločanje, izločanje po 4 porazih itd.).

Ko se določen turnir zaključi in smo vnesli vse podatke tega turnirja, je izgled generiranega drevesa podoben rezultatu na sliki 3.4(b). To sliko lahko naknadno objavimo tudi na kakšni drugi spletni strani. Ta opcija zahteva mesečno naročnino te spletne aplikacije, ki je dostopna na:

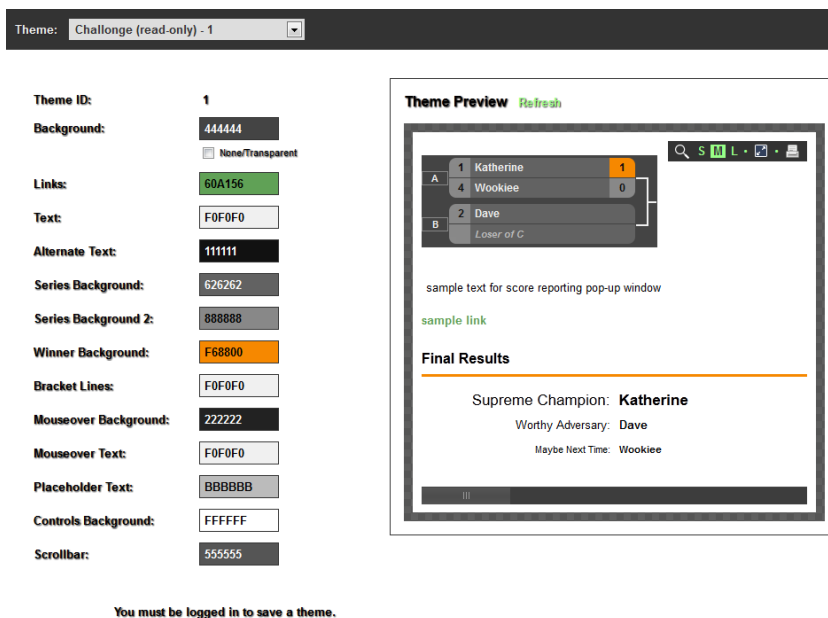
<http://www.bracketmaker.com/>.

3.3 Challenge

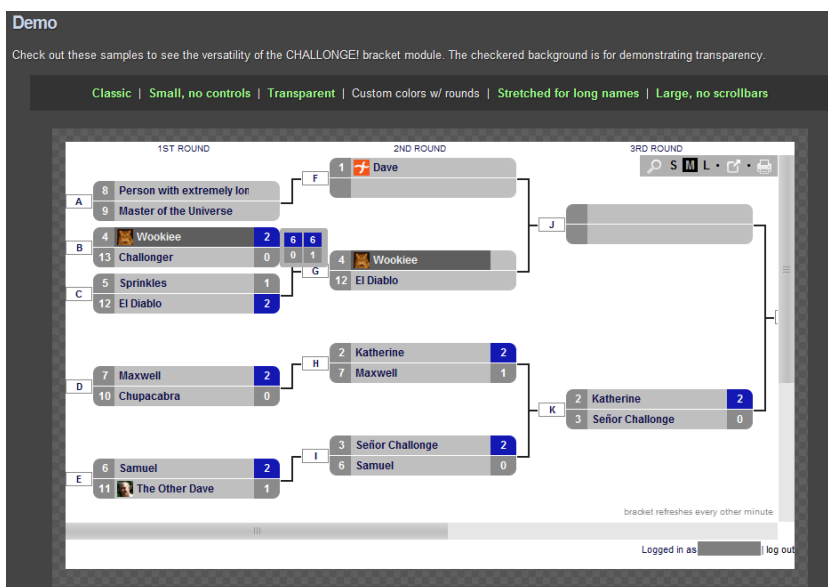
Challenge je uporabnikom na voljo brezplačno. Obstaja tudi plačljiva izvedba pri kateri aplikacija uporabnikom ne prikazuje raznih oglasov, omogoča pa jim spreminjanje tem in izgleda drevesa. Urejevalnik teh nastavitvev je viden na sliki 3.5(a). Glavna prednost plačljive izvedbe je predvsem možnost objave drevesa, poteka turnirja na neki drugi spletni strani. Pri plačljivi različici uporabnik kupi določeno število žetonov, ki se porablja, ko si nek uporabnik ogleda drevo določenega turnirja objavljeno na drugi spletni strani (1 ogled stane 1 žeton). Uporabnik se mora pred uporabo aplikacije registrirati ali identificirati z obstoječim Facebook uporabniškim računom. Vsak uporabnik ima možnost urejanja svojega profila (urejanje osebnih podatkov, možnost nalaganja lastnega portreta, nastavitve ure glede na to v katerem časovnem pasu prebiva, spreminjanje obstoječega gesla za vstop v spletno aplikacijo, različne možnosti nastavljanja obvestil - npr. ko ga drug uporabnik povabi na turnir itd.). Aplikacija mu tudi omogoča pregled nad lastnimi nakupi žetonov (zgodovina transakcij med ponudnikom in uporabnikom).

Mogoč je še prikaz dreves turnirjev v katerih lahko sodeluje do 256 igralcev. Uporabnik, ki želi kreirati nov turnir, lahko izbira med enojnim, dvojnim in krožnim sistemom (angl. Round-robin tournament) izpadanja. Lahko se določi tudi to ali se bodo dvojice (igralcev ali ekip), ki se bodo pomerile v prvem krogu turnirja naključno izžrebale oziroma bodo določene na tak način, da najboljši igralce oziroma ekipa igra z najnižje uvrščenim. Slika 3.5(b) prikazuje nekaj funkcionalnosti: določitev velikosti prikaza, možnost tiskanja itd., ki jih ponuja spletna aplikacija. Uporabniki te spletne aplikacije lahko tudi medsebojno komunicirajo preko zasebnih sporočil. Spletna aplikacija je dostopna na: <http://challonge.com/>.

POGLAVJE 3. PREGLED OBSTOJEČIH BREZPLAČNIH APLIKACIJ
 ZA VODENJE TURNIRJEV



(a)



(b)

Slika 3.5: Challenge-jev urejevalnik prikaza turnirskega drevesa (a) in primer prikaza drevesa uvrstitev na testnem turnirju generiranem s spletno aplikacijo Challenge (b).

Poglavje 4

Sistemi za ocenjevanje kvalitete tekmovalcev

V tem poglavju bomo predstavili nekaj najpogosteje uporabljenih sistemov, namenjenih ocenjevanju kvalitete igralcev. S pomočjo števil in matrik, ki opisujejo posameznega igralca določene igre, lahko že pred tekmovanjem napovemo rezultat tekme. Navedli bomo nekaj športnih disciplin ali tekmovanj na katerih se opisani sistemi uporabljajo.

4.1 Sistem ELO

ELO rangirni sistem omogoča primerjavo relativne moči posameznega igralca ali ekipe v igrah, kjer sodelujeta 2 igralca oziroma 2 ekipi. Relativna moč je celoštevilaska vrednost. Višja kot je ta številka, boljši je opazovani igralec oziroma ekipa. S pomočjo primerjave razlik relativne moči (v nadaljevanju tudi ratinga) 2 igralcev lahko približno napovemo rezultat tekme. Pričakovani rezultat tekme za igralca je seštevek verjetnosti zmage igralca 'a' proti igralcu 'x' in polovica verjetnosti remija tekmovalca 'a' proti tekmovalcu 'x'.

ELO sistem je izumil Árpád Élő (1903-1992), na Madžarskem rojeni ameriški profesor fizike. Prvotno je bil ELO sistem namenjen primerjanju relativne moči med igralci šaha. Svetovna šahovska federacija (angl. World

Chess Federation) je ELO sistem uvedla leta 1970. Številke relativne moči na svetovni šahovski lestvici se gibljejo med 0 in 2800. Danes se ta sistem pogosto uporablja tudi v večigralskih spletnih videoigrah (Guild wars, League of legends, Counter Strike: Global Offensive, ...), ameriškem nogometu, košarki itd. Relativno moč igralca ali ekipe po odigrani tekmi lahko izračunamo po naslednjih formulah:

Formula 4.1 *Nova relativna moč igralca a se izračuna po odigranem dvojboju.*

$$R'_a = R_a + K(S_a - E_a) \quad (4.1)$$

Formula 4.2 *Pričakovani rezultat za igralca a v primerjavi z igralcem x.*

$$E_a = \frac{1}{1 + (10^{\frac{R_a - R_x}{400}})} \quad (4.2)$$

Formula 4.3 *Pričakovani rezultat za igralca a v primerjavi z igralcem x na še en način.*

$$E_a = \frac{10^{\frac{R_a}{400}}}{10^{\frac{R_a}{400}} + 10^{\frac{R_x}{400}}} \quad (4.3)$$

- R'_a - nova relativna moč igralca a.
- R_a - trenutna relativna moč igralca a.
- E_a - pričakovani rezultat za igralca a v primerjavi z igralcem x.
- S_a - konstanta katere vrednost je:
 - 1 - če tekmovalec a zmagata proti tekmovalcu x.
 - 0,5 - če tekmovalec a remizira z igralcem x.
 - 0 - če tekmovalec a izgubi proti tekmovalcu x.
- K - izravnalna konstanta, ki ima ponavadi vrednost 16 za šibkejše igralce in 32 za močnejše.

Iz zgoraj opisanega sledi tudi: $E_a + E_x = 1$. Relativna moč tekmovalca, ki je tekmo izgubil se bo zmanjšala, relativna moč zmagovalca se bo povečala. V primeru, da igralec a premaga igralca x, ki je precej bolje rangiran se bo moč igralca a občutno izboljšala, relativna moč igralca x pa se bo zmanjšala za enako vrednost kot se je igralcu a povečala. V primeru, ko pa precej bolje uvrščeni igralec zmaga proti slabšemu igralcu, bo bolje uvrščeni igralec pridobil razmeroma majhno izboljšavo svoje relativne moči (poraženčeva relativna moč se bo za isto vrednost zmanjšala).

V razviti spletni aplikaciji za vodenje turnirjev smo za primerjanje relativnih moči igralcev izbrali ta rangirni sistem. Aplikacija možnosti remija na določeni tekmi ne omogoča - možnost remija v pokalnem turnirskem sistemu namreč ni možna. Konstanto K pa smo postavili na 32, ker smo za začetek hoteli nekoliko pohitriti spreminjanje ratingov tekmovalcev. V spletni aplikaciji se vsakemu igralcu, ki še ni igral določene igre na prijavljenem turnirju dodeli rating 1200. S pomočjo te predpostavke si nato pomagamo pri določanju dvojic, ki se bodo na turnirju pomerile v prvem krogu.

4.2 Rangirni sistem Glicko

Sistem je bil sprva namenjen uporabi na šahovskih turnirjih. Razvil ga je Mark Glichman kot popravek k že obstoječemu ELO sistemu. Problem ELO sistema je zanesljivost relativne moči igralca, ki v določenem časovnem obdobju ni tekmoval. Za primer si zamislimo 2 igralca šaha, ki imata enako relativno moč. Prvi igralec je zadnjo tekmo odigral včeraj, drugi igralec pa je svojo zadnjo igro odigral pred 1 letom. Če drugi igralec premaga prvega, se bo njegova relativna moč enako izboljšala, kot v primeru, če bi igro dobil prvi igralec. Sistem Glicko pa v primeru, ko zmaga igralec, ki daljše časovno obdobje ni odigral nobene tekme dobi več moči kot, če bi tekmo dobil prvi igralec, ki je svojo zadnjo tekmo odigral včeraj. Drugače rečeno, sistem Glicko upošteva in drugače vrednoti igralce, ki aktivno igrajo in igralce, ki so se k aktivnemu tekmovanju vrnili po določenem časovnem premoru.

Sistem Glicko uvaja novo vrednost RD^1 (deviacija ratinga), ki meri zanesljivost ratinga. Za primer imamo tekmovalca z ratingom 1500 in RD 50. Glickov sistem za takega igralca pravi, da je njegova relativna moč (rating) nekje v intervalu med 1400 in 1600 s verjetnostjo 95%. Zgornjo mejo intervala določimo tako, da igralčevi trenutni moči prištejemo $2 * RD$, spodnjo mejo pa tako, da igralčevi trenutni moči odštejemo $2 * RD$. Po odigrani igri se igralčeva moč spremeni glede na velikost RD . Višji kot je RD bolj se spremeni relativna moč, nižji kot je RD manjša je relativna moč. Sprememba relativne moči je torej odvisna od zanesljivosti trenutne moči. Skratka, nižji RD označuje točnejšo relativno moč (interval relativne moči je manjši). Po odigranem dvoboju novo relativno moč in RD določimo z uporabo formul 4.4 do 4.6.

Formula 4.4 *Določimo trenutni RD .*

$$RD = \min(\sqrt{RD_x^2 + c^2 * t}, 350) \quad (4.4)$$

- RD - nov RD .
- RD_x - star RD .
- t - število časovnih period od zadnje tekmovalčeve tekme. To število ponavadi določi organizator turnija glede na pogostost določanja novih iger (npr. čas potreben za izvedbo enega kroga turnirja je lahko 1 časovna preioda). Glickov sistem deluje najboljše, ko je število časovnih period med 5 in 10.
- c - konstanta, ki določa verjetnost, da je igralčeva relativna moč res taka kot se kaže (verjetnost da je igralčeva moč res na takem nivoju kot to kaže njegov rating). To konstanto se ponavadi določi z analizo preteklih rezultatov in sprememb le teh. Lahko pa jo določimo tudi z rešitvijo enačbe: $350 = \sqrt{50^2 + 100 * c^2}$ (upoštevamo, da je za tipičnega igralca RD enak 50).

¹RD angl. Ratings deviation

Novemu igralcu v igri, ki za rangiranje igralcev uporablja rangirni sistem, Glicko se dodeli relativna moč (rating) 1500 in RD 350.

Formula 4.5 *Priredimo novo vrednost relativne moči za vsakega igralca.*

$$r' = r + \left(\frac{q}{\frac{1}{RD^2} \frac{1}{d^2}} \right) \sum_{i=1}^m g(RD_i)(s_i - E(s \mid r, r_i, RD_i)) \quad (4.5)$$

Formula 4.6 *Priredimo novo vrednost RD za vsakega igralca.*

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2} \right)^{-1}} q = \frac{\ln 10}{400} = 0,0057565 \quad (4.6)$$

$$q(RD) = \frac{1}{\sqrt{1 + 3 * q^2 * \frac{RD^2}{\pi^2}}} \quad (4.7)$$

$$E(s \mid r, r_i, RD_i) = \frac{1}{1 + 10^{-\frac{g(RD_i)(r-r_i)}{400}}} \quad (4.8)$$

- r_i - relativna moč i -tega igralca.
- s_i - predstavlja izid določene igre. Za zmago se s_i postavi na 1, za remi $1/2$ in za poraz na 0.
- RD' - nova vrednost RD.
- M - število tekmovalcev, ki je odigralo dvoboj v časovni periodi.
- r' - nova relativna moč (rating).

Ti izračuni se izračunajo za vsakega igralca posebej, če je slednji v določeni časovni periodi odigral tekmo.

Sistem Glicko se od sistema ELO razlikuje tudi v neenakomernosti spremembe relativne moči zmagovalca in poraženca. Pri sistemu ELO se poraženčeva relativna moč zmanjša za isto vrednost kot se zmagovalcu poveča. Pri sistemu Glicko pa to ponavadi ne drži. Na spremembo relativne moči vpliva tudi velikost RD . Glavna slabost sistema Glicko se pokaže, ko tekmovalec tekmuje zelo pogosto. V tem primeru je njegov RD zelo nizek in posledično se njegova moč ne spreminja veliko (RD in relativna moč ne odražata zelo natančno igralčevega napredka). Vsebina tega poglavja je povzeta iz vira [15].

4.3 Rangirni sistem Glicko 2

Ta sistem je nadgradnja sistema Glicko. Kot lahko že iz imena sklepamo, je avtor tega sistema prav tako Mark Glichman. Sistem vpeljuje novo spremenljivko σ , ki označuje spremenljivost (volatilnost) relativne moči posameznega igralca. Volatilnost je visoka, ko je določen igralec v nekem časovnem obdobju zmagal veliko dvobojev (njegova relativna moč se je precej spremenila); nizka pa je, ko so rezultati določenega igralca konsistenti (npr. enako zmag in porazov - igralčeva moč se ni bistveno spremenila).

Rangirni lestvici sistemov Glicko in Glicko 2 se nekoliko razlikujeta, vendar lahko rezultate iz ene lestvice dokaj enostavno pretvorimo v drugo in obratno. Naslednji koraki predpostavljajo, da so relativne moči vzete iz lestvic sistema Glicko. Te relativne moči so potem pretvorjene v skalo sistema Glicko 2 in nazaj v originalno skalo (Glicko).

Vsebina tega poglavja je povzeta po viru [4].

4.4 TrueSkill

TrueSkill je algoritem osnovan na Bayesovem pravilu. Razvilo ga je podjetje Microsoft kot nadgradnjo sistema Glicko. Idejno ta sistem zelo spominja na sistem Glicko 2. Igralčeva relativna moč je predstavljena kot normalna (ali Gaussova) distribucija, ki je karakterizirana s predvideno relativno močjo igralca - μ in zanesljivostjo predvidenega igralca - σ . Po vsaki odigrani tekmi se verjetnost pravilnosti predvidene relativne moči igralca poveča. Igralčevo relativno moč računamo po spodnji formuli:

Formula 4.7 *Nova relativna moč igralca izračunana z uporabo TrueSkill sistema.*

$$R = \mu - 3 \times \sigma \tag{4.9}$$

Ta sistem trenutno uporablja Microsoft znotraj spletnega servisa Xbox Live. Ta uporabnikom ob plačilu mesečne naročnine omogoča igranje raznih spletnih večigralnih iger (igre, ki jih ta servis podpira) na igralnih konzolah

Xbox in platformah Windows (XP, 7, Phone). Ko uporabnik prvič igra določeno igro, se mu vredosti spremenljivk nastavijo na sledeče vrednosti: $R = 0$, $\sigma = 25/3$ in $\mu = 25$. Dobra lastnost tega sistema v primerjavi s sistemom ELO je v tem, da je potrebno za določitev realnega igralčevega ratinga manj iger.

Poglavje 5

Tehnologije uporabljene pri izdelavi spletne aplikacije

V tem poglavju bomo na kratko opisali tehnologije, ki smo jih uporabili za izdelavo spletne aplikacije. Te so: HTML, CSS, JavaScript, Php in MySQL. Aplikacijo smo zasnovali in razvili sami in v njen nismo uporabljali tujih knjižnic z izjemo googlove knjižnice Google Chart Tools¹ s pomočjo katere smo izrisali grafa in tabeli za prikaz statističnih podatkov.

5.1 HTML

HTML (angl. Hyper Text Markup Language) je označevalni jezik, namenjen razvoju spletnih strani. Predstavlja ogrodje vsakega spletnega dokumenta. HTML kodo lahko danes pišemo v vsakem urejevalniku besedil. HTML sintaksa je dokaj preposta in enostavna za razumevanje in učenje. Gradniki jezika HTML so značke (angl. tags). Tukaj gre za določene rezervirane besede, ki jih pišemo med “<” in “>”. Določenim značkam lahko dodatno specificiramo določene lastnosti. Sintaksa HTML značke z atributi je sledeča: <rezervirana_HTML_beseda atribut_1 = “vrednost” atribut_2 = “vrednost

¹Google Chart Tools dostopno na:

<https://google-developers.appspot.com/chart/interactive/docs/index>

2" ... atribut_n = " vrednost n">. Za primer podajmo značko, ki določa začetek tabele (rezervirana beseda "table") z atributoma, ki določata prikaz okvirja tabele (atribut "border") in razmik med celicami, ki sestavljajo tabelo (atribut "cellspacing"): <table border="1" cellspacing="0">. Vrednosti atributov lahko v HTML kodi pišemo med enojnimi ali dvojnimi narekovaji.

5.2 CSS

CSS (angl. Cascading Style Sheets) so predloge, ki določajo izgled skupine ali posameznih HTML elementov. Danes jih podpirajo vsi spletni brskalniki, ki pa se v prikazu nekaterih elementov razlikujejo. CSS kodo lahko v HTML dokument vstavimo na več različnih načinov. Lahko jo vključimo s pomočjo zunanje CSS datoteke, lahko jo pišemo v glavo HTML dokumenta (znotraj značke <style type="text/css">, ki se nahaja znotraj značke <HEAD>) ali pa direktno na značko znotraj telesa HTML dokumenta (HTML elementi v notranjosti značke <BODY>. Npr. <div style="background-color: black">). Jezik CSS je neodvisen od jezika HTML in ga lahko uporabljamo v povezavi z vsemi jeziki, ki bazirajo na XML ozanačevalni strukturi.

5.3 JavaScript

JavaScript je objektni skriptni jezik. Razvilo ga je podjetje Netscape leta 1995, ki ga je sprva imenovalo Mocha. Kasneje so jezik preimenovali v LiveScript, nekaj mesecev za tem pa v JavaScript. Oče tega jezika je Brendan Erich. Jezik pomaga razvijalcem spletnih aplikacij pri gradnji (razvoju) dinamičnih spletnih strani. JavaScript kodo lahko pišemo v HTML dokument, ki predstavlja ogrodje strani ali ga v ta dokument vključimo (zunanja datoteka z JavaScript kodo). Možne so tudi kombinacije obojega. Naloge, ki jih ponavadi opravlja JavaScript so sledeče: preverjanje pravilnosti vnesenih podatkov, skrivanje in prikazovanje določenih elementov na spletni strani, od-

piranje in zapiranje novih oken, enostavni izračuni, shranjevanje določenih podatkov v spremenljivke, spreminjanje strukture strani itd. Skripte JavaScript izvaja spletni brskalnik ob različnih dogodkih. Skripte se lahko izvajajo ob nalaganju strani, ob premiku miške, ob kliku na določen element itd. Glavna slabost tega skriptnega jezika je nekonsistentno interpretiranje kode na vseh spletnih brskalnikih. Za podporo več brskalnikov je zato ponavadi potrebno pisati dodatne funkcije, ki opravljajo isto opravilo.

5.4 MySQL

MySQL je sistem za upravljanje z relacijskimi podatkovnimi bazami. Za delo s podatki uporablja povpraševalni jezik SQL. Napisan je v C in C++. Izdan je bil leta 1995. Razvilo ga je švedsko podjetje MySQL AB, ki je sedaj v lasti korporacije Oracle. Deluje na večini današnjih platform (Linux, Microsoft Windows, Mac OS X, Solaris, Open BSD itd.). Trenutno je v uporabi verzija 5.5.22 (zadnja stabilna izdaja - 22. marec 2012). MySQL deluje na principu odjemalec - strežnik. Danes se MySQL najpogosteje uporablja v povezavi s skriptnim programskim jezikom PHP. Osnovne naloge MySQL-a je urejanje podatkov v podatkovni bazi in izvajanje poizvedb nad podatki. Ob povezavi spletne aplikacije z bazo moramo podatkovni bazi sporočiti uporabniško ime, geslo, ime podatkovne baze do katere dostopamo in naslov gostitelja, kjer se baza nahaja. Za administracijo podatkov znotraj podatkovne baze lahko uporabljamo orodja, ki ukaze izvajajo iz ukazne vrstice, kot tudi okenske aplikacije z grafičnimi uporabniškimi vmesniki (npr. MySQL Workbench, MySQL Administrator ...). Zelo priljubljena je tudi odprtokodna (licenca: GNU General Public License) spletna aplikacija PhpMyAdmin. Obstaja cela vrsta različnih sistemov za upravljanje z relacijskimi podatkovnimi bazami (npr. Oracle, IBM DB2, PostgreSQL, Ingres, Microsoft Access, itd). Za MySQL smo se odločili, ker je brezplačna in ta sistem najdemo v programskem paketu XMAPP. MySQL je ponavadi tudi najpogostejša izbira razvijalcev spletnih aplikaciji v povezavi s programskim jezikom PHP.

5.5 PHP

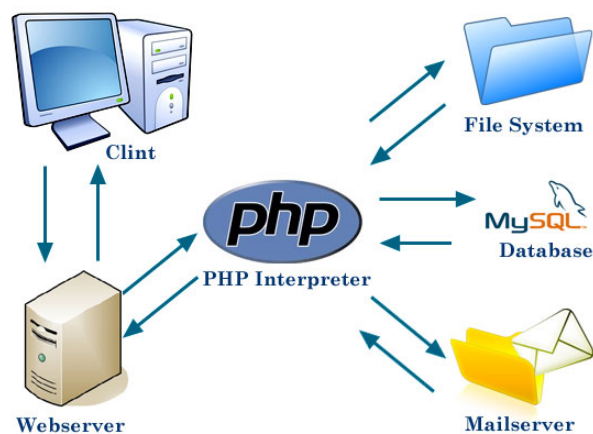
PHP (tričrkovni rekurzivni akronim za PHP Hypertext Processor, izvirno Personal Home Page Tools, (slovensko orodje za osebno spletno stran)) je razširjen odprtokodni skriptni jezik. Primarno je namenjen izdelavi spletnih aplikacij. PHP sistem se izvaja na spletnih strežnikih. Sestavljen iz petih delov (povzeto po viru [1]):

- Thread Safe Resource Management skrbi, da lahko določena instanca sistema istočasno obdeluje več zahtev.
- SAPI (Server API) je zadolžen za kodiranje življenjskega cikla PHP-ja in je običajno vgrajen (angl. embedded) v spletni strežnik. Primer SAPI-ja je Apache HTTP Server (v nadaljevanju “Apache”), ki na določenih vratih (angl. port) posluša in prejete HTTP zahteve po določeni PHP skripti posreduje PHP sistemu.
- PHP Core skrbi za inicializacijo osnovnih funkcij (obdelava datotek, delo z napakami itd). Te funkcije programerjem niso programsko dostopne.
- Zend Engine skrbi za razčlenjevanje in prevajanje PHP skripte v strojno kodo ter njeno izvajanje.
- Extensions vsebuje funkcije, ki so razvijalcem spletnih aplikacij programsko dostopne (npr. delo z bazo, sejami, nizi itd.).

PHP sistem sprejema HTTP zahteve po PHP skriptah ter jih razčleni in interpretira. Poenostavljen primer tega delovanja je prikazan na sliki 5.1². Rezultat tega procesiranja je ponavadi spletna stran (HTML dokument), ki jo spletni strežnik posreduje spletnemu brskalniku, ki je to skripto zahteval.

PHP-jev “oče” je dansko-kanadski programer Rasmus Lerdorf. Napisan je v programskem jeziku C. PHP se je prvič pojavil leta 1995. Trenutno je v

²Slika je dostopna na: <http://www.web-design-wiki.com/images/how-php-works.jpg>



Slika 5.1: Shema delovanja sistema PHP.

rabi verzija PHP 5.4. PHP je dokaj razumljiv in enostaven za učenje. Podpira večino današnjih podatkovnih baz (DB2, Oracle, FilePro, MySQL itd.). Uporabljamo ga lahko na večini operacijskih sistemov (Microsoft Windows, Linux, Mac OS X).

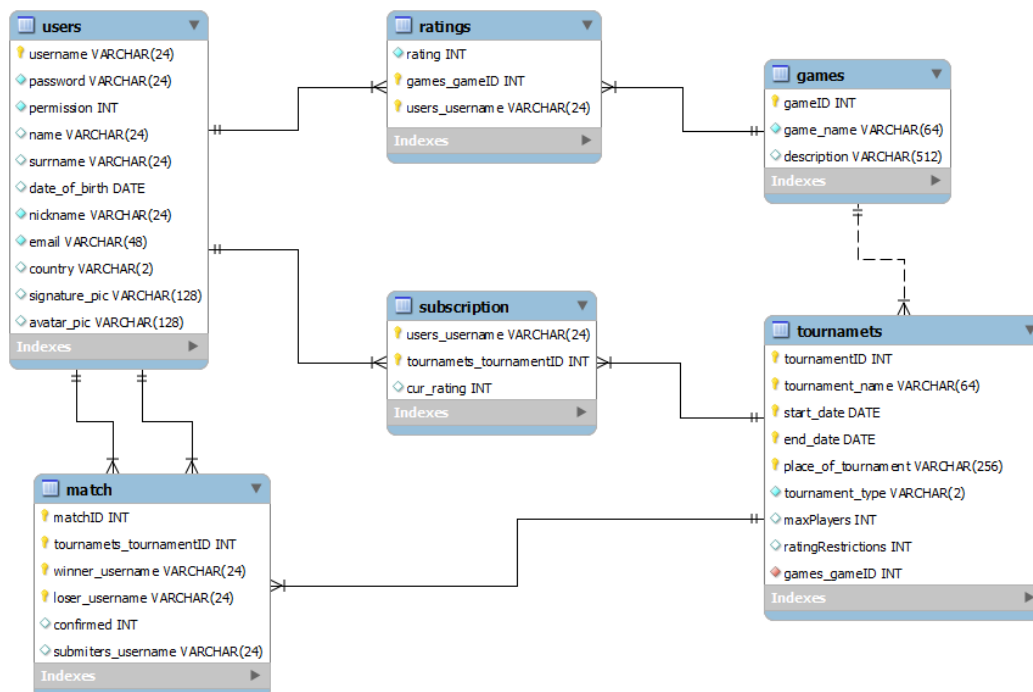
Poglavje 6

Razvoj spletne aplikacije

Glavni cilj diplomskega dela je bil izdelava spletne aplikacije, katere namen je vodenje in pregledovanje raznih turnirjev v različnih športnih disciplinah.

Prvi korak pri izdelavi spletne aplikacije je bil izdelava podatkovnega modela (EER Diagram¹) baze podatkov. Za izdelavo podatkovnega modela smo uporabili orodje MySQL Workbench. Zanj smo se odločili zaradi razumljivosti njegovega uporabniškega vmesnika. V orodju MySQL Workbench podatkovni model izdelamo tako, da na začetni strani izberemo možnost “Create New EER Mode” ali v nadzorni vrstici v zavihku “File” izberemo “New Model”. V oknu, ki se nam odpre izberemo možnost “Add Diagram”. V oknu (“Diagram”) narišemo podatkovni model in vanj vnesemo vse potrebne podatke (atributi tabel, s pripadajočimi tipi in omejitvami, povezave med tabelami itd.). Orodje ponuja tudi funkcijo (Forward Engineer to Database, ki jo najdemo v zavihku Database v orodni vrstici), s katero lahko podatkovni model pretvorimo v bazo podatkov. Workbench torej za nas izdelava vse skripte in z njihovo pomočjo ustvari relacijsko bazo podatkov. Funkcija nam iz podatkovnega modela zgradi tabele z ustreznimi lastnostmi (atributi, primarni in tujimi ključi, povezave med tabelami, omejitve vnosa podatkov

¹ERR diagram je konceptualni model s katerim opišemo strukturo podatkovne baze ali informacijskega sistema. Definicija EER diagrama in postopek izdelave le tega je opisan v viru: [14].



Slika 6.1: Podatkovni model turnirske baze podatkov.

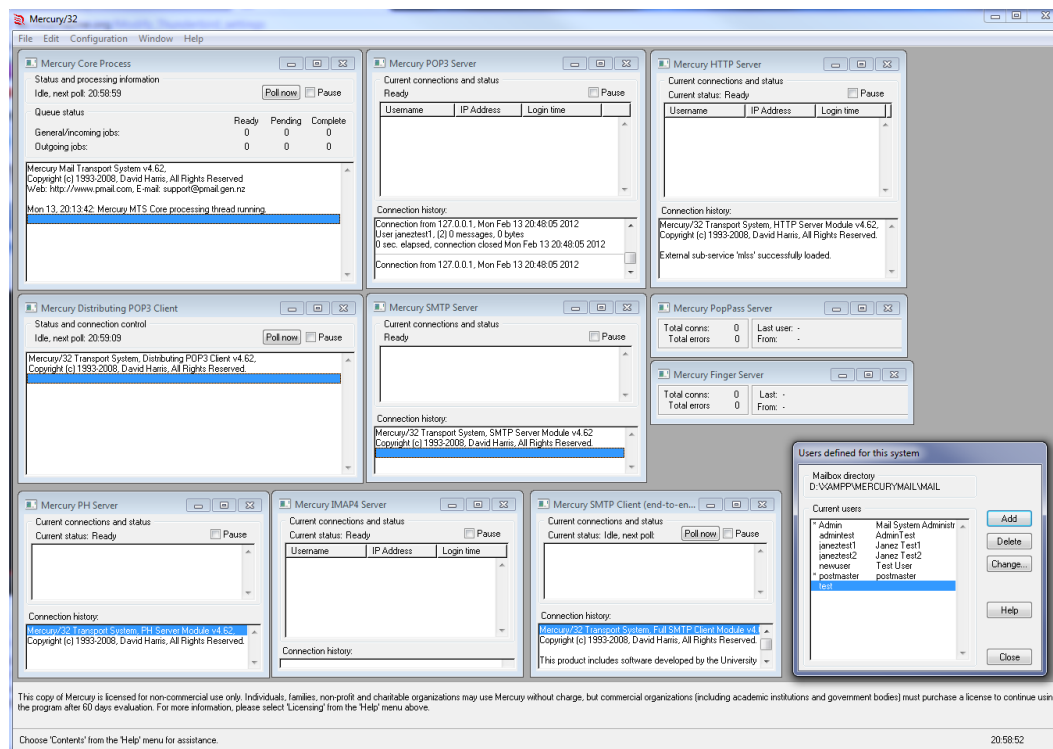
v določenih atributih itd.). To orodje omogoča tudi urejanje podatkov (kreiranje, brisanje in urejanje tabel in podatkov znotraj le teh, izvajanje SQL poizvedb, itd.) znotraj podatkovne baze MySQL, vendar smo za potrebe diplomske naloge te operacije raje izvajali s pomočjo orodja PhpMyAdmin. Slika 6.1 prikazuje podatkovni model iz katerega smo zgradili podatkovno bazo, uporabljeno v naši spletni aplikaciji. To sliko smo naredili, tako, da smo v orodni vrstici v zavihku “File” izbrali možnost “Export”, znotraj te pa možnost “Export as PNG”.

Pri razvoju in testiranju smo uporabljali odprtokodni programski paket XAMPP. Ta programska rešitev je namenjena hitri vzpostavitvi spletišča. Ob njeni namestitvi se nam na naš operacijski sistem namestijo sistemi: spletni strežnik Apache, podatkovna baza MySQL in tolmača za programska jezika PHP in Perl. Iz zgoraj opisanega je razvidno, da je XAMPP akronim sestavljen iz začetnih črk sistemov, ki jih ta programska rešitev vsebuje in črke X,

ki pomeni možnost namestitve programa na več operacijskih sistemih (program lahko namestimo znotraj Windows, Linux in Mac OS X operacijskih sistemov - angl. Cross-platform). Poleg zgoraj opisanih sistemov se nam ob namestitvi XAMPP-a namestijo še drugi sistemi in moduli kot so: strežnik FTP FileZilla, Tomcat, PhpMyAdmin, Mercury MTS, OpenSSL in drugi. Za potrebe razvoja in testiranja spletne aplikacije smo uporabljali le: tolmač skriptnega programskega jezika PHP, Apache, podatkovno bazo MySQL in programska modula Mercury MTS ter PhpMyAdmin.

Modul PhpMyAdmin je odprtokodno (licenca: GNU GPL) orodje namenjeno administraciji podatkovne baze MySQL znotraj spletnega brskalnika. Omogoča opravljanje različnih nalog kot so: urejanje podatkovnih baz, tabel, polj, vrstic, dodeljevanje pravic uporabnikom, izvajanje raznih poizvedb sql itd.

Modul Mercury MTS (Mercury Mail Transmission System) je poštni strežnik. Podpira večino današnjih protokolov za delo s spletno pošto. Pri razvoju in testiranju spletne aplikacije smo ga potrebovali predvsem za testiranje pravilnosti delovanja funkcionalnosti aplikacije, ki so tesno povezane z uporabo spletne pošte. Z njegovo uporabo smo, kreirali nekaj lokalnih navideznih poštnih predalov (poštni predal za spletno pošto). To storimo tako, da v zavihku "Configuration" izberemo možnost "Manage local users". Odpre se nam okno z naslovom "Users defined for this system", kot je vidno na sliki 6.2. V tem oknu kliknemo gumb "Add". Odpre se nam okno v katerega vnesemo podatke (uporabniško ime in geslo sta obvezna podatka) potrebne za kreiranje lokalnega navideznega uporabniškega računa (za poštni predal). Potem zaženemo odjemalec spletne pošte. Za potrebe diplomske naloge smo uporabljali Mozilin Thunderbird. V orodni vrstici izberemo zavihek "Datoteka", znotraj tega izberemo možnost "Novo" in znotraj te "Poštni račun". Program Thunderbird se nam ponavadi prenese v našem materinem jeziku, zato imena zavihkov pišemo v slovenščini. V čarovnik za kreiranje novega poštnega predala vnesemo enake podatke, kot smo jih vnesli v Mercuryju. Pomembno je, da v polje "E-poštni račun" vnesemo enako uporabniško



Slika 6.2: Izgled uporabniškega vmesnika programa Mercury.

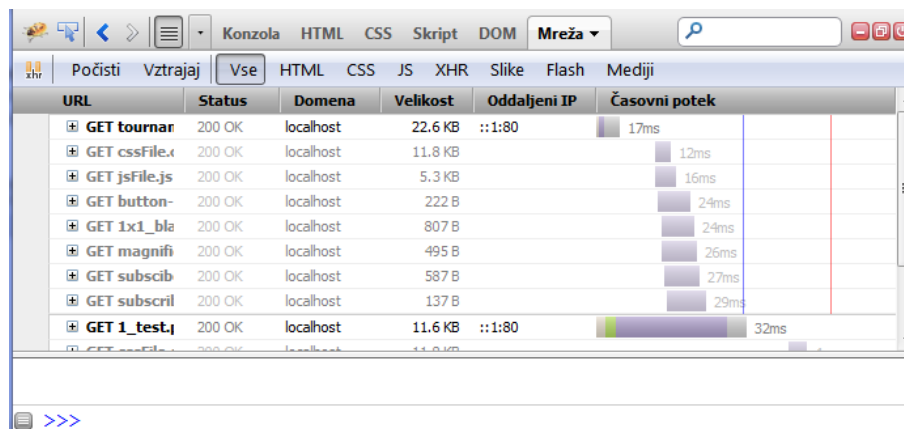
ime kot smo ga vnesli v Mercury-ju, ki dodamo še “@localhost”. Pravilnost delovanja lokalnih navideznih poštnih uporabniških računov preverimo tako, da ustvarimo vsaj 2 poštna predala in v orodju Thunderbird iz enega poštnega računa poskušamo poslati email drugemu navideznemu računu. Če je drugi navidezni račun prejel poslani email, smo pravilno kreirali navidezne poštno predale in lahko pričnemo s testiranjem funkcionalnosti (npr. pošiljanje določenih podatkov na določen email naslov) naše spletne aplikacije. V spletni aplikaciji smo z uporabo Mercury-ja in Thunderbirda testirali pravilnost delovanja funkcije, ki nam ob registraciji novega uporabnika pošlje email na vnešeni email naslov ter funkcionalnost, ki uporabniku nastavi novo geslo in mu le to pošlje na email naslov.

Poleg sistemov vsebovanih v programskem paketu XAMPP, smo pri razvoju uporabljali tudi program NetBeans IDE. Ime NetBeans IDE predsta-

vlja tako programsko orodje (angl. framework) za razvoj namiznih javanskih aplikacij kot tudi integrirano razvojno okolje (angl. IDE), namenjeno razvoju programov in skript napisanih v programskih jezikih: Java, JavaScript, PHP, C, C++ in drugimi. NetBeans IDE je napisan v programskem jeziku Java. Namestimo ga lahko znotraj vseh operacijskih sistemov (Microsoft Windows, Linux, Mac OS, Solaris itd.), ki podpirajo JVM². NetBeans IDE smo izbrali za pisanje spletne aplikacije predvsem zato, ker vsebuje urejevalnik JavaScript (angl. NetBeans JavaScript editor). Ta urejevalnik nam pri pisanju spletnih aplikacij pomaga z barvanjem določenih delov sintakse, avtomatskim zaključevanjem določenih delov kode (npr. HTML značk, delov CSS pravil, ...) in avtomatskim generiranjem povratnih Ajax klicev (angl. Ajax callbacks) iz predloge, avtomatskim preverjanjem izvedljivosti kode v spletnih brskalnikih. Slika 6.4 prikazuje izgled programa NetBeans IDE.

Ko smo zaključili s programiranjem posamezne skripte PHP, smo le to shranili v datoteko s končnico `.php`. Datoteko smo nato shranili znotraj mape, ki predstavlja spletno aplikacijo. To mapo smo predhodno ustvarili znotraj mape "htdocs", ki se kreira ob namestitvi XAMPP-a. Npr. XAMPP smo namestili znotraj operacijskega sistema Windows na lokaciji "D:\xampp". Po končani namestitvi sistema se znotraj te mape ustvari mapa "htdocs". Znotraj te mape (torej: "D:\xampp\htdocs") kreiramo mapo, ki predstavlja ime našega projekta. Med razvojem smo aplikacijo poimenovali "Diploma63080032". V nadaljnem razvoju smo vse skripte PHP shranjevali znotraj te mape (pot za shranjevanje vseh skript v nadaljem razvoju: "D:\xampp\htdocs\Diploma63080032\" in sledečih podmap na tej poti). Pravilnost prikaza in delovanja skript PHP smo testirali tako, da smo odprli različne spletne brskalnike (za potrebe diplomskega dela smo pravilnosti prikaza skript testirali znotraj spletnih brskalnikov Mozilla Firefox in Google Chrome) in v naslovno vrstico vnesli (za skripte, ki se nahajajo na poti iz prejšnjega primera): "http://localhost/Diploma63080032/" in ime skripta

²JVM so javine navidezne naprave sposobne izvajanja datotek Java bytecode (to so datoteke s končnicama `.class` ali `.jar`)

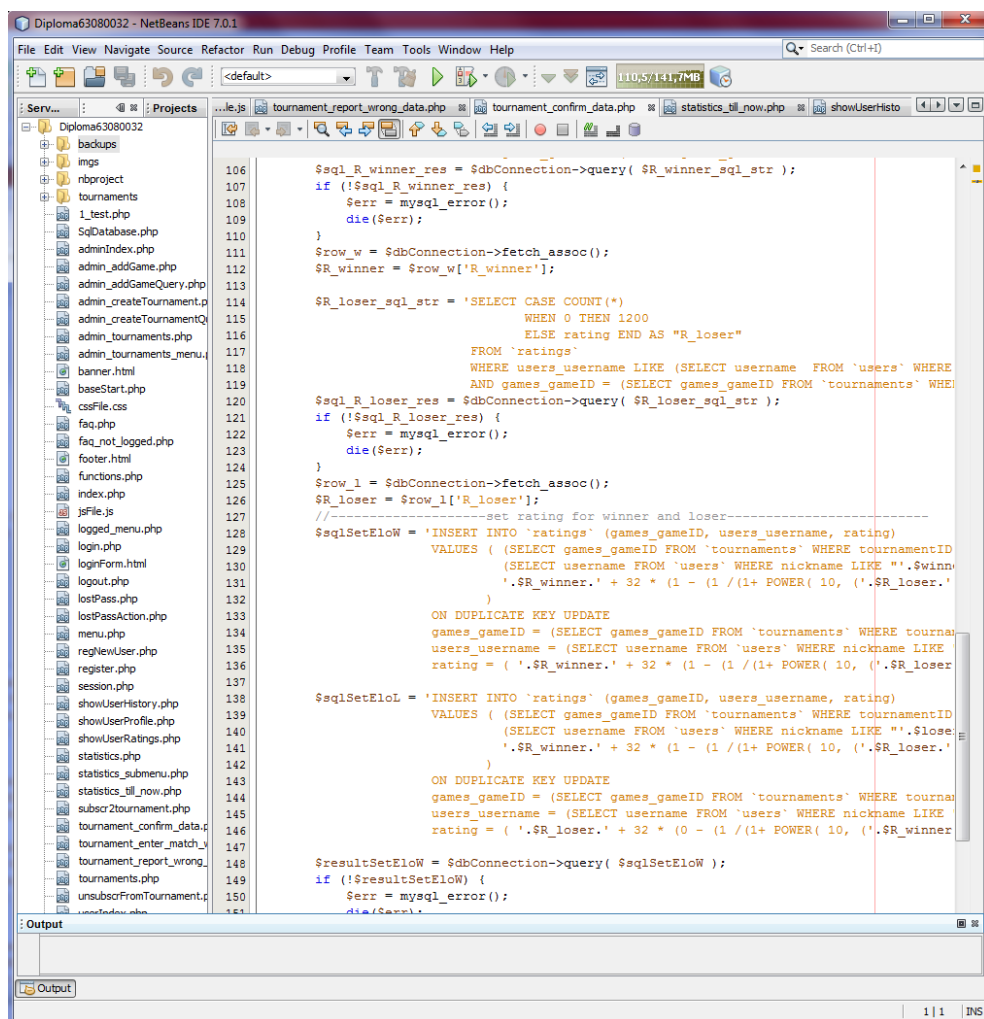


Slika 6.3: Izgled razhroščevalnika Firebug.

PHP, ki smo ga želeli testirati (npr. za stran, ki se nam prva pojavi ob obisku spletne strani je URL sledeč: “<http://localhost/Diploma63080032/index.php>”). Znotraj posameznega brskalnika smo pravilnosti prikaza posameznega skripta PHP preverili z uporabo razhroščevalnikov. V spletnem brskalniku Firefox smo namestili vtičnik Firebug³, katerega uporabniški vmesnik je viden na sliki 6.3, brskalnik Google Chrome pa ima razhroščevalnik že integriran. V obeh spletnih brskalnikih razhroščevalnik prikažemo s pritiskom tipke F12. Razhroščevalnika nam omogočata razhroščevanje: kode HTML generirane na strežniku, kode CSS in JavaScript, sestavo objekta DOM⁴ in zahtev ter odgovorov HTTP (angl. HTTP request, respons).

³Firebug. Dostopno na: <http://getfirebug.com/>

⁴Document Object Model je platformno in jezikovno nevtralen vmesnik (angl. interface), ki omogoča programom in skriptam dinamično dostopanje in spreminjanje vsebine, stila in strukture dokumenta (HTML, XHTML ali XML). S pomočjo tega vmesnika je dokument lahko nadaljno procesiran, rezultati tega procesiranja se lahko nadaljno vključijo v predstavitev trenutnega dokumenta (npr. spletne strani).



Slika 6.4: Izgled orodja NetBeans IDE.

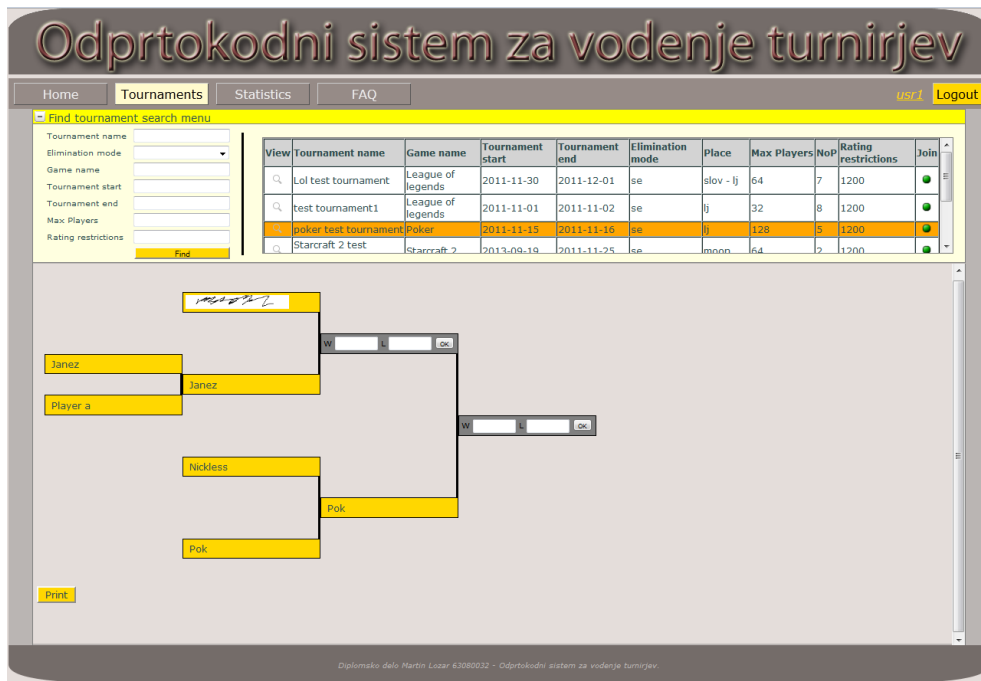
Poglavje 7

Opis funkcionalnosti spletne aplikacije za vodenje turnirjev

Kot pri večini današnjih spletnih aplikacij se je potrebno tudi za uporabo razvite spletne aplikacije najprej registrirati. Za registracijo klikne nov uporabnik na začetni strani (index.php) na povezavo “Register”. Odpre se mu okno, v katerega uporabnik vnese lastne podatke. To so ime, priimek, email naslov (s katerim se ni registril še noben uporabnik), uporabniško ime in 2-krat enako geslo. Nato uporabnik klikne na gumb “Register”. Če uporabnik ni vnesel pravih ali vseh podatkov, ga aplikacija o tem obvesti. Ob pravilnem vnosu podatkov aplikacija uporabnika obvesti o uspešno izvedenem opraviilu in mu na vnešen email naslov pošlje sporočilo z dobrodošlico.

Če registrirani uporabnik pozabi svoje geslo lahko to ponastavi tako, da na prvi strani izbere “Forgot password”. Odpre se mu okno s sledečimi vnosnimi polji: ime, priimek, uporabniško ime ter email naslov. Obvezno mora vnesti prvi dve in vsaj eno izmed zadnjih dveh polj. Če se vnešeni podatki ujemajo s tistimi v podatkovni bazi, aplikacija generira novo geslo naključne dolžine (med 5 in 24 znaki), sestavljeno iz malih črk angleške abecede in številčk med 0 in 9. To geslo shrani v uporabnikov profil in ga pošlje uporabniku na vnešen email naslov.

Po uspešni prijavi se uporabniku prikaže stran (userIndex.php), na kateri



Slika 7.1: Izgled strani, ki se prijavljenemu uporabniku prikaže ob kliku na gumb “Tournaments”.

lahko uporabnik v orodni vrstici izbira med gumbi: Home, Tournaments, Statistics, ogledom lastnega profila in Logout. Uporabniku se ob kliku na gumb “Tournaments” odpre okno (tournaments.php - izgled te strani je prikazan na sliki 7.1) znotraj katerega je iskalni menu in seznam vseh vnešenih turnirjev (že odigranih in prihajajočih).

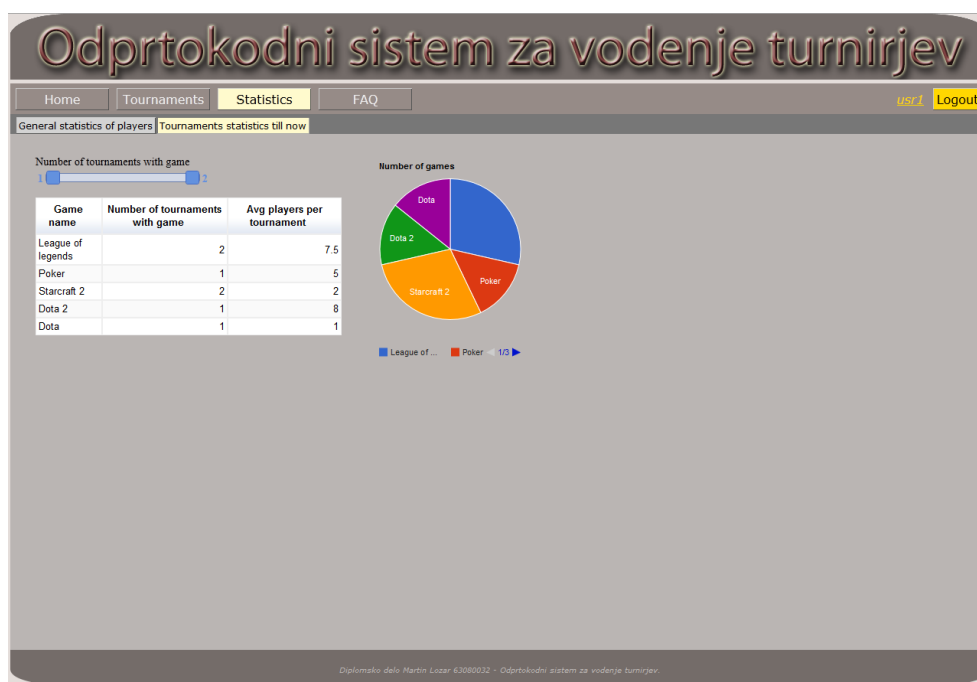
Uporabnik lahko ta izpis z uporabo iskalnega menija nekoliko zmanjša in prilagodi lastnim željam in potrebam. Prijavi se lahko na vse turnirje, ki se še niso začeli in za katere izpolnjuje vstopne pogoje (ali je njegov rating pri igri, ki se bo na turnirju igrala enak ali višji kot je omejitev napisana v stolpcu Rating restrictions) ter na turnir, ki še ni zapolnjen. Če je turnir že polno zaseden, se v vrstici, ki opisuje ta turnir v zadnjem stolpcu izriše rdeč krog. V primeru, ko je uporabnik na določen turnir prijavljen, se v tem stolpcu turnirja izriše zelen krog. Če pa uporabnik še ni prijavljen

na turnir, se v vrstici z izbranega turnirja izriše zelen križec. Uporabnik lahko pregleduje poljuben turnir. To stori s klikom na ikono lupe v prvem stolpcu. V primeru, ko želi uporabnik pregledovati turnir na večji površini lahko s klikom na (minus poleg napisa "Find tournament search menu"). Ob kliku ikone se uporabniku iskalni menu skrije, ikona pa se spremeni v križec. Ob ponovnem kliku na to ikono se uporabniku iskalni menu spet prikaže. Aplikacija ponuja uporabniku tudi tiskanje posameznega turnirja (gumb "Print" pod drevesom, ki predstavlja potek turnirja). V primeru, ko določen turnir še poteka, lahko delne rezultate vnaša administrator strani ali pa uporabnika (tekmovalca), ki sta določeno tekmo odigrala. V primeru, ko rezultat vnese administrator, je rezultat takoj veljaven, tekmovalcema, ki sta to tekmo odigrala se takoj izračuna in popravi rating pri igri, ki se igra na turnirju. V primeru, ko uporabnik sam vnese svoj rezultat (lahko vnaša samo v polja, tekem na katerih je sam igral), mora le ta rezultat potrditi ali ovreči uporabnik proti kateremu je igral. Če se vnosa rezultatov teh dveh tekmovalcev ujemata, je vnešeni rezultat potrjen in šele po tem se uporabnikoma izračuna nova ELO relativna moč. Uporabnik potrди rezultat tako, da v polju, ki predstavlja pravkar odigrani dvoboj klikne na gumb z napisom "Y". Rezultat pa razveljavi tako, da klikne na gumb X. Če rezultat še ni bil vnešen, se v polju, ki predstavlja pravkar odigrano tekmo, pojavita vnosni polji, v kateri je potrebno vnesti vzdevek (nickname) zmagovalca in poraženca tekme. Da bi aplikacija nekoliko varčevala s sistemskimi viri, se uporabnikom, ki niso prijavljeni na določen turnir, a ga želijo pregledovati posreduje HTML dokument z zadnjim potrjenim stanjem. To stanje se posodobi vsakič, ko administrator ali tekmovalec, ki sodeluje na tem turnirju vnese nove podatke. Administratorju in uporabnikom, ki pregledujejo turnir na katerega so prijavljeni, se stran vsakič generira s podatki, ki so trenutno v bazi (tudi s še ne potrjenimi rezultati). Administrator lahko spreminja tudi že vnešene rezultate. To stori tako, da v vnosna polja pod prikazanim turnijem vnese zaporedno številko igre (matchid) ter vzdevek zmagovalca in poraženca. Uporabnik lahko zaporedno številko tekme izve tako, da po-

stavi miško znotraj polja, ki predstavlja želeno tekmo (izven slike - podpisa tekmovalca ali izven napisa njegovega vzdevka). Uporabnik lahko tudi izve tekmovalčev vzdevek tako, da postavi miškin kazalec na njegov podpis. Ta funkcionalnost uporabnikom pomaga razbrati tekmovalčev vzdevek takrat, ko je ta uporabnik naložil podpis (sliko), ki ne izdaja njegovega vzdevka. Če uporabnik postavi miškin kazalec na določenega tekmovalca, se mu vsa polja v katerih se nahaja ta tekmovalec obarvajo (s svetlo modro barvo v ozadju polja). To omogoča lažje pregledovanje uvrstitev posameznih tekmovalcev.

Prijavljen uporabnik si lahko ogleda tudi dva statistična prikaza. Ob kliku na gumb “Statistics” se uporabniku prikaže stran (statistics.php) na kateri se nahaja tortni graf in tabela v katerih so prikazani podatki o tekmovalcih, ki uporabljajo aplikacijo. V tabeli so prikazani sledeči podatki uporabnikov: ime, priimek, starost, država bivanja in število zmag na vseh turnirjih za vsakega uporabnika posebej. S klikom na naslov vsakega stolpca uporabnik interaktivno uredi podatke (padajoče ali naraščajoče). Nad tabelo se nahaja drsnik s katerim uporabnik omeji količino podatkov prikazanih v tabeli. Z uporabo tega drsnika uporabnik nastavi interval starosti (minimalna in maksimalna starost) tekmovalcev, katerih podatke želi pregledovati. Poleg tabele se nahaja še tortni graf, ki prikazuje število zmag vsakega tekmovalca, ki se nahaja v tabeli (drsnik, ki določa interval starosti dinamično spreminja tudi tortni graf). Na desni strani grafa je legenda, v kateri je predstavljena barva, ki predstavlja določenega tekmovalca.

Na zgoraj omenjeni strani se nahaja še en podmeni na katerem lahko uporabnik preklaplja med sedanjo stranjo (klik na gumb z napisom “General statistics of players”) in stranjo (klik na gumb z napisom: “Tournaments statistics till now”), ki prikazuje statistične podatke o vseh do sedaj odigranih turnirjih in igrah, ki so se na teh turnirjih igrale. Izgled slednje je prikazan na sliki 7.2. Notranji del strani je zgrajen na enakem principu kot prej opisana stran (statistics.php). Na strani se nahaja tudi interaktivna tabela znotraj katere so prikazani trije stolpci (ime igre, število turnirjev na katerih se je igrala določena igra in povprečno število sodelujočih na vseh



Slika 7.2: Izgled dela spletne aplikacije, ki prikazuje statistiko turnirjev.

turnirjih z določeno igro). Nad tabelo je drsник, s katerim lahko uporabnik omeji število vseh turnirjev na katerih se je igrala določena igra (nastavitev prikaza intervala za drugi stolpec tabele). Vsebina tabele je tudi na tej strani predstavljena v tortnem grafu z legendo.

Aplikacija uporabniku omogoča tudi urejanje lastnega profila, pregledovanje ratingov pri posamezni igri in pregledovanje lastne zgodovine odigranih tekem na vseh turnirjih na katerih je sodeloval. Do podatkov svojega profila dostopa tako, da klikne na povezavo s svojim vzdevkom (desno od gumba "Logout"). Ob kliku te povezave se uporabniku prikaže stran s podmenijem s tremi gumbi ("My profile", ki je tudi privzeta vrednost ob kliku na lasten vzdevek, "My ratings" in "History"). Na zavihku "My profile" se uporabniku prikažejo njegovi podatki in sliki avatarja ter podpisa (če je le ti pred tem naložil na strežnik). Uporabnik svoje podatke ureja tako, da klikne na ikono križca poleg vrstice, ki predstavlja podatek, ki ga želi uporabnik spremeniti.



Tournament name	Winner	Looser	Game
test tournament1	Nickless	u7ni	League of legends
poker test tournament	Pok	Nickless	Poker
Starcraft 2 test tournament	Nickless	misterJ	Starcraft 2
Lol test tournament	misterJ	Nickless	League of legends

Slika 7.3: Izgled dela spletne aplikacije, ki prikazuje tekmovalčevo zgodovino odigranih dvobojev na vseh turnirjih.

Ko uporabnik vnese vse zelene podatke, pritisne tipko “Save changes”. Aplikacija uporabnika obvesti bodisi ob uspešni spremembi podatkov, bodisi o napaki, ki jo je uporabnik zagrešil ob vnosu (npr. prevelika slika za podpis ali avatar, izbrana datoteka ni slika, datum rojstva je neveljaven, itd.).

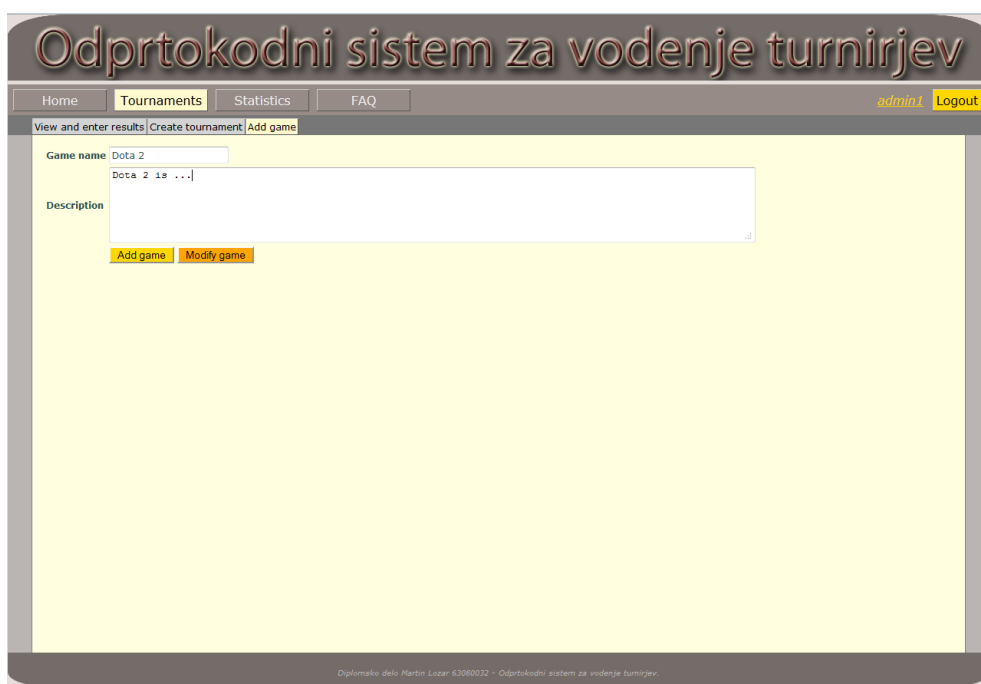
Naslednji gumb v podmeniju (“My ratings”) uporabniku prikaže tabelo v kateri so zbrane vse igre, ki so se igrale na turnirjih na katerih je prijavljeni uporabnik sodeloval. Za vsako igro je izpisana tudi trenutna tekmovalčeva relativna moč igrane igre. Za lažje branje podatkov sta dve sosednji vrstici poobarvani z nekoliko drugačno (temnejšo ali svetlejšo) barvo (t.i. zebra efekt). Zadnji gumb v tem podmeniju (“History” - izgled strani je viden na sliki 7.3) prikaže uporabniku celotno njegovo zgodovino dvobojev v tabeli. Prvi stolpec tabele predstavlja ime turnirja na katerem je bil dvoboj odigran. V naslednjih dveh stolpcih sta zapisana vzdevka zmagovalca in poraženca dvoboja. V zadnjem stolpcu pa je zapisano ime igre, ki se je igrala na turnirju. Vrstica v kateri je logirani uporabnik (njegov vzdevek) zapisan v stolpcu zmagovalca (“Winner”), ima ozadje obarvano svetlo zelene barve, vrstica v kateri pa je uporabnikov vzdevek napisan v stolpcu za poraženca (“Loser”) je obarvana rdeče.

Administratorju so znotraj aplikacije dostopne vse storitve do katerih lahko dostopa vsak prijavljen uporabnik. Razlika med dvema vrstama upo-

rabniških računov se pojavi le znotraj strani, na kateri se prikazujejo turnirji (ob kliku na gumb "Tournaments"). Ob obisku te strani se uporabniku pod gumbom "Tournaments" prikaže še podmeni z naslednjimi tremi gumbi: "View and enter results", "Create tournament" in "Add game". Znotraj strani, ki se nam prikaže ob kliku prvega gumba v podmeniju ("View and enter results"), lahko administrator pregleduje in vnaša rezultate posamezne tekme na turnirju, ki ga pravkar pregleduje.

Ob kliku gumba "Create tournament" se administratorju prikaže vnosna forma (več vnosnih polj) s pomočjo katere administrator vnese podatke za ime turnirja (maksimalna dolžina je 64 znakov), ime igre, ki se bo na turnirju igrala, kraj dogajanja turnirja (največ 256 znakov), začetni in končni datum turnirja, maksimalno število igralcev ter neobvezen atribut omejitev ratinga (relativne moči).

Zadnji gumb v podmeniju ("Add game" - stran, ki se nam prikaže ob kliku na ta gumb je vidna na sliki 7.4) omogoča administratorju dodajanje novih iger, ki se lahko igrajo na nekem turnirju (ob kreiranju novega turnirja administrator lahko izbira zgolj med igrami, ki jih je predhodno vnesel s pomočjo prej opisanega obrazca). Administrator na tej strani vnese ime igre (ki ne sme biti daljše od 64 znakov) in kratek opis te igre (maksimalno 512 znakov). Po vnosu zahtevanih podatkov administrator za vnos le teh pritisne gumb "Add game". S tem obrazcem lahko administrator tudi spreminja že obstoječe opise iger, ki jih je pred tem dodal v bazo podatkov. To stori tako, da v vnosno polje poleg (desno od) napisa "Game name" vnese ime obstoječe igre. V tekstovno polje poleg napisa "Description" vnese nov opis igre in pritisne gumb "Modify game".



Slika 7.4: Izgled dela spletne aplikacije, ki omogoča vnos novega tipa igre oziroma spreminjanje opisa že obstoječe igre.

Poglavje 8

Zaključek

Glavna cilja diplomskega dela sta bila pregled obstoječih aplikacij namenjenih vodenju turnirjev in prikazu sprotnega stanja na njih ter razvoj spletne aplikacije, ki bi to omogočala. Pregled obstoječih aplikacij namenjenih vodenju in vizualizaciji poteka turnirjev smo predstavili v drugem poglavju. Uspelo nam je tudi razviti spletno aplikacijo, ki uporabnikom omogoča pregled vseh prihajajočih in že odigranih turnirjev (ki so bili vodeni ali kreirani s pomočjo razvite spletne aplikacije). Spletna aplikacija trenutno omogoča le kreiranje turnirjev na katerih se zmagovalca določi z uporabo pokalnega sistema. Za vsako različno igro (npr. poker, košarka ..), ki jo je posamezen igralec odigral na določenem turnirju ali več turnirjih, se hranijo podatki o njegovi trenutni kvaliteti v tej igri (kvaliteto igralca pri posamezni igri računamo s pomočjo sistema ELO). Igralec lahko tudi pregleduje svojo zgodovino odigranih tekem na vseh turnirjih na katerih je bil oziroma je še udeležen. Da bi administratorju aplikacije nekoliko olajšali delo smo uporabnikom, ki sodelujejo na določenem turnirju ponudili možnost lastnoročnega vnosa rezultatov. Uporabnik lahko sam vnaša lastne trenutne rezultate (rezultat pravkar odigranega dvoboja na katerem je sam tekmoval). Ti rezultati postanejo veljavni, če oz. ko tudi njegov nasprotnik (v pravkar odigranem dvoboju) potrdi vnešen rezultat. Administratorjev vnos določenega rezultata je veljaven takoj po vnosu. Vrednost razvite funkcionalnosti pride do izraza

na večjih turnirjih na katerih sodeluje večje število ljudi, saj administratorju tako ni potrebno vnesti vsakega sprotnega rezultata. Za izdelavo diplomskega dela s to tematiko smo se odločili, ker smo želeli izpopolniti znanje s področja spletnega programiranja.

Za izboljšavo uporabniške izkušnje bi lahko v prihodnosti razvili še nekatere popravke spletne aplikacije. Eden od najpomembnejših in prvih popravkov bi bil, da bi aplikaciji dodali možnost kreiranja turnirjev, na katerih je uporabljen še kakšen drug sistem za določitev zmagovalca (npr. pokalni sistem, kjer tekmovalec izpade po dveh porazih - angl. Double elimination, krožni sistem - angl. Round robin) in vizualizacijo le teh. Drugi popravek bi omogočal vodenje evidenc o igralčevih relativnih močeh v različni igratih v različnih rangirnih sistemih (sistemi opisani v poglavju "Sistemi za ocenjevanje kvalitete tekmovalcev"). Poleg tega bi administrator lahko izbral rangirni sistem s pomočjo katerega bi se določile dvojice (na podlagi razvrstitve igralcev glede na njihov trenutni rating pri izbranem rangirnem sistemu v določeni igri), ki se bodo igrale v prvem kolu turnirja. Seveda bi se po odigrani igri igralcema izračunali in vnesli rezultati za vsak rangirni sistem posebej (ne samo za ELO rangirni sistem kot je to realizirano v obstoječi aplikaciji). Sledil bi popravek, ki bi uporabnikom spletne aplikacije omogočil komunikacijo med njimi. To bi lahko realizirali z uporabo zasebnih sporočil ali z uvedbo kakšnega odprtokodnega sistema, namenjenega upravljanja s forumi (npr. phpBB). Nadaljna dopolnitev, bi uporabnikom omogočala prijavo v spletno aplikacijo z uporabo Facebookovega uporabniškega računa.

Seznam slik

2.1	Primer izgleda turnirja na katerem se uporablja krožni razporeditveni sistem. Slika je generirana s spletno aplikacijo Challenge, ki je natančneje opisana v poglavju 3.3.	7
3.1	Grafični vmesnik aplikacije Tournament bracket builder z vnešenimi nastavitvami, ki generirajo prikazno drevo turnirja na sliki 3.2.	10
3.2	PDT generirano z aplikacijo Tournament bracket builder.	11
3.3	Primer turnirja na katerem se uporablja različica pokalnega sistema z dvojnimi izpadanjem (angl. Double elimination).	12
3.4	Grafični vmesnik aplikacije Bracket maker namenjen vnosu podatkov določenega turnirja, ki uporablja pokalni turnirski sistem (a) in izgleda PDT, ki je generiran s spletno aplikacijo Bracket builder po zaključenem turnirju (b).	14
3.5	Challenge-jev urejevalnik prikaza turnirskega drevesa (a) in primer prikaza drevesa uvrstitev na testnem turnirju generiranem s spletno aplikacijo Challenge (b).	16
5.1	Shema delovanja sistema PHP.	29
6.1	Podatkovni model turnirske baze podatkov.	32
6.2	Izgled uporabniškega vmesnika programa Mercury.	34
6.3	Izgled razhroščevalnika Firebug.	36
6.4	Izgled orodja NetBeans IDE.	37

7.1	Izgled strani, ki se prijavljenemu uporabniku prikaže ob kliku na gumb “Tournaments”	40
7.2	Izgled dela spletne aplikacije, ki prikazuje statistiko turnirjev.	43
7.3	Izgled dela spletne aplikacije, ki prikazuje tekmovalčevo zgodovino odigranih dvobojev na vseh turnirjih.	44
7.4	Izgled dela spletne aplikacije, ki omogoča vnos novega tipa igre oziroma spreminjanje opisa že obstoječe igre.	46

Tabele

2.1 Primer Bergerjeve razporeditvene tabele za krožni turnir s 5 ali 6 tekmovalci.	5
---	---

Literatura

- [1] S. Golemon, *Extending and Embedding PHP*, Sams, Pogl. 1, 2006.
- [2] Document Object Model (DOM) (2012). Dostopno na:
<http://www.w3.org/DOM/>
- [3] Elo rating system (2012). Dostopno na:
http://en.wikipedia.org/wiki/Elo_rating_system
- [4] Example of the Glicko-2 system (2012). Dostopno na:
<http://www.glicko.net/glicko/glicko2.pdf>
- [5] Glicko rating system (2012). Dostopno na:
http://en.wikipedia.org/wiki/Glicko_rating_system
- [6] Group stage (2012). Dostopno na:
http://en.wikipedia.org/wiki/Group_stage
- [7] Group tournament ranking system (2012). Dostopno na:
http://en.wikipedia.org/wiki/Group_tournament_ranking_system
- [8] HTML & CSS (2012). Dostopno na:
<http://www.w3.org/standards/webdesign/htmlcss>
- [9] Javascript web APIs (2012). Dostopno na:
<http://www.w3.org/standards/webdesign/script>
- [10] PhpMyAdmin (2012). Dostopno na:
http://www.phpmyadmin.net/home_page/index.php

- [11] Round-robin tournament (2012). Dostopno na:
http://en.wikipedia.org/wiki/Round-robin_tournament

- [12] Single Elimination Tournament Introduction (2012). Dostopno na:
<http://www.indiana.edu/~324/r324singleelimintro.htm>

- [13] Swiss system (2012). Dostopno na:
http://www.petanque.org/downloads/files/swiss_system_who_plays_whom_sheets.pdf

- [14] IV. The (Extended) Entity-Relationship Model (2012). Dostopno na:
www.cs.toronto.edu/~jm/2507S/Notes04/EER.pdf

- [15] The Glicko system (2012). Dostopno na:
<http://www.glicko.net/glicko/glicko.pdf>

- [16] TrueSkill (2012). Dostopno na:
<http://en.wikipedia.org/wiki/TrueSkill>

- [17] TrueSkillTM: A Bayesian Skill Rating System (2012). Dostopno na:
http://research.microsoft.com/pubs/67956/NIPS2006_0688.pdf