# DIZAJN I IZRADA SUSTAVA ZA POREDAK PAROVA I POJEDINACA U TIMSKIM SPORTOVIMA

**Britvar, Antonio**

**Undergraduate thesis / Završni rad**

**2019**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Algebra University College / Visoko učilište Algebra**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:225:075649

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-03-29**

*Repository / Repozitorij:*

Algebra Univerity College - Repository of Algebra Univerity College

**VISOKO UČILIŠTE ALGEBRA**

ZAVRŠNI RAD

# DIZAJN I IZRADA SUSTAVA ZA POREDAK PAROVA I POJEDINACA U TIMSKIM SPORTOVIMA

Antonio Britvar

Zagreb, rujan 2019.

*Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada.*

*U Zagrebu, 18. rujna 2019.*

# Predgovor

Iako su samostalnost i upornost preduvjeti za uspješan akademski život, mnogi su ljudi od velikog utjecaja i značaja za studenta tijekom tog cjelokupnog procesa i više no što su svjesni. Bez njih moje studiranje, kao ni njegov završetak ne bi bili mogući i stoga koristim ovu priliku kako bih posebne zahvale uputio

svojim roditeljima, na beskrajnoj podršci, potpori i strpljenju,

cijeloj obitelji i svim svojim prijateljima, na razumijevanju i osloncu,

Franki Grubišić, na usmjeravanju ka stalnoj potrazi za većim smislom,

Bruni Kumeru, na isticanju važnosti izgradnje i pripovijedanja dobrih priča,

Karmen Murn, Nilsu Leichsenringu, Nilsu Rexhäuseru, Darjanu Bunti, Mariji Ratković Vidaković i Sanji Zanki Pejić na energiji, povjerenju i suradnji,

Predragu Šuki, mentoru, iskrenom uzoru s pravim poticajima i savjetima,

Sonji Stahor Gašpar, za entuzijazam prema struci kojeg je s lakoćom prenijela,

Heleni Bukovac, za posvećenost osobnom i profesionalnom razvoju studenata,

Luki Baranoviću, za svaku priliku, kritiku i pohvalu te povjerenje,

Visokom učilištu Algebra, na prilici, financijskoj podršci i prepoznavanju mojih uspjeha kroz aktivnosti koje su prethodile onim akademskim te

svim svojim kolegama i svim profesorima, zaposlenicima i stručnim suradnicima Algebre koji su u meni još više potaknuli interes za raznim područjima multimedije, kao i svim dosadašnjim profesorima i mentorima od kojih sam imao čast učiti prilikom osnovnoškolskog i srednjoškolskog obrazovanja, a koji su me usmjeravali upravo prema ovom trenutku.

Za kraj, iznimno sam zahvalan korisnicima ove aplikacije na suradnji kroz testiranje i korištenje, povratnim informacijama i usmjeravanju na njihove želje i potrebe.

**Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi**

# Sažetak

Na temelju postojećih i zastarjelih algoritama za rangiranje pojedinaca razrađen je novi algoritam za definiranje poretka parova i pojedinaca u timskim sportovima. Definiranjem strategije i opsega projekta, razvijene su funkcionalne specifikacije i analizirani takvi algoritmi čiji će temeljni principi biti od značaja za izradu novog algoritma. Opisan je tijek izvođenja i način pripreme prototipa aplikacije za primjenu. Uz to, osmišljen je, dizajniran i izrađen vizualni prototip te konačni proizvod platforme u obliku web aplikacije u kojoj grupe sportaša mogu pri igranju utakmica digitalno bilježiti svoje rezultate natjecanja. Nakon registracije u aplikaciji i unošenja ishoda utakmica s ostalim igračima, kroz algoritam se dinamički stvaraju, bilježe i prikazuju grupni i pojedinačni rezultati i bodovi. Na taj način nastoji se postići preciznije određivanje ishoda timova i pojedinaca i utjecaj tih ishoda na njihove pozicije na rang listi s minimalnim ulaznim parametrima o utakmicama. Posebnost ovog sustava jest činjenica da algoritam ima realan utjecaj na formiranje pozicija timova i pojedinaca, čak i kada nisu svi parovi međusobno povezani utakmicama. Ideja jest da se takva preciznost postiže učestalom evaluacijom bodova protivnika te vlastitih bodova kroz vrijeme, uz dodatak da se biranje partnera i protivnika prema dosadašnjim rezultatima odražava i na igračev vlastiti rezultat. Ovakav sustav mnogo znači igračima koji se povremeno bave sportom i žele kvalitetnije organizirati vlastito vrijeme te pritom detaljnije pratiti svoj napredak. Glavni izazov jest da ovakav algoritam djeluje linearno, odnosno da je vremenski poredak od iznimnog značaja za oblikovanje rang liste, kao i provjera ispravnosti i istinitosti unesenih podataka. Najzanimljivija značajka koju ova aplikacija donosi pored statističkih prikaza o napretku jest predlaganje odabira partnera prema vještinama i rezultatima kako bi turnirska struktura bila ujednačenija. U konačnici, ovakvo sortiranje može biti iskorišteno i u drugim sportovima, ali i područjima gdje ovakva vrsta poretka može biti od značaja pri procjeni i rangiranju.

**Ključne riječi:** algoritam za poredak, web aplikacija, timski sportovi, sportska platforma, parovi i pojedinci, procjena vještina

# Abstract

Based on existing and mostly outdated algorithms for ranking individuals, a new algorithm is developed to define the ranking of teams and individuals in team sports. By defining the strategy and the scope of the project, functional specifications have been developed and relevant algorithms are analysed, forming the fundamental principles for the design of the new algorithm. Its flow is described and charted, along with the steps of preparing the prototype of an algorithm testing application. In addition, a visual prototype of the user application is designed and built, which leads to the final product of the sports platform in the form of a web application in which groups of athletes can digitally record the results of their matches. After completing the registration process and entering outcomes of matches with other players, the algorithm dynamically generates, records and displays the scores of both teams and individuals. In this way, an effort is made to more accurately determine the outcomes of teams and individuals and the impact of those outcomes on their ranking positions with minimal match input parameters. The peculiarity of this system is the fact that the algorithm has a realistic and up-to-date impact on the formation of the ranking positions of teams and individuals, despite the fact that not all pairs and players are interconnected by matches. The idea is that such precision is achieved by frequent evaluation of opponents' points and the player's own points over time, reflecting the differences between them and impacting their scores. This kind of system means a lot to players who occasionally play sports and want a better way of organizing their own time and tracking their progress. The main challenge is that such an algorithm works linearly, that is, the time order is extremely important for the creation of ranking list, as well as checking the validity and veracity of the entered data. Aside from the dynamic progress statistics, the most interesting feature of this application is the suggestion and the selection of best partners according to an accurate estimate of skills by scores, making the tournament structure more uniform. Ultimately, this system could be used in other sports, as well as areas where this type of ranking can have an impact in evaluation.

**Key words:** ranking algorithm, web application, team sports, sports platform, teams and individuals, skills assessment

# Sadržaj


1. Uvod ..... 1
2. Poredak parova i pojedinaca ..... 2
   2.1. Strategija i opseg ..... 2
      2.1.1. Ciljevi projekta i potrebe korisnika ..... 3
      2.1.2. Funkcionalne specifikacije i zahtjevi za sadržajem ..... 9
   2.2. Postojeći algoritmi za stvaranje poretka ..... 13
   2.3. Problematika i korištene tehnologije ..... 15
3. Razrada algoritma ..... 16
   3.1. Tijek izvođenja ..... 17
      3.1.1. Unos parova ..... 18
      3.1.2. Sortiranje parova ..... 23
      3.1.3. Sortiranje pojedinačnih igrača ..... 29
   3.2. Priprema prototipa za primjenu ..... 31
      3.2.1. Konačna struktura modela ..... 31
      3.2.2. Funkcije sinteze i prikaza podataka ..... 33
      3.2.3. Masovni unos i brzina izvođenja ..... 34
4. Dizajn aplikacije ..... 39
   4.1. Istraživanje, prikupljanje informacija i potreba korisnika ..... 39
      4.1.1. Istraživanje i pretpostavke ..... 39
      4.1.2. Rezultati i odluke ..... 40
   4.2. Wireframe, workflow i korisničko sučelje ..... 43
   4.3. Razvoj vizualnog sadržaja i izgleda ..... 48
   4.4. Prijedlog izgleda sučelja i analiza funkcionalnosti ..... 49

4.4.1. Polazni prijedlog ..... 49
4.4.2. Testiranje upotrebljivosti prema polaznom prijedlogu ..... 50
4.4.3. Prijedlog nakon unapređenja ..... 52
5. Razvoj aplikacije ..... 57
5.1. Sustav registracije, prijave i pohrane podataka ..... 58
5.2. Primjena algoritma u web aplikaciji ..... 61
5.3. Dodatne mogućnosti, primjene i skaliranje ..... 64
Zaključak ..... 67
Popis kratica ..... 68
Popis slika ..... 69
Popis tablica ..... 71
Popis kôdova ..... 72
Literatura ..... 73
Prilozi ..... 74

# 1. Uvod

Kroz stalni razvoj tehnologije i širenje opsega njene dostupnosti razvija se i učinkovitiji ljudski pristup prema rješenju raznih problema. Tako su u amaterskom i poluprofesionalnom sportu potrebne određene promjene u odnosu na postojeću organizaciju strukture turnira i interakcije saveza s igračima. Cilj bi trebao biti ispunjenje igračevih, odnosno korisnikovih želja, a to je prvenstveno struktura koja ih potiče da aktivno sudjeluju u klubovima i zajednicama i na taj način povećaju svoju aktivnost, prateći svoj realan napredak kroz vrijeme. Pritom se kao jedan od glavnih izazova pojavljuje upravo organizacija utakmica i turnira, a zatim i praćenje uspješnosti za svakog od sudionika pojedinačno, kao i njihovih timova. Nadalje, prisutno je i zemljopisno ograničenje – statistika o uspješnosti igrača iz jednoga dijela svijeta ne može se kvalitetno usporediti s onom nekog drugog igrača, pogotovo ako nisu igrali protiv istih protivnika.

U nedostatku univerzalne strukture i vremena uvelike može pomoći automatizacija takvoga sustava uz klasifikaciju igrača. Zbog zahtjevnosti praćenja svake utakmice pojedinačno, ta bi struktura idealno funkcionirala sa što manje ulaznih parametara o utakmicama od strane igrača. Istovremeno, imala bi što više izlaznih parametara poput podataka o uspješnosti igrača i timova i savjeta za uparivanje s drugim igračima za njihov zajednički napredak. Ovakav sustav omogućio bi jednostavnu upotrebu, a osigurao kontinuirano, intuitivno i učestalo ponašanje igrača kroz statistike, preglede i savjete. To je moguće ostvariti jer bi algoritamska primjena bila vrlo specifična, unaprijed određena i manje fleksibilna u početku tijekom stvaranja okosnice sustava.

Uspješna primjena i realizacija te okosnice ovisit će upravo o početnoj hipotezi, istraživanju i analizi potreba korisnika koja će biti provedena putem ankete s ciljanom skupinom budućih korisnika, nakon čega će funkcionalnosti i jasnoća dizajna i upotrebljivosti aplikacije biti ispitane putem intervjua na računalu, uživo ili putem Skype aplikacije. Na računalu će se naknadno mjeriti i trajanje izvođenja programa odnosno skripte koja računa poredak parova za uspješniju optimizaciju frekvencije izvršavanja algoritma i dobivenih rezultata. U konačnici, putem Google Analyticsa pratit će se općeniti trend ponašanja korisnika kroz njihovo stvarno korištenje aplikacije.

# 2. Poredak parova i pojedinaca

Sustav za rangiranje parova i pojedinaca u sportu jest onaj sustav koji analizira rezultate sportskih natjecanja kako bi se osigurala objektivna ocjena za svaku od ekipa i igrača.[1] Prema unaprijed definiranim pravilima, formulama i algoritmima procjenjuju se određene varijable prema podacima o igračima i njihovim ishodima, iz kojih se zatim izvode rezultati, odnosno broj zasluženih bodova, a na kraju se stvara konačan popis odnosno poredak na čijem je vrhu par ili pojedinac s najviše postignutih bodova. Varijacijama u algoritmu mijenja se konačni poredak, pa se algoritam formira prema određenim vrijednostima koje u sustavu služe kako bi najbolji sportaši bili istaknuti i nagrađeni.

## 2.1. Strategija i opseg

Amaterski i poluprofesionalni sportaši navikli su na postojeću i ustaljenu turnirsku strukturu koja im nije dovoljno pristupačna, dok su istovremeno svjesni da ta izvedba može biti unaprijeđena. U postojećoj strukturi moraju se registrirati kao članovi prepoznatih sportskih klubova i prijavljivati na određene turnire unutar sezone. Učestalost organizacije takvih turnira ponekad je rjeđa nego što bi to igrači željeli, pogotovo zbog složenosti izvedbe i financijskih mogućnosti. Stoga igrači skupo plaćaju kotizacije da bi pri sudjelovanju na turniru uglavnom bili prisutni u cjelodnevnom događaju, igrajući utakmice samo dio vremena i pritom provodeći većinu vremena čekajući svoj red za igru. Nakon odigranih utakmica, sportski savezi moraju pratiti i zabilježiti sve ishode, definirati vlastita pravila za poredak te ga izračunati i objaviti. Ta procedura može potrajati i više od tjedan dana od održavanja turnira, što umanjuje interes igrača zbog otežanog praćenja trenutnog stanja, a igrači se izvan turnira teško mogu uspoređivati.

Pored ovakve službene državne ili lokalne turnirske strukture, igrači često i sami organiziraju utakmice s prijateljima i poznanicima. U njima sami određuju kad i gdje će igrati, kao i s kim i protiv koga, te na taj način vježbaju za turnire. Praćenje napretka posebno im je otežano, pogotovo ukoliko bismo u obzir uzeli veće područje djelovanja poput onoga izvan

[1] Hoppenbrouwer, J., Winkels, M. *Sport Ratings*. Amsterdam, 2013., p. 2.

njihovih klubova. Uspoređivanje s međunarodnim okruženjem u tom kontekstu zvuči gotovo nemoguće.

Za početak, bilježeći uspjehe svojih igara kroz vježbe i neformalne utakmice, sportaši bi mogli pratiti svoje i tuđe napretke unutar sustava, čak i naprednije nego što im to nude službeni turniri. Uviđajući niz prednosti ovakvog igranja i praćenja statistika, dio će korisnika zasigurno u potpunosti usvojiti ovaj način praćenja, jer u njemu pored vlastitih utakmica mogu unositi i ishode službenih utakmica. Broj unesenih ishoda ovisit će o nizu čimbenika poput ukupnog broja igrača unutar sustava, generalne upotrebljivosti i dostupnosti kao i prihvaćenosti unutar sportskih klubova, kvalitete funkcionalnosti koja im se ponudi te o općem zadovoljstvu strukture poretka igrača prema novim izračunima, odnosno percepciji preciznosti algoritma koji će određivati poredak. Naknadno je pored neovisnih, manjih i spontanih utakmica moguće organizirati i posebne turnire i nagrade za najbolje rangirane sportaše unutar tog novog sustava.

### 2.1.1. Ciljevi projekta i potrebe korisnika

Glavni je cilj rada omogućiti igračima samostalno unošenje rezultata utakmica i praćenje statistike igara u timovima, odnosno parovima, kao i praćenje njihove osobne statistike, gdje će najveći utjecaj na proračun imati upravo ti rezultati o utakmicama koji su uneseni u sustav. Kako bi se ostvario taj, ali i glavni poslovni cilj popularizacije sustava i strukture, bit će ključno uzeti u obzir povećanu dostupnost i fleksibilnost za korisnike od početka. Stoga se kao rješenje predlaže stvaranje i korištenje web aplikacije kojoj će korisnici pristupati kroz svoj korisnički račun s bilo kojeg uređaja koji ima povezivost s Internetom. Iako bi mogao imati i širu primjenu u svijetu sporta, ovaj će projekt za početak biti razvijan i primijenjen na stvarnom primjeru za amaterske i poluprofesionalne klubove odbojkaša na pijesku u Njemačkoj, dok će nakon te prvotne razrade biti spomenute prilike upotrebe sličnih sustava i u drugim sportovima i ostalim stvarnim slučajevima.

#### 2.1.1.1 Odbojka na pijesku i praćenje ishoda

Odbojka na pijesku je sport koji se počeo razvijati početkom 20. stoljeća. Od polovice istog ima profesionalnu strukturu i status, a tek se 1996. godine prvi puta pojavljuje na Olimpijskim igrama. U svome su radu navedeni autori opisali osnovne značajke tog sporta na sljedeći način, preuzimajući službena pravila FIVB-a:

*Pravila su odbojke na pijesku u velikoj podudarnosti s pravilima dvoranske odbojke, ali imaju nekoliko značajnih razlika. Odbojka na pijesku igra se na pješčanome terenu gdje timovi imaju samo po dva igrača, za razliku od šest igrača po timu u dvoranskoj odbojci. Utakmicu dobiva momčad koja je osvojila dva seta po 21 bod, dok se u slučaju izjednačenog rezultata treći i odlučujući set igra do 15 bodova.* [2]

Budući da je prema ovim pravilima prilično jednostavno odrediti pobjednika utakmice, praćenje sveukupnih rezultata svih sudionika na turniru moguće je na nekoliko načina. Stoga je uloga upravo međunarodne federacije, kao i u svakom drugom sportu, da odredi pravila sudjelovanja na svim službenim turnirima, ali i uvjete za sudjelovanje na međunarodnim turnirima.[3] Službena FIVB-ova struktura u potpunosti je transparentna, pa je uvelike olakšano praćenje i osiguravanje poštene igre na način da timovi u turniru zauzimaju određene pozicije na ljestvici i prema tome im se dodjeljuju bodovi bez obzira na status u prethodnim turnirima.

Pored ove, postoji i niz neslužbenih struktura za praćenje ishoda turnira. Njihova je glavna mana činjenica da se moraju uhvatiti u koštac s pokušajima „varanja“ igrača, odnosno njihovim prilagodbama prema načinu na koji algoritmi funkcioniraju, ali je mana i što nisu u stanju usporediti igrače koji nisu iz istih klubova ili turnira. Često zbog svoje jednostavnosti algoritmi niti ne uzimaju u obzir sve moguće uvjete, poput igranja protiv slabijih protivnika ili uparivanja s jačim igračima, što su zapravo taktike za nepravedno i neutemeljeno povišenje vlastite pozicije na ljestvici. Uzimajući u obzir upravo tu činjenicu, presudan je uvjet algoritma da se sustav ne može prevariti odabirom partnera i protivnika, odnosno da ne postoji jasan uzorak kojeg bi igrači mogli pratiti kako bi na taj način povisili svoj poredak na ljestvici. Isti taj uvjet ključan je za uspješnu osnovnu usporedbu kvalitete igrača izvan vlastitih klubova, partnera i protivnika.

Ukoliko bi algoritam uspio ostvariti te preduvjete, to bi značilo da će igrači pored samostalnog bilježenja rezultata biti u mogućnosti samostalno i slobodno birati partnere i protivnike. Za razliku od klasičnih turnirskih struktura, mogli bi igrati svoje utakmice bilo kada i neovisno o velikim grupama ljudi, istovremeno zadržavajući poštenost i dosljednost

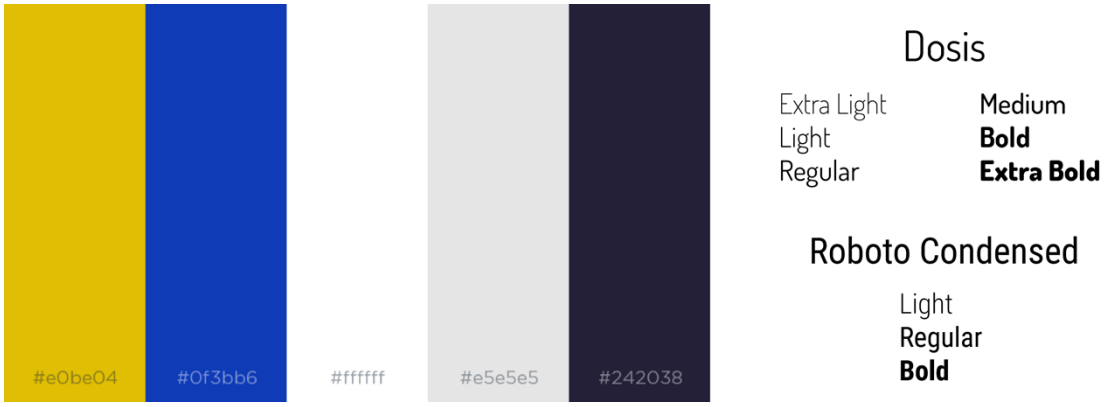
---

[2] Glickman, M. E., Hennessy, J., Bent, A. *A Comparison of Rating Systems for Competitive Women's Beach Volleyball*. Cambridge, MA, 2016., pp. 2-3.
[3] Glickman, Hennessy and Bent, *A Comparison of Rating Systems for Competitive Women's Beach Volleyball*, p. 2.

unutar sustava. Algoritam bi sve te njihove slobodne izbore s obzirom na stanje sustava u realnom vremenu ispravno vrednovao, što je i njegova jedinstvena prodajna vrijednost (engl. *unique selling point*).

#### 2.1.1.2 Uspostavljanje identiteta branda i buduće aplikacije

Budući da niti jedna ovakva javna aplikacija nije utemeljena kao relevantna unutar krugova timskih sportaša, pa tako ni među odbojkašima na pijesku, za početak je nužno uspostaviti snažnu marku (engl. *brand*) koju će korisnici u cjelokupnom smislu razumjeti, s kojom će se moći poistovjetiti i kojoj će vjerovati. Ta marka i aplikacija bit će im kroz sve vrste komunikacije predstavljene kao suvremenije rješenje za njihove potrebe. S vremenom bi ju usvojili kao svoje glavno središte za sportske aktivnosti. Kao naziv marke i aplikacije predlaže se jednostavno ime B.ELO, kratica koja se referira na odbojku na pijesku (engl. *beach volleyball*). Nadalje, naziv je inspiriran imenom Arpada Ela, tvorca jednog od popularnih algoritama za poredak. Slogan je jednostavan i u nekoliko riječi objašnjava cjelokupnu svrhu i cilj – vrednovanje odbojke na pijesku (engl. *evaluating beachvolleyball*). U vizualnom identitetu i logu naglasak je od početka na nazivu, a ne na znaku, dok će dominantne boje biti žuta, plava i bijela. Na odabir upravo tih boja utječe jednostavna poveznica boja odbojkaške lopte koje su u prošlosti zamijenile standardnu bijelu boju radi većeg kontrasta i lakše uočljivosti. Samim time logo će biti ujedinjena kombinacija jednostavnog *wordmarka* i jednostavnog *brandmarka* koji ne funkcioniraju jedan bez drugoga, a zajedno tvore ime B.ELO. Kao glavni font odabran je sans-serifni, visoki i zaobljeni Dosis, a njegov sporedni font bit će Roboto Condensed.




Slika 2.1 Odabrane boje i fontovi

Slika 2.2 Logo branda B.ELO (autor D. Bunta)

U početku će se promocija branda i korištenje aplikacije oslanjati isključivo na manju zajednicu njemačkih klubova i njihovu usmenu predaju. Tako bi prvotna primjena započela sa zatvorenim krugom ljudi i sustavom pozivnica za nove igrače. Pored poboljšane kvalitete testiranja učinkovitosti algoritma i rada aplikacije, prednost je manjeg broja ljudi sigurnost sustava u smislu inicijalnog sprječavanja stvaranja lažnih korisničkih računa i rezultata utakmica. U sljedećim fazama van opsega ovog rada namjera je uspostaviti upraviteljske račune za vođe klubova koji će imati kontrolu nad stvaranjem i praćenjem korisničkih računa, a u konačnici uspostaviti u potpunosti javni sustav s otvorenim pristupom.

Pored interakcija unutar aplikacije, sportaši bi u početku imali snažne interakcije s brandom i na društvenim medijima, na svojim dresovima i u svojim klubovima, kao i na ostalim sportskim rekvizitima poput lopti i steznika. Na svim turnirima koji će biti organizirani u sklopu klubova koji će koristiti aplikaciju za početak bit će jasno istaknut *branding*. Ton komunikacije prema korisnicima bit će opušten i ležeran i poticat će ih na aktivnost. Glavne emocije i impresije koje se žele postići su radost i sportska energija, preciznost, usredotočenost, pravednost i poštenje u igri te zdrava konkurencija.

Slika 2.3 Branding B.ELO-a u primjeni u početku

Glavni poslovni cilj za organizaciju bit će zarada od strane igrača u smislu naknada za korištenje aplikacije s naprednim mogućnostima, dok će igrači time štedjeti svoje vrijeme i

novac u smislu ubrzanog i digitaliziranog procesa koji će ujedno biti i jeftinije rješenje za njih. Pri stvaranju nove platforme prednost je što promocija u širem opsegu nije nužna za uspjeh u početku rada sustava jer je povjerenje u digitalnom smislu zahtjevno za postići kao novom brandu s novom domenom, serverom i uslugom. Pokazatelji uspjeha na koje će se obratiti posebna pozornost jesu:

- broj korisnika u početku u odnosu na broj korisnika nakon pola godine i godinu dana,
- broj unesenih utakmica u odnosu na broj korisnika,
- broj prihvaćenih i potvrđenih utakmica,
- broj klubova zainteresiranih za suradnju,
- broj sesija i posjeta aplikaciji te
- porast u vremenu zadržavanja unutar aplikacije u odnosu na nove značajke.

Vrijeme zadržavanja unutar web aplikacije direktno će ovisiti o periodu godine, odnosno sezoni i vremenskim uvjetima, ali i klupskim aktivnostima. Pri pokretanju aplikacije valja očekivati povećan generalni interes za novim iskustvom i funkcionalnostima. Unošenjem novih utakmica, najvjerovatnije s terena i što je brže moguće, povećat će se količina posjeta, ali će se smanjiti njihovo trajanje. Kasnije, generiranjem novih statističkih prikaza i poretka s mogućnošću naprednije analize produljit će se i vrijeme posjeta.

#### 2.1.1.3 Potrebe korisnika

U timskim sportovima poput odbojke na pijesku većina utakmica igra se u parovima, dakle dvoje igrača igra protiv drugih dvaju igrača. Pritom sva četiri igrača mogu pripadati istome klubu, neki igrači mogu pripadati drugim klubovima, a ponekad i određeni igrači u utakmici ne pripadaju niti jednome od klubova. Oni igrači koji pripadaju klubu uglavnom to čine zbog prijateljstava i emocija, ali i zdrave konkurencije, kako bi usporedili svoje rezultate s onima iz te grupe ljudi. Pri igranju utakmica, svaki od parova ili tek počinje igrati zajedno ili ima određenu povijest pobjeda i poraza iz prošlih utakmica unutar sustava. Pritom igrači najviše prate svoj rang kroz timove koje su utemeljili unutar kluba, ali i kroz svoju osobnu statistiku pobjeda i poraza u dosadašnjim igrama. Jednako tako zanimljiva im je statistika njihovih protivnika. Naprednijim je igračima kasnije zanimljiva i sveukupna, svjetska struktura poretka najposvećenijih sportaša. Tako je glavna potreba svih korisnika jedinstveno mjesto s autentifikacijom za upravljanje svojim sportskim aktivnostima i utakmicama te praćenje svojeg i tuđeg napretka.

Aplikacija je primarno namijenjena svih entuzijastima, amaterima i poluprofesionalnim igračima odbojke na pijesku u cijelom svijetu koji odigraju barem nekoliko utakmica u sezoni, bez obzira na njihovu dob ili spol. Ipak, zbog ograničenja pri stvaranju sustava, u obzir će biti uzete osobe isključivo muškog spola, dok će sustav za osobe ženskog spola te sustav tzv. miješanih igara (isključivo jedan muškarac i jedna žena u timu protiv istog takvog para) biti naknadno razvijen. Primarni jezik u komunikaciji bit će engleski jezik kao standard za kasniju širu geografsku primjenu.

Kao najzastupljeniji segment i primjer tipičnih korisnika našeg sustava ističu se odbojkaši od 25 do 44 godina starosti koji su zaposleni i većinu radnog vremena provode unutar ureda. Svoje slobodno vrijeme i sportske aktivnosti stoga vole unaprijed isplanirati, ali za njih imaju ograničene termine. Nepredvidive okolnosti mogu ih spriječiti u takvom planiranju, pogotovo ako je ono cjelodnevno, što može neizravno negativno utjecati na njihov rang. Zaslužuju sustav poretka koji se prilagođava njihovim potrebama i mogućnostima, a ne obrnuto.

Tablica 2.1 Persone prvih korisnika aplikacije

| ime i dob | **Viktor, 41** | **Klaus, 36** | **Uwe, 29** |
|---|---|---|---|
| trenutno radno mjesto | konzultant za optimizaciju poslovnih procesa | financijski savjetnik | front end inženjer |
| obitelj | oženjen, dvoje djece | u vezi, nema djece | oženjen, nema djece |
| sport i slobodno vrijeme | često vježba u teretani, uglavnom stigne igrati odbojku, barem 8 utakmica tjedno | voli trčanje, fotografiju i film, stigne igrati više nego drugi, barem 12 utakmica tjedno | slobodno vrijeme uglavnom provodi u prirodi, ne stigne uvijek igrati odbojku, 2 do 4 utakmice tjedno |
| prihodi | iznad prosjeka | iznadprosječno visoki | iznad prosjeka |
| razlozi za korištenje aplikacije | zanimljivo mu je biti dio zajednice, voli isprobavati nove stvari | voli pratiti statistiku kako bi što više napredovao | sloboda organizacije utakmica s prijateljima i poznanicima |

Nakon izrade algoritma i prototipa prvo će biti provedeno istraživanje o korisnicima koje će se usredotočiti na razumijevanje njihovog ponašanja, potreba i motivacija kroz anketu na kojoj će biti moguće zasnivati povratne informacije. Postavljajući vlastita pitanja i pokušavajući razumjeti u kojem trenu smo došli do konačnih odgovora unaprijedit ćemo njihovo iskustvo i povećati interes pri korištenju naše aplikacije.[4] Utvrđujući usklađenost parametara pretpostavki i istraživanja, bit će izrađen polazni prijedlog vizualne strukture web aplikacije. Prema njemu će se obaviti testiranje s pet različitih korisnika kako bi se provjerila uspješnost funkcionalnosti i snalaženja unutar sustava i napravile korekcije prije izrade konačne verzije i javne objave.

### 2.1.2. Funkcionalne specifikacije i zahtjevi za sadržajem

Praktični će se rad sastojati od prototipa u kojemu će težište biti na procjeni najnužnije potrebnih podataka i izradi samog algoritma, dok će se u drugome dijelu razrađivati aplikacija koja će biti orijentirana prema sportašima i njihovoj interakciji. Konačni popis funkcionalnih specifikacija će se i razlikovati u ovisnosti o tim segmentima rada. Pojedine stavke vjerojatno će naknadno biti dodane ili promijenjene u ovisnosti o rezulatima istraživanja s korisnicima.

Jedan od najvećih problema današnjih aplikacija jest nesrazmjernost ulaznih varijabli od strane korisnika i korisnih izlaznih varijabli koje za njih predstavljaju neko značenje. Što više ulaznih varijabli od korisnika zahtijevamo, više će vremena morati utrošiti u unošenje parametara. Time će se gomilati i količina podataka koju moramo pohraniti te produljiti njihova obrada pri prikazu statistika i informacija o utakmicama i igračima. Svodeći ulazne varijable pri praćenju statusa utakmica na minimum, uštedjet ćemo vrijeme korisnicima u interakciji koju i sami žele skratiti i ubrzati. Dodatno, tražeći od korisnika samo najnužnije podatke u onoj količini kojoj otprilike očekuju, smanjit ćemo šanse za greške pri unosu, a povećati želju, odnosno stopu ispunjavanja podataka te kvalitetu podataka.[5] Na taj način omogućit ćemo im korisniji utrošak vremena u radu s aplikacijom, praćenjem svojih i tuđih rezultata utakmica. Osim toga, određivanjem svih parametara na početku i kvalitetnim

---

[4] Hall, E. *Just Enough Research*. New York: A Book Apart, 2013., pp. 5, 37-55.

[5] Enders, J. *Designing UX: Forms*. Colingwood: Sitepoint, 2016., p. 54.

postavljanjem strategije i opsega organizacije i sustava stvaramo temelj za izvršenje ciljeva i time povećavamo šanse za uspješnost projekta.[6]

Pri izradi inicijalnog popisa najvažnijih funkcija prototipa i aplikacije glavna je vodilja omjer automatizacije kojeg je moguće postići u tom slučaju u odnosu na korisne parametre i podatke koje sustav bilježi. Uzimajući u obzir tu najmanju količinu potrebnu za rad i analizirajući neke od postojećih algoritama za stvaranje poretka, bit će stvorena konačna verzija „B.ELO algoritma“ za poredak parova i pojedinaca.

#### 2.1.2.1 Prototip aplikacije kao prvi korak

Unutar prototipa, naprednom korisniku za isprobavanje rada s algoritmom i podacima u svrhu prilagođavanja valja omogućiti sljedeće funkcije, gdje su zvjezdicom (*) označeni neobavezni unosi:

- unošenje novog kluba u sustav i osnovnih podataka o njemu:
    - naziv
    - grad
    - država (prema popisu svih država)
- unošenje novog igrača u sustav i osnovnih podataka o njemu:
    - ime i prezime
    - klub kojem pripada
    - spol
    - korisnički podaci za prijavu (e-mail adresa i lozinka)*
    - datum rođenja*
- unošenje nove utakmice u sustav prema kojoj će kasnije biti pokrenut izračun:
    - svi igrači unutar utakmice
    - pobjednik
    - mjesto održavanja*
    - datum i vrijeme održavanja*
    - (mogući dodatni parametri, ovisno o odabranom algoritmu i njegovim zahtjevima)
- brisanje posljednje unesene utakmice u sustav
- pregledi:

---

[6] Garrett, J.J. *The Elements of User Experience*. Berkeley: New Riders, 2011., pp. 35-37.

    - svih unesenih podataka
    - poretka timova i igrača sortiranih prema klubovima
- pokretanje izračuna poretka prema unesenim podacima
- unos velike količine podataka odjednom putem vanjske, strukturirane datoteke

Pritom bi sve funkcije u prototipu trebale biti omogućene od početka kako bi se na temelju njega mogla graditi pojedina verzija konačne aplikacije gdje se algoritam uspješno primjenjuje i u korisničkom smislu.

#### 2.1.2.2 Aplikacija za sportaše kao završni korak

Za razliku od prototipa, unutar konačne aplikacije značajke se mogu postupno otključavati za upotrebu, a svim je korisnicima u konačnici nužno omogućiti:

- registraciju i prijavu u sustav, kao i ponovno postavljanje lozinke te odjavu,
- prikaz početne stranice, odnosno kontrolne ploče (engl. *dashboard*) s osnovnim prikazima i utakmicama te njihovoj validaciji,
- unos nove utakmice,
- prikaz vlastite profilne stranice i izmjena profilne fotografije te prikaz tuđih profilnih stranica,
- prikaz profilnih stranica timova,
- prikaz poretka:
    - unutar vlastitog kluba
    - unutar drugih klubova
    - unutar svih klubova
- izdavanje pozivnice za registraciju poznanicima
- integraciju s društvenim medijima
- poziv drugim timovima, odnosno izazivanje na utakmicu
- sustav obavijesti o bitnijim promjenama koje se tiču korisnika te
- novosti unutar sustava ili kluba.

Detaljna razrada popisa svih potrebnih podataka za prikaz po stranicama ovisit će o specifikacijama algoritma za stvaranje poretka, pa tako i potrebe za specifičnim sadržajem unutar aplikacije. One se uglavnom očituju kroz jednostavan pisani materijal (engl. *copy*) za funkcije i preglede unutar aplikacije. Posebnu pažnju valja obratiti na pisanje i dizajn poruka greški na koje korisnik eventualno može naići pri korištenju. S razradom i testiranjem

jednostavnog sučelja podrazumijeva se i univerzalno razumijevanje funkcija i snalaženja korisnika bez dokumenta s uputama za korištenje. Stoga je zajedno sa širenjem opsega korisničke aplikacije uputno stvoriti stranicu često postavljenih pitanja u vezi algoritma, kao i kontakt formu ukoliko netko od korisnika poželi napisati specifičan upit za informacijom do koje nije moguće doći. Uz registraciju će biti raspisana izjava o zaštiti podataka zajedno s odredbama i uvjetima koje korisnici prihvaćaju pri stvaranju svog korisničkog računa počekom upotrebe aplikacije.

Analizirajući primjere pojedinih scenarija koji vode do različitih potreba i rješenja korisnika kroz web aplikaciju pobliže će nas usmjeriti prema stvaranju algoritma.

U prvome je primjeru ranije opisana persona Viktor na odbojkaškom terenu upravo pobijedio u svojoj trećoj utakmici u danu. U pregledniku svog mobilnog uređaja upisuje web adresu aplikacije i prijavljuje se u sustav gdje kroz navigaciju pristupa sučelju za unos utakmica i brzinski ih unosi. Nakon toga se odjavljuje iz aplikacije i završava s interakcijom.

Druga persona, Klaus, ovoga je vikenda zauzet zbog izvanrednog rada od kuće i ne može igrati utakmice, ali predvečer preko svog prijenosnog računala pristupa aplikaciji i pregledava najnovije utakmice koje su unesene u sustav. Pritom potvrđuje status dvaju svojih prethodnih utakmica koje su drugi igrači unijeli. Zatim kratko proučava svoju profilnu stranicu i profilnu stranicu dvaju svojih partnera s kojima najčešće igra utakmice. Proučava i najnovije igre svojih najčešćih protivnika iz drugih klubova. Prisjeća se da je obećao jednome od kolega poslati pozivnicu za registraciju, pa ju izrađuje i prosljeđuje. Zadovoljan s novim informacijama, odjavljuje se iz sustava i kontaktira jednog od partnera s kojim dogovara nove protivnike. Za kraj, pregledava njegov najnoviji album fotografija današnjih utakmica na društvenoj mreži i time završava interakciju.

Treća persona, Uwe, vraća se s poslovnog puta i prijavljuje se u aplikaciju kako bi provjerio koja se osoba u njegovom klubu trenutno najbolje podudara s njegovim vještinama i sposobnostima igranja. Zaključuje da je to jedan od njegovih bližih prijatelja te ga kontaktira kako bi dogovorio novu utakmicu u budućnosti. Dijeli svoju prijašnju pobjedu protiv najboljih igrača u svome klubu na društvenoj mreži i odjavljuje se iz aplikacije.

Zaključujući osnovnu strategiju i opseg, jasne su potrebe svih zainteresiranih strana i sljedeći korak je analiza drugih primjera algoritama te izrada onog koji je najprimjereniji spomenutim čimbenicima.

## 2.2. Postojeći algoritmi za stvaranje poretka

Uspješnom analizom nekoliko najpoznatijih i najčešće primjenjivanih algoritama za stvaranje poretka doći ćemo do poboljšane procjene nama potrebnih varijabli i tijeka izvođenja za slučaj ove aplikacije. Glavni zahtjev jest da pored stvarnog stanja i situacije u svakoj od utakmica od korisnika tražimo što manju interakciju pri unošenju podataka u sustav. Iz tih podataka u procesu izračuna bit će stvoren novi set informacija i podataka koji će utjecati na poredak sportaša, a u konačnome pregledu bit će prikazana samo jedna podskupina tih generiranih podataka, ovisno o uvjetima poput količine odigranih utakmica. Što je više podataka za svakog od igrača u sustavu, to će s većom sigurnošću biti moguće odrediti njihov poredak. Stoga je nužno razlikovati koncepte opsega kojim sustav upravlja i opsega korisničkih pregleda. Vizualizacija cjelokupnoga koncepta prema procesima rada algoritma u odnosu na stvarni korisnički proces igara i unosa prikazana je na slici niže. Slijedeći upravo taj koncept, bit ćemo u mogućnosti postići sve dosad postavljene ciljeve.

Slika 2.4 Prikaz količine podataka unutar algoritma prema procesima

Vjerovatno najpoznatiji i najčešće primjenjivani algoritam u sportskim ranigranjima jest onaj tvorca Arpada Ela za poredak šahovskih igrača. U njemu svaki od igrača pri ulasku u sustav, prije bilo koje interkacije dobije unaprijed definiranu početnu vrijednost bodova. Igranjem protiv drugih igrača, uspoređuju se njihove šanse za pobjedu na temelju trenutnog stanja bodova u sustavu i na temelju ishoda – pobjede, izjednačenja ili poraza, igrači u okršaju dobivaju ili gube bodove prema zadanoj formuli.[7] Unutar formule posebno je

[7] Elo, A. E. *The rating of chessplayers, past and present.* New York: Arco Pub, 1978. citiran u Glickman, M. E., Hennessy, J., Bent, A. *A Comparison of Rating Systems for Competitive Women's Beach Volleyball*. Cambridge, MA, 2016., pp. 7-8.

zanimljiv faktor K kojim se utječe na raspon bodova koji se mijenja uslijed ishoda okršaja. Zanimljiva je i odlika tog sustava da igrači ponekad smiju odudarati od svojih uobičajenih učinaka bez da trenutno budu kažnjeni ili nagrađeni.[8] Koristeći dijelove ovog algoritma u proračunu za timove, moći ćemo pobliže odrediti učinke i nagrađivanje odnosno kažnjavanje timova koristeći isključivo informaciju o pobjedi ili porazu. Ostaje nedorečena primjena u slučaju rangiranja pojedinaca koju valja naknadno razraditi. Specifična mana na koju valja obratiti pozornost jest da je česta pojava ovakvog sustava situacija u kojoj se noviji, ambiciozni igrači nalaze u otežavajućoj poziciji za napredak. To se događa jer rijetko imaju prilike igrati protiv boljih igrača, budući da oni pobliže biraju svoje protivnike.[9] Dodatkom unutar sustava koji bi ujednačavao rezultate usporedbom s prosjekom i usmjeravao bodove svih timova prema prosjeku mogli bismo bolje utjecati na sprječavanje odstupanja.

Algoritam procjene najveće moguće vjerodostojnosti jedan je od osnovnih principa u statistici. Pomoću njega je moguće procijeniti parametre statističkog modela na način da zadana skupina podataka poput utakmica unešenih u sustav bude najvjerojatnija od svih drugih ishoda unutar turnira. To bi značilo da je za precizan poredak svakog od timova nužno izračunati broj njihovih pobjeda u odnosu na zbroj svih njihovih igara protiv svakog od timova podijeljen zbrojem njihovih prethodnih procjena poredaka. Prednost koju bi iz sustava trebalo iskoristiti jest kontinuirana procjena koja je pravedna, jer bi pri proračunu uzimala vrijednosti svih timova u međusobnim odnosima, dok su mane otežana primjena u nepostojanom broju timova koji se dinamički mijenja i velik broj potrebnih proračuna pri svakom novom unosu.[10] Sličan bi proračun, uzimanjem u obzir svih parametara i suodnosa, mogao biti primijenjen u dijelu algoritma koji računa poredak pojedinaca.

Valja imati na umu činjenicu da se kao dobra praksa pokazalo pridavanje većeg značaja novijim rezultatima, bodovima i igrama kako bi preciznost tog predskazujućeg sustava bila što veća.[11]

---

[8] Hoppenbrouwer and Winkels, *Sport Ratings*, p. 9.
[9] Breznik, K., Bagatelj, V. *FIDE Chess Network*, Ljubljana, 2011., pp. 13-14.
[10] Hoppenbrouwer and Winkels. *Sport Ratings*, p. 7.
[11] Sorensen, S. P. *An Overview of Some Methods for Ranking Sports Teams*. Tennessee, 1999., pp. 1-3.

## 2.3. Problematika i korištene tehnologije

Pri stvaranju algoritma potrebno nam je sučelje prototipa kojem ćemo pristupati kroz web preglednik, a koje ima sve funkcionalnosti unošenja podataka koje su opisane ranije u specifikacijama. HTML i CSS kôd prototipa neće imati naglašen vizualni izgled jer je težište u toj fazi rada upravo na uspješnoj razradi tijeka izvođenja. Bit će povezani PHP programskim jezikom u kojem će se izvoditi programski kôd i cjelokupni algoritam. Budući da bi pri unosu igrača utakmice korisniku uvelike pomogao prediktivni unos, dio će kôda za to biti napisan u JSu, odnosno jQueryju. Jednostavnom bazom podataka upravljat ćemo kroz MySQL, odnosno MariaDB sustav, a upogonivši lokalni XAMPP web server omogućit ćemo sveukupno izvođenje i prikaz rezultata kôda. Tako će glavna interakcija s prototipom kroz rad biti stvaranje algoritma, a za eventualne će korisnike biti upravljanje velikom količinom podataka o sportašima i njihovim utakmicama. Ovaj je dio namijenjen entuzijastima i voditeljima saveza ili klubova koji stvaraju vlastite poretke i žele imati prilagođenu kontrolu nad parametrima i njihovim utjecajima na konačni poredak. Većina igrača neće imati doticaja s ovim dijelom aplikacije. Stoga neće biti izravne poveznice između prototipa i konačne aplikacije koju će igrači koristiti, pa će izmjena parametara u oba dijela projekta podrazumijevati izmjene u izvornome kôdu.

Za drugi će dio aplikacije biti nužno koristiti programski okvir Laravel i njegovi Query Builder i Eloquent Model, razvojni okvir Bootstrap te programske biblioteke poput jQueryja i Typeaheada koji će nam pomoći u bržoj i kvalitetnijoj izradi. Samostalno prateći detaljniju statistiku o pojedinoj utakmici na terenu uživo, pri njenom će završetku igrači sami zaključiti tko je pobjednik. Tu jednostavnu informaciju o igračima utakmice i pobjednicima unijet će u sustav. Bit će nužno uklopiti sustav povjerenja u kojem će korisnici međusobno potvrđivati unesene utakmice koje ih se izravno tiču. Na taj način, samo će utakmice koje su potvrđene od strane oba tima biti uzete u konačni proračun rezultata i bodova.

# 3. Razrada algoritma

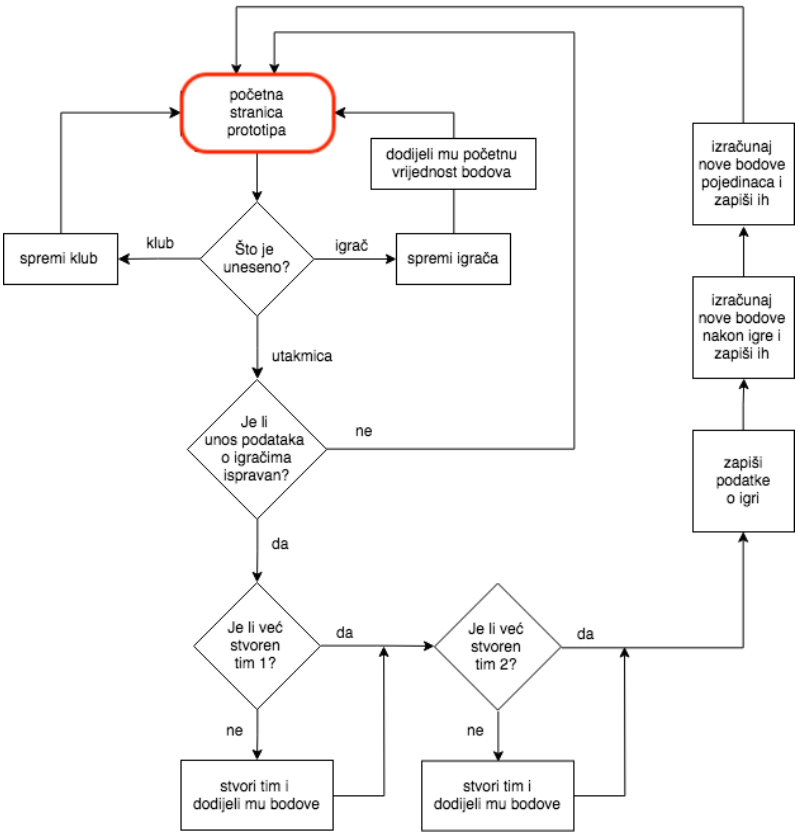
Počevši od prikaza za unos svih pojedinačnih podataka o igrama, gradit ćemo okosnicu modela za algoritam te odrediti što će nam biti ulazni parametri u sustavu. Te parametre unosit ćemo kroz formulare na jednoj web stranici. Oni će biti odvojeni prema interesnim objektima, a ispod njih će se nalaziti tablice s podacima koji su pohranjeni.

Prvi objekti koje možemo primijetiti unutar sustava su igrači, timovi, klubovi i igre. Svaki od objekata ima vlastite podatke koje moramo pratiti. Najjednostavniji objekt je klub kojem ovom prilikom dodjeljujemo naziv *Bubble*, a koji ima vlastito ime i državu kojoj pripada. Sljedeći od objekata jesu igrač koji ima svoje podatke i klub kojem pripada, zatim tim koji se sastoji od dvaju igrača te nekih parametara o bodovanju koje ćemo naknadno definirati. Tim je jedini objekt kojeg ne bi trebao stvarati korisnik kroz formular, već bi sustav samostalno trebao pratiti i bilježiti njihovo postojanje kako bi se smanjila redundancija u pohrani i prikazu. Na kraju je najsloženiji objekt utakmica koji se sastoji od dvaju timova i vlastitih podataka.

Pored najvećeg naglaska na dodjeli bodova pobjedniku igre, a oduzimanju gubitniku, temeljna je vrijednost algoritma procjena odabira partnera i protivnika. Ako tim igra protiv slabijih od sebe bit će kažnjeni, dok će biti nagrađeni ako igraju s jačima od sebe. Ta vrijednost nikako ne smije premašiti standardnu vrijednost izmjene bodova pri pobjedi ili porazu. U konkretnom primjeru, to znači da tim koji igra s jačima od sebe i radi toga je blago nagrađen, a izgubio je utakmicu, svejedno ne smije biti toliko nagrađen da dobije više bodova nego pobjednički tim. Drugim riječima, pobjednički tim ni u kojem slučaju ne smije gubiti bodove zbog svoje pobjede, čak ni ako igra s puno slabijim protivnicima. Jednak sustav vrijedi pri naknadnom računanju pojedinačnih rezultata za svakog od igrača iz timova. Ako igra s iznimno slabijim partnerima, bit će više nagrađen, dok ako igra sa blago slabijima od sebe i dalje će biti nagrađen, ali razmjerno manje. Jednako tako, ako igra zajedno s malo jačima od sebe bit će blago kažnjen, a ako igra s puno jačim partnerima od sebe bit će najjače kažnjen. Te vrijednosti procjene razlike u vještinama bit će određene kroz varijabilni koeficijent kojim će se moći utjecati na konačne rezultate. Povećavanjem tog koeficijenta u proračunu, povećavamo njegov značaj u sustavu.

## 3.1. Tijek izvođenja

S obzirom na tri tipa unosa objekata iz formulara prototipa, program će unijeti novi klub, igrača ili utakmicu. Kod unosa igrača potrebno je automatski unijeti i određene početne bodove koje će dobiti unutar sustava, dok je kod unosa utakmice potrebna provjera unesenih podataka o igračima te prekid rada ukoliko postoji pogreška pri unosu.




Slika 3.1 Glavni tijek izvođenja unosa podataka

Unos podataka za klub jednostavan je tzv. INSERT upit u bazu podataka u kojem unosimo njegovo ime i državu s predefiniranog popisa. Kad je gumb za unos na web stranici pritisnut, odnosno ukoliko je zadovoljen uvjet `isset($_POST["btnBubble"])`, pokreće se dio kôda. Podatke je nužno filtritati zbog sigurnosti sustava što činimo funkcijom `mysqli_real_escape_string()`. To ćemo činiti i u budućnosti za svaki korisnički unos, a korištenjem pripremljenih upita (engl. *prepared statements*) dodatno ćemo smanjiti izglede neželjenih unosa. Upiti za unos igrača i utakmice slični su, uz iznimku da kod unosa igrača nužno moramo odrediti i početnu vrijednost bodova koje će dobiti. Obično se ta vrijednost u sustavima postavlja na 1000, ali česta je i primjena od 1200 početnih bodova. Ukoliko se odlučimo za 1000 kao našu vrijednost, prvim gubitkom bodova igrači će pasti na

troznamenkastu vrijednost, a i bit će manje ukupnih bodova u sustavu. Stoga ćemo u ovoj verziji odmah isprobati vrijednost od 1200 kako bi se ti učinci ublažili. Bilježenje tog podatka bit će naknadno objašnjeno.

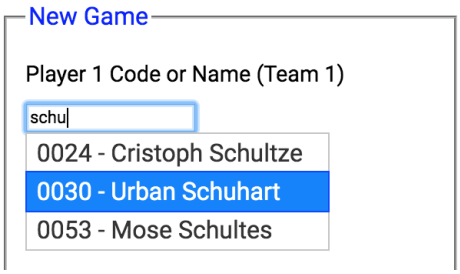
```
if(isset($_POST["btnBubble"])) {
  $bubbleName = mysqli_real_escape_string($con,
                $_POST["bubbleName"]);
  $bubbleCountryID = mysqli_real_escape_string($con,
                     $_POST["bubbleCountry"]);

  $stmt = $con->prepare("INSERT INTO Bubble (Name, CountryID)
                         VALUES (?, ?)");
  $stmt->bind_param("si", $bubbleName, $bubbleCountryID);

  if($stmt->execute()) {
    $messageBubble = 'New bubble added successfully.';
  } else {
    $messageBubble = 'Error while adding new bubble.';
  }
  $stmt->close();
}
```

Kôd 3.1 Program za unos kluba u bazu podataka

U varijabli `$con` sadržani su parametri za spajanje na lokalnu bazu podataka, a ona je definirana ranije u kôdu.

### 3.1.1. Unos parova

Svaki će igrač imati jedinstveni četveroznamenkasti identifikacijski broj, počevši od 0000 ili 1000. Osim toga, u pretrazi i unosu igrača pretpostavlja se da će se izvoditi po identifikacijskom broju, imenu ili prezimenu. Stoga će se struktura unosa sastojati od identifikacijskog broja, imena i prezimena.




Slika 3.2 Prikaz strukture polja za unos igrača u novoj utakmici

Podatke o igračima ćemo dohvaćati iz baze podataka ovisno o unosu.

```
if (isset($_GET['term'])){
$return_arr = array();
try {
      $conn = new PDO("mysql:host=".DB_SERVER.";dbname=".
                        DB_NAME, DB_USER, DB_PASSWORD,
                        array(PDO::MYSQL_ATTR_INIT_COMMAND =>
                        "SET NAMES utf8"));
      $conn->setAttribute(PDO::ATTR_ERRMODE,
                          PDO::ERRMODE_EXCEPTION);
      $stmt = $conn->prepare('SELECT CONCAT(LPAD(IDPlayer, 4,
                  "0"), " - ", Name, " ", Surname) AS Result
            FROM Player WHERE Name LIKE :term OR Surname LIKE
            :term OR LPAD(IDPlayer, 4, "0") LIKE :term');
      $stmt->execute(array('term' => '%'.$_GET['term'].'%'));
      while($row = $stmt->fetch()) {
         $return_arr[] =  $row['Result'];
      }
} catch(PDOException $e) {
      echo 'ERROR: ' . $e->getMessage();
}
echo json_encode($return_arr);
```

Kôd 3.2 Program za pretragu i dohvaćanje podataka o igračima

Prikaz podataka o igraču unutar formulara koristit će upravo te aktivno dohvaćene podatke.

```
$(function() {
      $(".playerac").autocomplete({
            source: "searchPlayers.php",
            minLength: 3
      });
});
```

Kôd 3.3 Program za prikaz dohvaćenih podataka o igračima

Ovim smo načinom unosa riješili problem korisnika da ne mora nužno voditi brigu oko ispravnog načina za unos jer može jednostavno označiti vrijednost koju mu pretraga prikaže. To se poklapa s ispravnim dizajnom sustava u kojemu ne kažnjavamo korisnika jer ne slijedi

točno propisane upute koje nisu u uobičajenoj primjeni, već mu osiguravamo korištenje prema već stečenim navikama.[12]

Ako razmotrimo niz znakova (engl. *string*) tog polja, primijetit ćemo da se prvi dio prije prvog razmaka sastoji od identifikacijskog broja, nakon čega se pojavljuje znak crtice i novi razmak te slijede ime, razmak i prezime. Funkcija `explode()` u PHP-u omogućuje nam da takav niz znakova rastavimo na manje poddijelove prema znaku graničnika (engl. *delimiter*), odnosno razmaku, a `(int)` ispred oznake varijable prisilno određuje (engl. *cast*) tip podataka kao cijeli broj. Time uspješno provjeravamo ispravnost unosa i uzimamo samo relevatni podatak, odnosno identifikacijski broj kao prvi element rastavljenoga polja za kasniju obradu.

```
$gameP1 = mysqli_real_escape_string($con, $_POST["gameP1"]);
$gameP1E = explode(" ", $gameP1);
$gameP1 = $gameP1E[0];
$gameP1 = (int)$gameP1;
$dupeArrayTest[] = $gameP1;
```

Kôd 3.4 Program za filtriranje identifikacijskog broja iz unosa

Taj broj dodajemo u polje `$dupeArrayTest[]` kako bismo nadalje provjerili ispravnost unosa – niti jedan igrač ne može igrati sam sa sobom ili protiv sebe, odnosno, ne smije biti unesen u popis igrača utakmice više od jedanput. Nakon što to napravimo za sva četiri igrača i u polju imamo sva četiri identifikacijska broja, možemo izvršiti provjeru ispitujući ima li u polju dvostrukih vrijednosti što ćemo činiti unutar funkcije `array_has_dupes()`. U njoj ćemo prebrojati sve vrijednosti te sve jedinstvene vrijednosti. Ako se ti brojevi poklapaju, funkcija će vratiti vrijednost 0, a ako pronađe manje jedinstvenih vrijednosti će vratiti 1.

```
function array_has_dupes($array) {
   return count($array) !== count(array_unique($array));
}
```

Kôd 3.5 Program za provjeru dvostrukih vrijednosti unesenih igrača

Program ćemo zato nastaviti izvoditi samo ako je vrijednost koju funkcija vrati jednaka 0. U tom slučaju možemo nastaviti s provjerom jesu li igrači već nekada bili upareni unutar sustava.

---

[12] Krug, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. San Francisco: New Riders, 2014., pp. 163.

```
$team1 = teamCheck($gameP1, $gameP2, $con);
if($team1 == 0) {
      list($team1ID, $team1Belo) = teamCreate($gameP1,
                                    $gameP2, $con);
}
else if ($team1 == 1)   {
      list($team1ID, $team1Belo) = teamGet($gameP1,
                                    $gameP2, $con);
}
```

Kôd 3.6 Logika za provjeru postojanja prvog od timova u bazi podataka

Funkcijom `teamCheck()` provjeravamo postojanje tima. Ako ne postoje, stvorit ćemo ih funkcijom `teamCreate()`, a ako postoje, dohvatit ćemo ih funkcijom `teamGet()`.

```
SELECT COUNT(IDTeam) AS Result FROM Team WHERE
(Team.PlayerID1 = ? AND Team.PlayerID2 = ?)
OR (Team.PlayerID2 = ? AND Team.PlayerID1 = ?)
```

Kôd 3.7 Upit za provjeru postojanja tima u bazi podataka u funkciji teamCheck()

```
SELECT IDTeam FROM Team WHERE (PlayerID1 = ? AND
PlayerID2 = ?) OR (PlayerID2 = ? AND PlayerID1 = ?)
```

Kôd 3.8 Upit za dohvaćanje tima iz baze podataka u funkciji teamGet()

Pri stvaranju novog tima i bilježenju događaja utakmice istovremeno je potrebno zapisati podatke o njihovim početnim bodovima. Ispočetka se kao najpoštenija opcija nameće dodjela 1200 početnih bodova svim novim timovima, kao što smo prethodno odredili i za sve nove igrače. Međutim, nakon određenog broja igara gdje sustav postaje precizniji, postavlja se pitanje koliko je pošteno timu dvaju igrača od primjerice 850 i 1000 bodova dodijeliti veći broj nego što to zaslužuju. Prva opcija jest aritmetička sredina dvaju vrijednosti pojedinačnih bodova igrača. Tako bi izračun za bodove novog tima (T) koji se sastoji od igrača s po 850 ($I_1$) i 1400 ($I_2$) bodova bio

$$T = \frac{I_1 + I_2}{2}$$

Jednadžba 1 Izračun za dodjelu bodova novom timu aritmetičkom sredinom

i iznosio bi 1125 bodova. Iako se ovaj rezultat doima poštenim, stvarna snaga tima ipak nije najpreciznija jer slabiji igrač tako igra u timu s velikom razlikom bodova u odnosu na svoje vlastite. Kako bi sustav ipak u obzir uzeo i tu razliku, koriteći jednadžbu inspiriranu Chebyshevljevim teoremom o nejednakosti zbroja izračun bodova bi bio

$$T = \frac{2 \cdot I_1 \cdot I_2}{I_1 + I_2}$$

Jednadžba 2 Izračun za dodjelu bodova novom timu inspiriran Chebyshevljevim teoremom

i iznosio bi 1057,777… bodova. Time je težište ukupnih bodova uspješno pomaknuto prema bodovima slabijeg igrača. Ovaj način dodjele bodova novim timovima ne bi smio biti usvojen od početka jer bi prve utakmice i prvi bodovi igrača u sustavu imali najviše utjecaja na snage timova. To bi se iznimno odražavalo i kasnije, što ćemo na primjeru skupa podataka i pokazati. Pitanje jest u kojem je trenutku najranije uputno uključiti ovaj način izračuna, a konačna odluka pada na voditelja liste poretka prema analizi i iskustvu te razlikama među igračima.

Radi upitne preciznosti računanja bodova i njihovih decimala, koristit ćemo biblioteku u PHP-u arbitratne preciznosti imena BCMath i njene funkcije. One će nam omogućiti rad s brojevima koji predstavljaju bodove unutar baze podataka.[13] Pri odabiru broja decimala i preciznosti sustava moramo uzeti u obzir stvarne potrebe, značaj, količinu pohrane te resurse koji su nam dostupni za izračune. Stoga za početak uzimamo preciznost od 6 decimala, dok ćemo korisniku prikazivati rezultate s 2 decimale, a u početku čak i rezultate bez decimala. Za zaokruživanje vrijednosti koristit ćemo ugrađenu funkciju `round()`, dok za skraćivanje bez zaokruživanja, odnosno odbacivanje decimala moramo napisati vlastitu funkciju.

```
function truncate_float($number, $places) {
      $power = pow(10, $places);
      if($number > 0){
            return floor($number * $power) / $power;
      } else {
            return ceil($number * $power) / $power;
      }
}
```

Kôd 3.9 Program za odbacivanje decimala broja

Cjelokupnu proceduru koju smo do sada opisali za prvi tim utakmice moramo ponoviti i za drugi, nakon čega možemo nastaviti s definiranjem unosa utakmice. Zapisujući informacije o pobjedniku, pozvat ćemo dvije funkcije koje će se izvršavati nakon svakog od unosa.

---

[13] PHP priručnik: *BCMath*, dostupno: https://web.archive.org/web/20190503142420/https://www.php.net/manual/en/book.bc.php [pristupano: 3. svibnja 2019.]

Prva će funkcija biti zadužena za sortiranje parova.

### 3.1.2. Sortiranje parova

Funkciji `teamCalculate` proslijedit ćemo sedam varijabli: identifikacijske brojeve oba tima, dosadašnje vrijednosti njihovih bodova prije utakmice, informaciju o pobjedničkom timu, redni broj zadnje igre u sustavu te informacije o vezi s bazom podataka.

Pomoću informacija koje se uvijek nalaze u sustavu, bez posebnih dodatnih procjena osim pri stvaranju novih igara, potrebno je uspostaviti odnos između bodova koje tim dobije ili izgubi zbog pobjede ili gubitka s faktorom koji objašnjava prethodne uspjehe ili neuspjehe tog tima unutar sustava. Taj će nam faktor donekle numerički prikazati kako se specifična promjena koja se događa završetkom ove igre uspoređuje s dosadašnjim događajima i igrama. Stoga ćemo za svaki od timova za početak pripremiti polje s podacima o identifikaciji njihovih prethodnih protivnika.

```
$W1array = array();
$W1data = array();

$stmtTeamWho1 = $con->prepare("SELECT IDGame, TeamID1,
TeamID2 FROM Game WHERE (TeamID1 = ? OR TeamID2 = ?)");
$stmtTeamWho1->bind_param("ii", $team1ID, $team1ID);
$stmtTeamWho1->execute();
$playerTeamFetch = $stmtTeamWho1->get_result();
while ($rowTW1 = $playerTeamFetch->fetch_assoc()) {
      if ($rowTW1['TeamID1'] != $team1ID) {
            $playerTeam = $rowTW1['TeamID1'];
      } else if ($rowTW1['TeamID2'] != $team1ID) {
            $playerTeam = $rowTW1['TeamID2'];
      }
      $W1data[$playerTeam] = 0;
}
```

Kôd 3.10 Program za identifikaciju dosadašnjih protivnika tima

U to polje ubacit ćemo i broj njihovih dosadašnjih utakmica s tim protivnicima.

```
foreach($W1data as $keyW1 => $valW1) {
      $stmtTeamWho2 = $con->prepare("SELECT COUNT(*) AS
      HowMany FROM Game WHERE (TeamID1 = ? AND TeamID2 = ?)
      OR (TeamID1 = ? AND TeamID2 = ?)");
```

```
        $stmtTeamWho2->bind_param("iiii", $team1ID, $keyW1, $keyW1,
        $team1ID);
        $stmtTeamWho2->execute();
        $playerTeamCountFetch = $stmtTeamWho2->get_result();
        while ($rowTW2 = $playerTeamCountFetch->fetch_assoc()) {
              $playerTeamCount = $rowTW2['HowMany'];
        }
        $W1data[$keyW1] = $playerTeamCount;
}
```

Kôd 3.11 Program za prebrojavanje dosadašnjih utakmica s protivnicima tima

U drugome će polju vrijednosti bodova protivnika biti unesene onoliko puta koliko su ti timovi igrali jedni protiv drugih.

```
foreach($W1data as $keyW1 => $valW1) {
      $stmtTeamWho3 = $con->prepare("SELECT BeloScore FROM
      TeamBelo WHERE TeamID = ?");
      $stmtTeamWho3->bind_param("i", $keyW1);
      $stmtTeamWho3->execute();
      $playerTeamScoreFetch = $stmtTeamWho3->get_result();
      while ($rowTW3 = $playerTeamScoreFetch->fetch_assoc()) {
            $playerTeamScore = $rowTW3['BeloScore'];
      }
      $playerTeamScore = truncate_float($playerTeamScore, 6);
      $repeatingTimes = $valW1;
      if ($repeatingTimes > 150) { $repeatingTimes = 150; }
      for ($i=0; $i < $repeatingTimes; $i++) {
            $W1array[] = $playerTeamScore;
      }
}
```

Kôd 3.12 Program za pripremu polja bodova dosadašnjih protivnika

Uzmimo za primjer neki tim s identifikacijskim brojem 8. Oni su već igrali protiv:

- tima 1 (s trenutnim stanjem bodova od 1400) dva puta
- tima 3 (s trenutnim stanjem bodova od 1300) šest puta
- tima 5 (s trenutnim stanjem bodova od 1100) tri puta

Polje s vrijednostima bodova svih protivnika tima 8 imat će vrijednosti [1400, 1400, 1300, 1300, 1300, 1300, 1300, 1300, 1100, 1100, 1100]. Pomoću tih vrijednosti izračunat ćemo koeficijent odvažnosti tima 8. On će biti jednak medijanu tih vrijednosti podijeljenim s

konstantom početnog broja bodova svih timova od 1200. U ovom primjeru medijan polja iznosi 1300 bodova, a podijeljen s 1200 daje koeficijent odvažnosti od 1,08333… Budući da je tim 8 prema tome dosad igrao protiv timova jačih od sebe, taj je koeficijent pozitivan. Broj dodatnih bodova koje će dobiti ovisi o razlici vrijednosti tog koeficijenta iz prethodne utakmice i nove vrijednosti koja je upravo izračunata. Ukoliko je ta razlika pozitivna, tim će dobiti dodatne bodove.

U obrnutome primjeru, ako bi razlika koeficijenata odvažnosti bila negativna, to bi značilo da je ovom utakmicom promjena u smjeru prema slabijim protivnicima i na temelju toga bi dobili negativne bodove. Kad bi promjena bila jednaka nuli, ne bi niti dobili niti izgubili dodatne bodove.

Pri ovom izračunu, odnosno pri ispunjavanju polja protivnika, postoji mogućnost za produljenje vremena trajanja izvršavanja kôda, budući da će se ona izvršavati u ovisnosti o broju protivnika i broju igara. Postavljanje gornje granice nije poželjno jer će to imati velik utjecaj na preciznost rezultata. Stoga je moguće kao rješenje postaviti sigurnosni limit od 200 ili više igara koji može biti pomican prema gore ovisno o dostupnim resursima za izračun. Idealno je rješenje uzimati u obzir samo igre do godinu dana starosti, čime ujedno ne opterećujemo igrače njihovim postupcima kojih se ne mogu niti prisjetiti.

Za izračunavanje medijana polja također ćemo morati pripremiti vlastitu funkciju.

```
function median ($arr) {
      sort($arr, SORT_NUMERIC);
      $count = count($arr);
      $mid = floor($count/2);
      if (($count % 2) == 0) {
            $output_sum = bcadd($arr[$mid--], $arr[$mid], 6);
            $output = bcdiv($output_sum, "2", 6);
            return $output;
      } else {
            return $arr[$mid];
      }
}
```

Kôd 3.13 Program za izračun medijana vrijednosti u polju

Isti ćemo postupak ponoviti i za drugi tim utakmice, nakon čega ćemo započeti s evaluacijom bodova koje će timovi dobiti na temelju ishoda utakmice. Oni se računaju uz pomoć ranije spomenutog Elovog algoritma.

Stanje bodova timova prije utakmice ($B_1$ i $B_2$) već smo predali funkciji. Iz njih ćemo izračunati pojednostavljeno stanje za oba tima ($P_1$ i $P_2$).

$$P_1 = 10^{\left(\frac{B_1}{400}\right)},\ P_2 = 10^{\left(\frac{B_2}{400}\right)}$$

Jednadžba 3 Pojednostavljeno stanje bodova timova u Elovom algoritmu

Zatim ćemo izračunati njihove očekivane rezultate ($O_1$ i $O_2$).

$$O_1 = \frac{P_1}{P_1 + P_2}, O_2 = \frac{P_2}{P_1 + P_2}$$

Jednadžba 4 Očekivani rezultati timova u Elovom algoritmu

Parametri statusa $S_1$ i $S_2$ iznosit će 1 ako je tim pobijedio, a 0 ako je tim izgubio. Faktor K određuje količinu promjene unutar sustava za svaki novi rezultat igre i može biti promjenjiv u ovisnosti o poretku tima. U našem će slučaju on iznositi 80 i bit će konstantan. Konačno, računamo nove bodove timova ($E_1$ i $E_2$) iz Elovog algoritma.

$$E_1 = B_1 + K\,(S_1 - O_1),\ E_2 = B_2 + K\,(S_2 - O_2)$$

Jednadžba 5 Konačni rezultati timova u Elovom algoritmu

Izračun bodova ovisit će o parametru statusa. Varijabla `$gameWon` koju unosi korisnik može iznositi 1 ako je prvi tim pobjednički ili 2 ako je drugi tim pobijedio. Shodno tome računamo konačne bodove jednostavnim uvjetom i pridodajemo ih rezultatima Elovog algoritma, varijablama `$ec1` i `$ec2`.

```
if($gameWon == 1) {
      $deltaBelo1 = bcmul($K, bcsub("1", $eb1, 6), 6);
      $deltaBelo2 = bcmul($K, bcsub("0", $eb2, 6), 6);
} elseif($gameWon == 2) {
      $deltaBelo1 = bcmul($K, bcsub("0", $eb1, 6), 6);
      $deltaBelo2 = bcmul($K, bcsub("1", $eb2, 6), 6);
}
$ec1 = bcadd($team1Belo, $deltaBelo1, 6);
$ec2 = bcadd($team2Belo, $deltaBelo2, 6);
```

Kôd 3.14 Program za izračun konačnih bodova timova u Elovom algoritmu

Ovo nisu konačni rezultati timova. Sljedeći je korak dohvaćanje koeficijenata odvažnosti prije utakmice kako bismo izračunali razliku ($K_R$) između nove ($K_N$) i stare ($K_S$) vrijednosti.

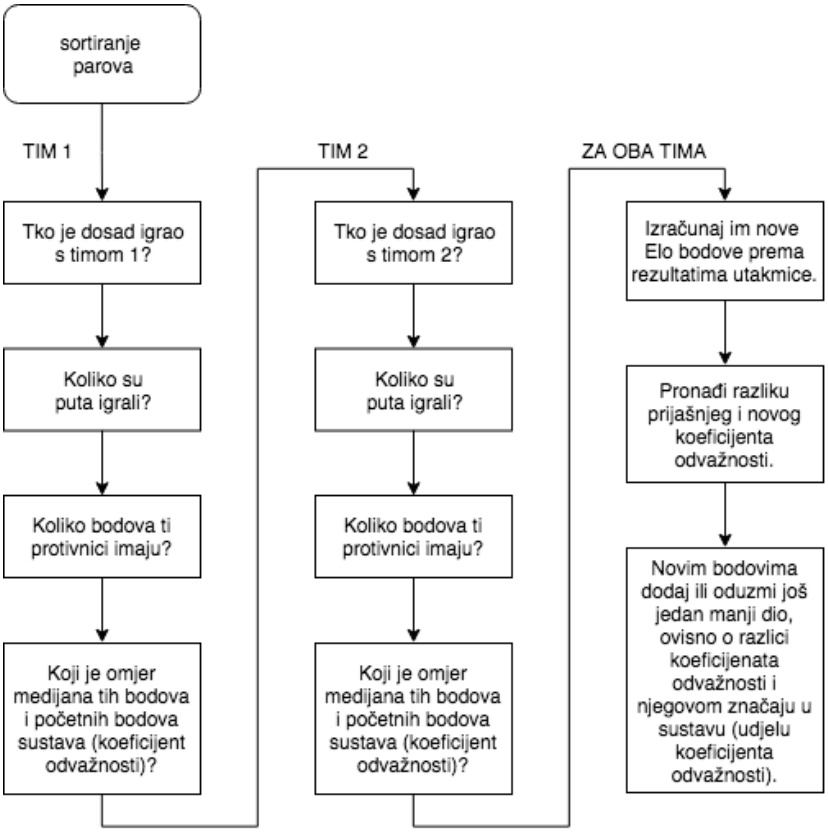
Uz njih, važno je spomenuti i faktor udjela koeficijenta odvažnosti ($K_K$) koji će određivati utjecaj, to jest količinu promjene unutar sustava, slično kao K faktor u Elovom algoritmu. U našem će sustavu u početku $K_K$ iznositi 0.5, a konačni bodovi timova ($T_1$ i $T_2$) iznosit će

$$T_1 = E_1 + E_1 (K_{K1} (K_{N1} - K_{S1})), \quad T_2 = E_2 + E_2 (K_{K2} (K_{N2} - K_{S2}))$$

$$T_1 = E_1 + E_1 (K_{K1} \cdot K_{R1}), \quad T_2 = E_2 + E_2 (K_{K2} \cdot K_{R2})$$

Jednadžba 6 Konačni rezultati timova u B.ELO algoritmu

Upravo ta razlika predstavlja glavnu promjenu u B.ELO algoritmu koja definira napredak timova. Ove konačne bodove zapisat ćemo u posebnu tablicu uz podatke o rednom broju igre u sustavu, kako bismo znali u kojem trenutku su timovi imali ove bodove. Na taj način uspješno čuvamo podatke o napretku bodova timova kroz vrijeme. Jednostavnije zapisano, ovaj dio algoritma za izračun bodova svakog tima u sustavu izvodi se na sljedeći način:




Slika 3.3 Prikaz algoritma za izračun bodova timova

Ovim smo postupkom uspješno zaključili bodove timova u danoj utakmici. Prije nego što sustav krene evaluirati sljedeću potvrđenu utakmicu, algoritam se može nastaviti i dalje i za ostale timove koji nisu bili dijelom te utakmice kroz dva različita procesa: revidiranje sustava i kalibracija sustava.

#### 3.1.2.1 Revidiranje sustava

Pri računanju bodova za tim koji je igrao utakmicu evaluirali smo njihovo stanje s obzirom na okršaj koji se dogodio, ali i na odvažnost koju su pokazali pomoću koeficijenta. Ako timovi općenito uzimaju duge vremenske pauze između igranja utakmica, njihovi bodovi i koeficijenti mogu značajno odudarati od stvarnoga stanja. Stoga bi se algoritam mogao nastaviti odvijati i za sve timove u sustavu, na način da pronađemo timove koji nisu bili u ovoj utakmici, a imaju neku povijest. Zatim izvučemo vrijednosti bodova njihovih protivnika, izračunamo koliko su puta igrali zajedno i na taj način i za njih pronađemo novu razliku koeficijenta odvažnosti i time i za njih izračunamo nove bodove nakon ove utakmice.

Osim ogromne količine podataka koju bi ovaj dio algoritma generirao, značajno bi produžio vrijeme u računanju bodova, što ćemo dokazati naknadno testiranjem vremena za izvođenje skripte i memorije koju baza podataka zauzima nakon izvođenja. Unatoč tome što bi preciznost izračuna bodova bila mnogo veća, upitna je ukupna održivost sustava, pogotovo uzimajući u obzir međunarodni opseg s mnogo igrača, timova i utakmica.

#### 3.1.2.2 Kalibracija sustava

U početku je oduzimanje i dodavanje bodova na temelju Elovog algoritma blago neprecizno, u smislu da dio bodova pri zaokruživanju nestaje ili nastaje, što se u sustavu ne bi smjelo događati. Na taj se način prosjek ukupnih bodova u sustavu odmiče u jednom ili drugom smjeru od početnih 1200 bodova, najveći bodovi postaju još viši, a najmanji još manji. Rješenje ovog problema jest kalibrirati sustav natrag prema početnom stanju od 1200 bodova. Ovaj se postupak sastoji od izvlačenja najnovijih bodova iz baze podataka za sve timove uključujući i najnovije podatke o dva tima iz utakmice te uzimajući aritmetičku sredinu svih timova, odnosno zbrajajući sve njihove vrijednosti bodova i dijeleći taj rezultat s ukupnim brojem timova. Na taj smo način dobili novi stvarni prosječni broj bodova u sustavu. Ponovno, za svaki od timova izračunat ćemo novo, preciznije bodovno stanje na način da njihov stari broj bodova pomnožimo s novim, prosječnim brojem i taj rezultat podijelimo s 1200 te zapišemo to kao najnoviji broj bodova tima.

Ključno je da se revidiranje i kalibriranje sustava izvodi prije zapisivanja najnovijih rezultata o utakmici, a uzimajući svježe izračunate podatke u obzir, kako bi se spriječilo kašnjenje u sustavu. Pri testiranju učinkovitosti izvođenja revidiranja sustava bit će testirana i učinkovitost izvođenja kalibracije.

### 3.1.3. Sortiranje pojedinačnih igrača

Drugom ćemo glavnom funkcijom `teamSingles` nakon izračuna bodova timova izračunati i bodove pojedinačnih igrača unutar sustava. Uz informacije o konekciji, funkciji ćemo proslijediti identifikacijski broj najnovije utakmice, kako bismo pratili trenutak u sustavu u kojem radimo izračun. Igračevu snagu moći ćemo procijeniti jedino na način da evaluiramo njihovu uspješnost unutar timova u kojima su igrali do trenutka evaluacije.

Za početak ćemo morati dohvatiti sve pojedinačne igrače iz baze podataka, pazeći pritom da uzimamo samo one koji su već odigrali neku utakmicu. To ne možemo riješiti jednostavnim ispisom igrača iz njihove tablice, već ćemo dohvatiti njihove identifikacijske brojeve iz timova stvorenih prilikom stvaranja utakmice.

```
SELECT PlayerID1 AS IDPlayer FROM Team
UNION SELECT PlayerID2 AS IDPlayer FROM Team
```

Kôd 3.15 Upit za dohvaćanje isključivo igrača s odigranim utakmicama

Za svakog od igrača, dohvatit ćemo i njihov dosadašnji broj bodova. Ako dosad nije igrao ni u jednoj utakmici, igraču je sigurno upisan barem početni broj bodova. Nakon toga, pronaći ćemo sve partnere s kojima je igrač dosad igrao, njihove pojedinačne rezultate te rezultate njihovih zajedničkih timova, kao i podatak koliko su puta oni zajedno igrali.

S obzirom na stvarni brojač tih timskih igara, razvrstat ćemo njihove partnere odnosno timove na sljedeći način:

- ako su zajedno igrali manje od 3 puta, računat ćemo kao da još nisu igrali zajedno
- ako su zajedno igrali više od 3 puta, a manje od 15 puta, uzet ćemo stvarni broj njihovih zajedničkih igara
- ako su zajedno igrali 15 ili više puta, računat ćemo kao da su igrali samo 15 puta, što predstavlja gornju granicu u ovom izračunu

Iz ovih podataka i logike proizaći će dvije vrijednosti prosjeka. Prvi je prosjek jačine timova (P*t*) koji nam za svakog igrača govori koji je prosjek bodova svih igračevih timova, uzimajući u obzir preračunati broj odigranih utakmica s njima, odnosno njihovu važnost za igrača. Taj prosjek predstavlja temelj za izračun bodova pojedinačnih igrača.

$$\mathrm{P}t = \frac{bodovi\ tima\ 1\ \cdot\ preračunati\ broj\ utakmica\ 1 + \ \ldots}{ukupan\ preračunati\ broj\ svih\ utakmica}$$

Jednadžba 7 Izračun prosjeka jačine timova (P*t*)

Drugi je prosjek jačine partnera (P$p$) koji nam na sličan način govori koji je prosječan broj bodova njihovih partnera, također uzimajući u obzir preračunati broj odigranih utakmica.

$$\mathrm{P}p = \frac{bodovi\ partnera\ 1\ \cdot\ preračunati\ broj\ utakmica\ 1 + \ \ldots}{ukupan\ preračunati\ broj\ svih\ utakmica}$$

Jednadžba 8 Izračun prosjeka jačine partnera (P$p$)

Nadalje, pri programiranju je nužno pripaziti da u slučaju bez utakmica ne dijelimo nulom, a u tom slučaju igračevi bodovi ostaju pri početnom broju bodova. U suprotnom nastavljamo s izračunom novih bodova za svakog od igrača.

Kako bismo odredili koliko je igrač općenito jači ili slabiji od svojih partnera, odredit ćemo omjer njegovih bodova i prosjeka bodova njegovih partnera.

$$O = \frac{broj\ bodova\ igrača - \mathrm{P}p}{1200}$$

Jednadžba 9 Izračun omjera igračevog izbora partnera

Ovaj nam podatak govori točno koliko su igračevi partneri jači ili slabiji od njega samog i na taj ćemo način kasnije odrediti broj bodova nagrade ili kazne za igrača. Ako igrač ima veći broj bodova nego njegovi partneri, to znači da češće igra sa slabijima od sebe, pa će $O$ biti pozitivan i obrnuto.

Kao i kod sortiranja parova, postavljamo faktor udjela omjera igračevog izbora ($K_K$) koji će predstavljati utjecaj nagrade ili kazne na konačni broj bodova igrača. U našem će slučaju iznositi polovicu koeficijenta timova, dakle 0.25 i konačno, dobivamo rezultat ($P_1$):

$$\mathrm{P_1} = \mathrm{P_t} + \mathrm{K_K}\ \cdot O\ \cdot \mathrm{P_t}$$

Jednadžba 10 Konačni rezultati pojedinaca u B.ELO algoritmu

Ukratko, ako za primjer uzmemo da je prosječan broj bodova igračevih timova jednak 1150, a igrač ima 120 bodova više od prosječnog broja bodova svojih partnera, to znači da je omjer igračevog izbora partnera pozitivan i iznosi 0.1, pa će konačni rezultat biti 1150 uvećan za 0.1 puta 1150 što je 115 bodova. Budući da taj bonus predstavlja iznimno visoku razliku u odnosu na uobičajenu promjenu, uvijek ćemo ga pomnožiti s faktorom udjela omjera igračevog izbora ($K_K$) te će konačni rezultat biti 1150 uvećan za 28,75 bodova, odnosno 1178,75 bodova.

Ovu proceduru moramo ponoviti za svakog igrača u sustavu koji je odigrao neku utakmicu i zabilježiti njihove najnovije brojeve bodova.

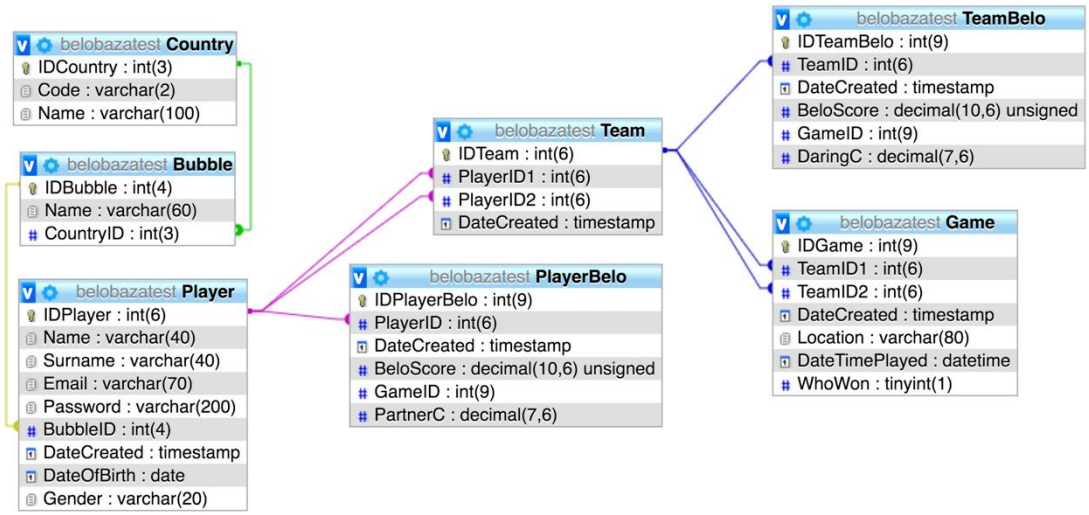
Postavlja se pitanje kada je sustav dovoljno precizan za prikaz s obzirom na broj varijabli koje uzima u obzir kako bi odredio snagu igrača i koji bi uvjet trebao biti zadovoljen kako bismo sa sigurnošću utvrdili broj bodova igrača, budući da oni najviše ovise o igračevim timovima. Za okviran prikaz pojedinačnih bodova igrača unutar aplikacije može biti određen minimalni broj igara po timu te minimalni broj timova koje igrač mora imati prije prikaza. Za primjer, igrač mora imati barem 5 timova s po 3 igre za prikaz bodova. U protivnom igrač dobiva obavijest kako je njegov rezultat u procesu određivanja. Nije nužno da igrač zna točne uvjete za rangiranje od početka jer će to naučiti kroz iskustvo korištenja aplikacije. Za razliku od klupskog poretka, u prikazu svjetske strukture potrebno je postaviti strože uvjeta, pa se može uzimati u obzir i broj timova igrača s partnerima i protivnicima izvan njegovog kluba. Uspješno rangiranje timova u tom poretku bit će od posebnog interesa za najozbiljnije igrače.

## 3.2. Priprema prototipa za primjenu

Definiravši glavni tijek izvođenja i algoritme od kojih se sastoji, nužno je podatke koje pohranjujemo, dohvaćamo i obrađujemo strukturirati prema nekoliko načela. Najbitnije je ono da unutar baze podataka nema redundancije te da postoji povijest o podacima igrača, ali samo ona koju je nužno pratiti kroz vrijeme. Ako igraču promijenimo identifikacijski broj ili ime, ta nam promjena neće biti presudna u sustavu. Najvažnije je pratiti promjenu broja bodova kroz vrijeme, pa će ti podaci o timskim i pojedinačnim bodovima imati vlastite tablice. Sve generirane podatke tek treba pripremiti i prikazati u prototipu, a nužno je razlikovati pojedinačni od masovnog unosa igara pri računanju i testiranju.

### 3.2.1. Konačna struktura modela

Pored zadovoljavanja osnovnih načela, u ovom je slučaju poželjna strategija uspostaviti bazu podataka na način da ishodi bodova timova i igrača uvijek mogu biti ispočetka preračunati iz tih istih podataka ukoliko dođe do neke pogreške. Primjerice, ukoliko se predomislimo oko koeficijenata, željeli bismo čuvati podatke o igrama, ali biti u mogućnosti samo pokrenuti novi izračun bodova. Isto vrijedi i u mnogo ozbiljnijem slučaju kasnije, ako se dogodi da netko od igrača prijavi da je kriva utakmica u sustavu potvrđena i bodovi već budu izračunati uključujući i nju. Vraćanjem na prethodno stanje, odnosno brisanjem novih bodova do trenutka krivog unosa i izmjenom statusa igara koje smo obrisali iz proračuna bodova moći ćemo ponovno pokrenuti izračun.




Slika 3.4 Nacrt objekata u bazi podataka prototipa i njihovih odnosa

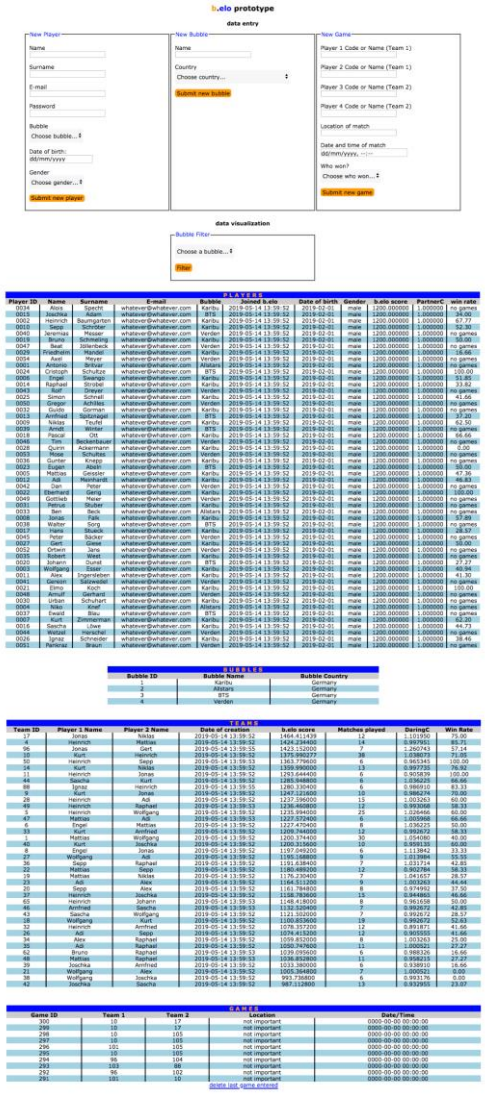
Cijeli je sustav sadržan upravo u nacrtu prikazanom iznad. Najmanji je objekt država (engl. *Country*) koja ima ISO 3166 dvoslovni kod i naziv. Slijedi ga objekt kluba (engl. *Bubble*) s državom kojoj pripada i nazivom. Zatim slijedi igrač (engl. *Player*) s imenom, prezimenom, e-mail adresom i lozinkom za login, klubu kojem pripada, datumu registracije, spolu te eventualno datumu rođenja. U tablici o pojedinačnim bodovima *PlayerBelo* zapisivat ćemo kojem igraču pripadaju, novi iznos bodova, novi omjer igračevog izbora partnera (*O*), pri kojoj igri u sustavu se zapis dogodio te datum unosa. Jedan igrač može igrati sa svim drugim igračima, što zapisujemo u tablici timova (engl. *Team*) samo jedanput za par, dakle bilježimo isključivo datum stvaranja te identifikacijske brojeve obaju igrača. Svaki od timova može imati svoje timske rezultate *TeamBelo*, pa slično kao i za pojedinačne bilježimo kojem timu pripadaju, novi iznos bodova, novi koeficijent odvažnosti ($K_N$), pri kojoj igri u sustavu se zapis dogodio te datum unosa. Konačno, zapisujemo podatke o igri (engl. *Game*), odnosno o dvama timovima, pobjedniku, datumu unosa, a možemo dodati i opcije poput datuma kada se utakmica odvila ili lokaciji.

U aplikaciji za korisnike će biti potrebno pratiti i identifikacijske brojeve korisnika koji unose i potvrđuju ili brišu utakmicu. Bez obzira na manje izmjene, okosnica i glavni odnosi će ostati isti.

### 3.2.2. Funkcije sinteze i prikaza podataka

Unose iz tablica baze podataka dohvatit ćemo i prikazati u jedinstvenom korisničkom pregledu za glavnog korisnika koji upravlja podacima u prototipu. Na istom mjestu ćemo omogućiti pojedinačni unos klubova, igrača i igara. Prema planu, prototip nema izričito razvijen dizajn već je u ovom dijelu razrade prioritet funkcionalnost i brzina.

Pri unosu manje količine podataka, korisnik to može učiniti kroz tri formulara za unos podataka za svaki od objekata. Za pregled podataka korisnik će pogled pomaknuti prema dolje, gdje se nalazi formular za filtiranje pregleda po klubovima te četiri tablice s podacima o igračima, klubovima, timovima i igrama. Na dnu se nalazi poveznica na akciju brisanja podataka o posljednje unesenoj igri. Navedene su sve najčešće operacije u radu s podacima i bazom podataka. Za sve ostale operacije s bazom, nužno je pisati vlastite SQL upite.




Slika 3.5 Vizualna struktura pregleda u prototipu

Prototip je organiziran kroz mnoge datoteke koje su međusobno povezane:

- *db_connection.php* – s informacijama za spajanje na lokalnu/online bazu podataka
- *deleteGame.php* – logika za brisanje posljednje unesene igre u sustav
- *index.php* (i *main.css*) – glavni dio prototipa koji povezuje sve funkcije
- *searchPlayers.php* – program za pretragu i dohvaćanje podataka o igračima (kôd 3.2)
- *submitBubble.php, submitGame* i *submitPlayer.php* – pojedinačni unos u bazu

Za pripremu spomenutog prikaza podataka koristimo još pet različitih datoteka u kojima smo definirali logiku iza SQL dohvaćanja podataka: *prepareBubbles.php, prepareCountries.php, prepareGames.php, preparePlayers.php* i *prepareTeams.php*.

U mapi functions nalaze se potporne datoteke koje se koriste unutar glavnih:

- *array_has_dupes.php* – za provjeru dvostrukih vrijednosti unesenih igrača (kôd 3.5)
- *median.php* – za izračun medijana vrijednosti u polju (kôd 3.13)
- *teamCalculate.php* – logika za glavno sortiranje parova
- *teamCalculateOHSA.php* – sortiranje parova s revidiranjem i kalibracijom sustava
- *teamCalculateSA.php* – sortiranje parova s kalibracijom sustava
- *teamCheck.php* – provjera postojanja tima u bazi podataka (kôdovi 3.6 i 3.7)
- *teamCreate.php* – stvaranje novog tima i zapis u bazu podataka
- *teamGet.php* – dohvaćanje tima iz baze (kôd 3.8)
- *teamSingles.php* – logika za glavno sortiranje pojedinačnih igrača
- *truncate_float.php* – odbacivanje decimala broja (kôd 3.9)

U posebnoj datoteci *csvLoad.php* pripremit ćemo logiku za unos velikog broja podataka iz CSV datoteke.

### 3.2.3. Masovni unos i brzina izvođenja

U slučaju kada moramo unijeti velike količine podataka, ne možemo koristiti jednostavni formular u zaglavlju prototipa. Čak i kada bismo jednom unijeli sve podatke u sustav i naknadno shvatili da moramo promijeniti neke varijable pri izračunu bodova, morali bismo iste podatke unositi ponovno. Ukoliko jednom pripremimo sve podatke koje želimo unijeti, strukturiramo ih standardiziranim pravilima u CSV datoteku i uskladimo s kôdom za masovni unos, jednostavno ćemo pretočiti podatke o klubovima, igračima i utakmicama iz datoteke u naš sustav prototipa prema parametrima u kôdu.

Za unos novog kluba u prototip, u redak CSV-a ćemo upisati

```
B;1;Peach
```

gdje B označava unos novog kluba (*Bubble*), 1 je ID oznaka kluba, a Peach je njegov naziv. Slično tome, za unos novog igrača u jedan redak upisat ćemo

```
P;0001;Antonio;Britvar;1
```

gdje je P oznaka za unos novog igrača (*Player*), 0001 je ID oznaka igrača, Antonio je ime, Britvar prezime, a 1 je ID oznaka kluba kojem će igrač pripadati. Konačno, za unos utakmice u sustav, u CSV upisujemo redak

```
G;0001;0002;0003;0004;1
```

gdje je G oznaka za novu utakmicu (*Game*) iza koje slijede dvije ID oznake igrača prvog tima i dvije ID oznake igrača drugog tima. Zadnja varijabla predstavlja informaciju o pobjedničkom timu, gdje je 1 vrijednost ako je pobijedio prvi, a 2 ako je pobijedio drugi tim.

```
1  B;1;Peach
2  B;2;Apple
3  B;3;Lemon
4  B;4;Apricot
5  P;0001;Antonio;Britvar;2
6  P;0002;Heinrich;Baumgarten;1
7  P;0003;Wolfgang;Esser;1
8  P;0004;Niko;Knef;2
9  P;0005;Mattias;Geissler;1
10 P;0006;Engel;Swango;1
11 P;0007;Kurt;Zimmerman;1
12 P;0008;Jonas;Falk;1
13 G;0001;0002;0003;0006;2
14 G;0001;0003;0002;0006;2
15 G;0006;0003;0002;0005;2
16 G;0006;0007;0008;0001;2
17 G;0002;0003;0006;0005;1
18 G;0002;0003;0006;0005;1
```

Slika 3.6 Primjer strukture CSV datoteke za unos podataka

U prototipu ne postoji sučelje za odabir CSV datoteke iz koje će se učitati podaci, već se njena lokacija prosljeđuje kao GET parametar u URL-u pri pokretanju *csvLoad.php* datoteke, primjerice `/csvLoad.php?fname=belodata.csv`.

Pomoću ugrađene funkcije `file()` učitat ćemo podatke iz CSV-a u varijablu u PHP-u i naknadno obrađivati informacije. Učitavajući redak po redak, prvo određujemo unosimo li klub, igrača ili utakmicu (B, P ili G), a zatim svaki od tipova podataka ima svoje informacije koje se unose u sustav. Za razliku od klasičnog unosa kroz formular, za ovakav unos podataka nužno je isključiti ograničenja (engl. *constraints*) u postavkama baze podataka budući da sami određujemo vrijednosti primarnih i stranih ključeva.

Nakon svake unesene utakmice izvodi se dio koda za sortiranje timova i za sortiranje pojedinih igrača. Na taj način kroz ukupno trajanje petlje pri masovnom unosu utakmica možemo mjeriti vrijeme koje je potrebno za izvođenje izračuna bodova i utjecaj na vrijeme kroz uređivanje parametara. Prvotni je cilj pronaći balans između zauzeća resursa (memorije i trajanja izvođenja koje ovisi o procesorskoj snazi računala) i razlike pri konačnim željenim vrednovanjima bodova u sustavu. Pripremimo li iznimno precizan, ali zahtjevan kôd kojem je potrebna velika količina vremena za izračun, sustav nikad neće zaživjeti.

Prema trenutno opisanim dijelovima algoritma, testirat ćemo izvođenje unosa više utakmica odjednom s različitim uvjetima izvršavanja kôda pri unosu. Unutar mjesec dana sezone s najboljim vremenskim prilikama, igrači iz 4 kluba odigraju otprilike 300 utakmica, što ćemo i uzeti kao broj utakmica za unos pri testiranju. Prvo ćemo izvoditi samo sortiranje timova nakon svakog od 300 unosa, zatim ćemo uz sortiranje izvesti i dio kôda za revidiranje sustava, a onda ćemo sustav pokušati revidirati rjeđe. Da bismo uvidjeli efekt kalibracije sustava na izvođenje kôda, njega ćemo izdvojiti kao zaseban test, a zatim također pokušati kalibrirati sustav rjeđe (a ne nakon svake igre). U konačnici ćemo pokušati pronaći optimalan omjer rjeđeg izvođenja revidiranja i kalibracije sustava pri unosu više utakmica.

Tablica 3.1 Vrijeme izvođenja i veličina baze pri različitim načinima unosa utakmica

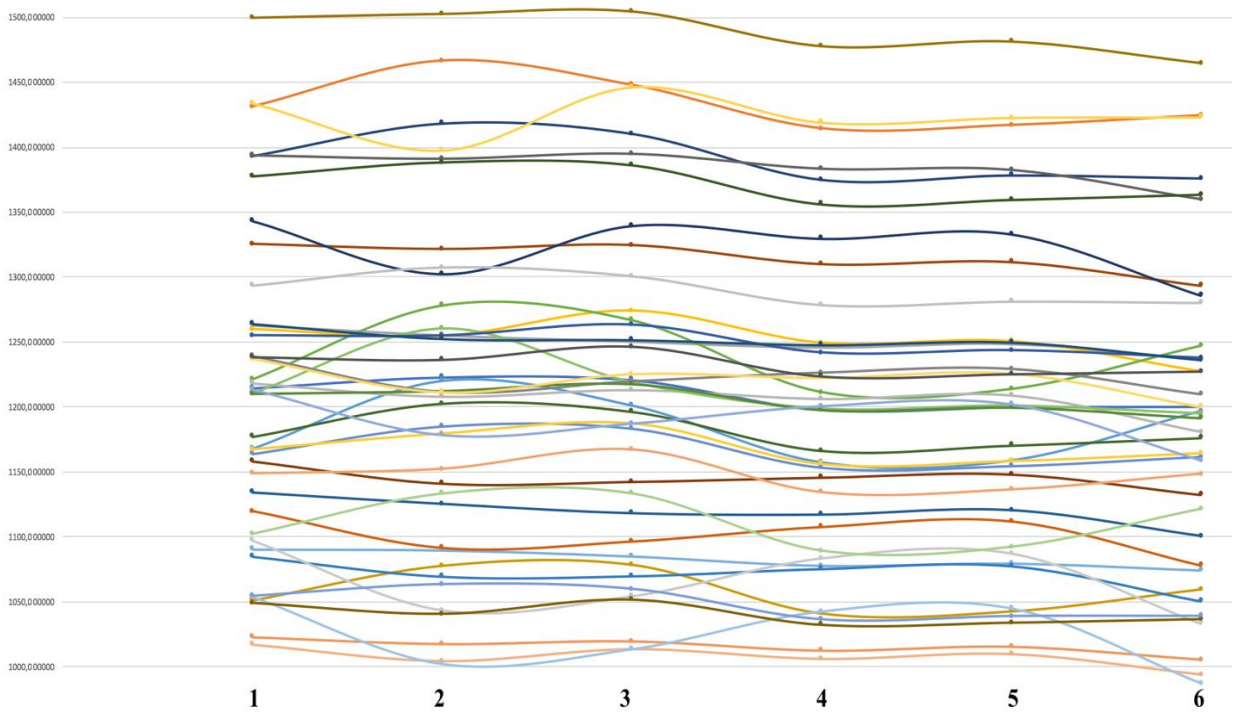
| # | opis | parovi | revidiranje sustava | kalibracija sustava | pojedinci | vrijeme (s) | veličina (kB) |
|---|---|---|---|---|---|---|---|
| 1 | sortiranje parova svaku igru | 1 | | | | **2,17** | **320** |
| 2 | sortiranje parova svaku igru + revidiranje sustava svaku igru | 1 | 1 | | | **91,98** | **2100** |
| 3 | sortiranje parova svaku igru + revidiranje sustava svaku 30. igru | 1 | 30 | | | **3,83** | **384** |
| 4 | sortiranje parova svaku igru + kalibracija sustava svaku igru | 1 | | 1 | | **8,28** | **2100** |
| 5 | sortiranje parova svaku igru + kalibracija sustava svaku 8. igru | 1 | | 8 | | **2,84** | **432** |
| 6 | sortiranje parova svaku igru + revidiranje sustava svaku 48. igru i kalibraciju sustava svaku 8. igru | 1 | 48 | 8 | | **4,03** | **480** |
| 7 | sortiranje parova svaku igru + revidiranje sustava svaku 48. igru uz kalibraciju sustava i sortiranje pojedinaca svaku 8. igru | 1 | 48 | 8 | 8 | **4,63** | **512** |

Testiranje je izvedeno na osobnom računalu s 2 GHz Intel Core i7 procesorom i 8 GB 1600 MHz DDR3, što je iznimno velika količina resursa pri usporedbi s resursima standardnih poslužitelja u oblaku.

Klasično sortiranje parova izvedeno je iznimno brzo i učinkovito; za nešto više od 2 sekunde moguće je izračunati obični poredak parova. Kada bismo sustav revidirali svaku igru, zauzeće memorije i vrijeme izvođenja zagušile bi sustav i učinile ga potpuno neupotrebljivim. Rjeđim revidiranjem ne žrtvujemo veliku razliku u bodovima parova, a značajno smo unaprijedili izvođenje kôda. Kalibracija nakon svake igre nema toliko drastične posljedice poput revidiranja, ali svejedno je s minimalno rjeđim izvođenjem u petlji moguće postići dobre rezultate. Rjeđim revidiranjem (svaku 48. igru) i kalibracijom (svaku 8. igru) sustav postaje vrlo učinkovit, a nema veće razlike u broju bodova parova.

Za potrebe dodatnog testiranja, uključivanjem rjeđeg sortiranja pojedinaca nema veće razlike u vremenu izvođenja, ali izvođenje će se značajno usporiti s više pojedinaca kvalificiranih za izračun i više od 300 utakmica unutar sustava. Na taj efekt možemo utjecati postavljanjem uvjeta za izračun bodova za pojedince, pitajući se u kojem trenutku je sustav dovoljno precizan da bez drastičnih odmaka u bodovima prikaže broj bodova za pojedine igrače.

Za analizu preciznosti izvedbi algoritma prikazani su bodovi svih 40 timova pri 6 različitih načina izvođenja algoritma (izuzev 7. načina u kojem nema promjene u bodovanju parova).




Slika 3.7 Razlike u bodovima parova pri različitim načinima unosa utakmica

Horizontalna os na grafu iznad predstavlja šest slučajeva izvođenja algoritma, odnosno načina unosa utakmica, dok vertikalna os predstavlja linearnu vrijednost broja bodova tima. Prema tome, jedna krivulja predstavlja broj bodova jednog tima pri šest različitih slučajeva. Prvi je zaključak kako se timovi na krajnjim rubovima grafa, odnosno oni s najvećim i najmanjim brojem bodova gotovo i ne mijenjaju različitim izvođenjem algoritma. Najslabiji je tim u svih šest slučajeva najslabiji, a isto vrijedi i za najjači. To je dobar pokazatelj u smislu da uvijek slično vrednujemo pobjednike i ne gubimo te odnose mijenjajući parametre algoritma. Posebno je zanimljivo primijetiti razlike u bodovima između 2. i 3. slučaja (kada sustav umjesto svake igre revidiramo svaku 30. igru), 4. i 5. slučaja (kada sustav umjesto svake igre kalibriramo svaku 8. igru) te 1. i 6. slučaja (kada uspoređujemo klasični poredak parova s unaprijeđenim). Najveća promjena u usporedbi sa svim slučajevima događa se u 2. slučaju sa stalnim revidiranjem sustava. Budući da taj izračun traje i do 30 puta duže od uobičajenog, jasno je da je taj slučaj mnogo precizniji. Prateći uzorak za svaki od timova, konačni (6.) slučaj sortiranja parova gotovo da predstavlja sredinu između klasičnog (1.) i najpreciznijeg (2.) slučaja.

Ovim smo testiranjem i konačnom odlukom o ispravno podešenom algoritmu (7.) postavili konačne temelje za daljnji razvoj klijentske strane aplikacije. Mnoge značajke aplikacije i dalje nisu u potpunosti usklađene s ovim sustavom, a igračima bi mogle biti od presudne važnosti za praćenje svojih rezultata. No u rudimentarnoj, testnoj verziji, korisnici će samostalno moći unositi informacije, a naš će algoritam prema tim unosima biti usklađen, određen i podešen prema potvrđenim razinama utjecaja koeficijenata.

# 4. Dizajn aplikacije

## 4.1. Istraživanje, prikupljanje informacija i potreba korisnika

Svrha je korištenja aplikacije od strane korisnika definirana pri razvoju algoritma, ali pojedine značajke aplikacije mogle bi odstupati od onih koje korisnici u početku žele, što ćemo provjeriti istraživanjem. Sistematskom istragom o korisnicima stječemo percepciju o njihovim navikama. Ukoliko početno istraživanje provedemo uspješno, povezat ćemo ono što je uistinu potrebno korisnicima i ono što je potrebno našoj tvrtki ili projektu u zajedničku i smislenu cjelinu.[14]

Na temelju ranijih pretpostavki i neformalnog istraživanja s tipičnim predstavnicima ciljane skupine prvih korisnika aplikacije postavili smo okvir i temelj za prvu verziju razvoja algoritma. Kako je ova aplikacija namijenjena specifičnoj grupi ljudi, nužno ih je vrlo dobro upoznati kako bismo za njih mogli dizajnirati bilo kakav proizvod. Pri izradi vizualnog sučelja tako moramo saznati više o motivima i namjerama, potrebama, očekivanjima, uzorcima ponašanja i navikama odbojkaša na pijesku. S tim ćemo saznanjima biti u mogućnosti izraditi prototip aplikacije u kojoj će oni moći postići sve potrebne osobne ciljeve i izvršavati zadatke u što manje koraka.

Kao sasvim novi proizvod kojeg većina sportaša nije navikla koristiti, aplikacija će u razvojnoj fazi imati jednostavan okvir. Daljnji razvoj bit će dinamičan i teško predvidljiv, pa ćemo stoga u istraživanju za početak postavljati osnovna pitanja s tek par naznaka značajki koje namjeravamo ponuditi u budućnosti. Tako će biti jednostavnije odrediti prioritete pri stvaranju plana izrade sljedećih verzija aplikacije.

### 4.1.1. Istraživanje i pretpostavke

Parametri koji su nam bitni u istraživanju kako bismo uskladili sve ciljeve i dalje razvijali aplikaciju mogu se podijeliti u one koji se tiču osnovnih i dodatnih značajki koje ćemo omogućiti te one parametre koji su bitni za okosnicu i brzinu samog sustava. Većina će ispitanika biti mlađe osobe i osobe srednje dobi muškog spola, budući da su oni primarna

[14] Hall, E. Just Enough Research. New York: A Book Apart, 2013., pp. 12-15.

ciljana skupina na kojoj će biti testiran razvoj aplikacije i kojima je aplikacija namijenjena. Mišljenja o značajkama uglavnom ćemo ispitivati na način da će ispitanici kvantitativno ocijeniti značaj što ranije prisutnosti neke značajke u aplikaciji, dok ćemo odgovore na pitanja o ispravnoj postavci okosnice i algoritma dobiti ispitivanjem njihovog ponašanja. Zanimat će nas vrste utakmica koje vole igrati, planirana učestalost unosa novih utakmica (što ćemo doznati kroz njihov dosadašnji broj tjednih utakmica u sezoni), željeni način prijave u aplikaciju te njihovi stavovi o osnovnoj strategiji za digitalni nastup aplikacije.

Pri inicijalnom istraživanju dobra je praksa definirati najviše par ciljeva i pritom postaviti sedam do deset glavnih pitanja. U suprotnom ciljamo previsoko, ponavljamo jednu te istu svrhu ili riskiramo neuspješnost ispunjavanja ciljeva zbog prevelikog opsega.[15]

Prije istraživanja, očekujemo da prosječan igrač neće unositi više od sedam utakmica tjedno, najviše vrednuje i igra utakmice u parovima (2v2), igra utakmice uglavnom s pripadnicima istog spola, najvažnija mu je osobna statistika i statistika njegovih parova, želi fleksibilnost pri korištenju aplikacije kako bi brzo unosio rezultate, a kasnije imao detaljniji pregled statistike, želi imati poseban korisnički račun za prijavu u aplikaciju te bi rado imao kontakt s brandom na većini društvenih mreža.

Kako je korisnicima zahtjevno unaprijed procijeniti svoje namjere u smislu korištenja aplikacije s različitih uređaja, nećemo ih izravno pitati za mišljenje o uređaju s kojeg će posjećivati aplikaciju. Umjesto toga, možemo ih neformalno pitati koliko svoje mobilne uređaje koriste u odnosu na osobna računala. Dakle, prema svjetskim trendovima i dobrim praksama, korisničko sučelje ćemo od početka postaviti na način da je responzivno i primarno namijenjeno uređajima s manjim ekranima.

### 4.1.2. Rezultati i odluke

Istraživanje je provedeno putem *online* ankete s 28 ispitanika muškoga spola i 2 ispitanice ženskog spola iz sjeverne Njemačke. Deset ispitanika pripada dobnoj skupini od 20 do 29, dvanaest ih pripada dobnoj skupini od 30 do 39, šestero ih ima između 40 i 49 godina, a dvoje je starije od 49 godina.

---

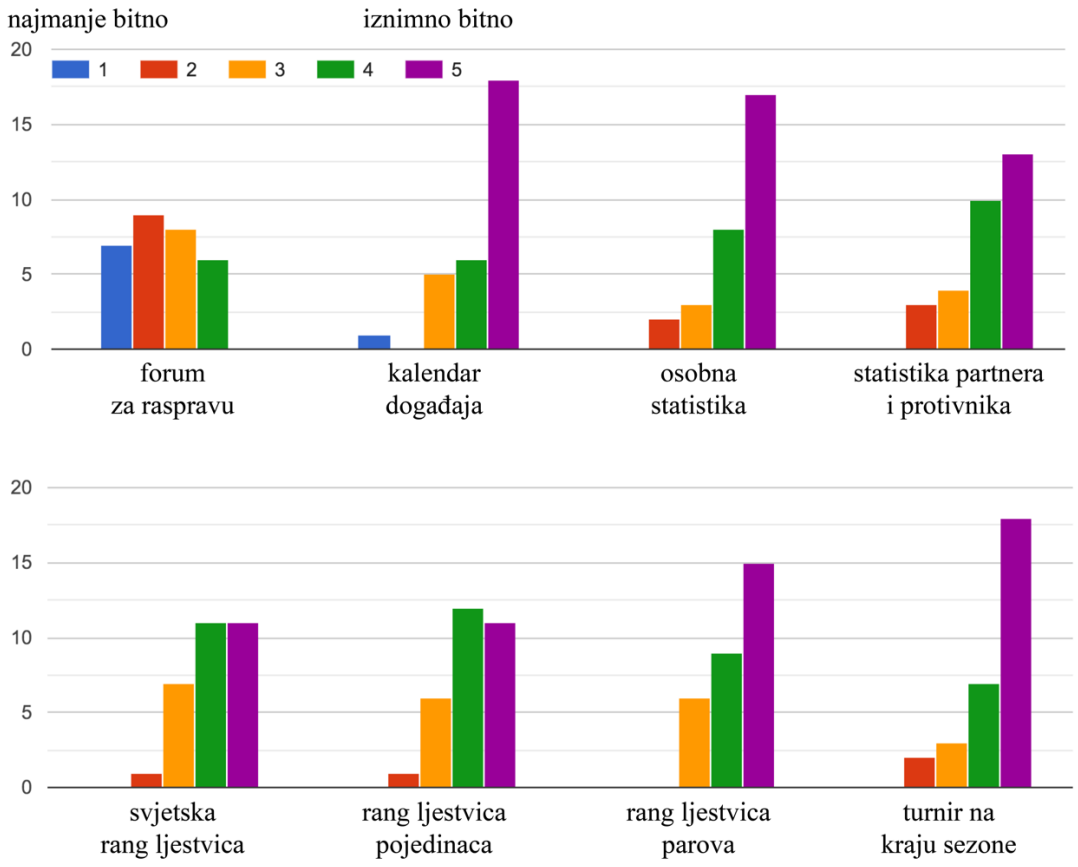
[15] Sharon, T. It's Our Research: Getting Stakeholder Buy-in for User Experience Research Projects. Waltham: Morgan Kaufmann, 2012., p. 76.

Za početak je bitno saznati koliko najviše utakmica igrači odigraju u vrhuncu sezone kako bismo mogli procijeniti broj unosa podataka i trajanja izračuna. Samo jedan ispitanik rekao je da igra više od 5 utakmica tjedno, više od trećine (njih 11) ih igra 3 do 4 utakmice tjedno, više od pola (njih 16) igra samo jednu do dvije utakmice, a dvoje ih igra i rjeđe od navedenog.

Na pitanje koju vrstu timova preferiraju, većina (93%) ispitanika izjavilo je da preferira parove (2v2), a dvoje ipak radije voli igrati s više ljudi u timu.

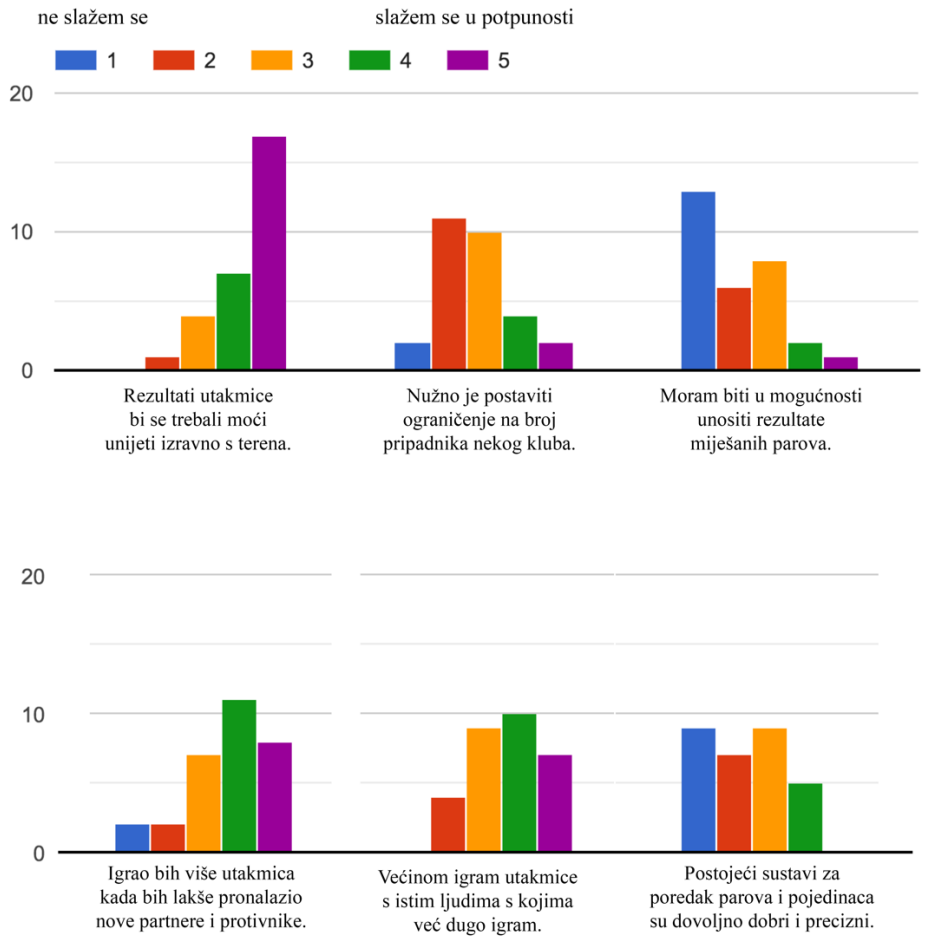
Nastavno na to pitanje, većina (87%) ispitanika je potvrdila da najčešće igra utakmice s partnerima i protivnicima istog spola, dok četiri ispitanika ponekad igra i „miješane“ utakmice.

Zatim smo ponudili osam najvažnijih značajki koje bi trebale imati svoje mjesto unutar prvih verzija aplikacije na ocjenjivanje, gdje je svaki ispitanik ocijenio važnost svake od značajki. Iako su mnoge značajke osnovne i planirane od početka, ovi podaci daju nam uvid u očekivanja još uvijek nepostojećeg proizvoda. Značajke s najvećim brojem najviših ocjena svakako će imati prioritet pri inicijalnom razvoju.




Slika 4.1 Rezultati istraživanja – najvažnije značajke buduće aplikacije

Nakon rangiranja prioriteta značajki, sljedeći je cilj razriješiti ključne nedoumice u glavnoj okosnici aplikacije postavljanjem bihevioralnih pitanja kroz skaliranje istinitosti pojedinih tvrdnji.




Slika 4.2 Rezultati istraživanja – stavovi o utakmicama i poretku

Igračima je najvažniji brz i bezbolan unos rezultata, nakon čega slijedi inteligentno uparivanje timova, a miješani parovi i ograničenja u klubovima su najmanje važne pojave za naše igrače. Pri ocjeni postojećih sustava za poredak parova i pojedinaca ispitanici su razjedinjeni. Velika je vjerojatnost da dio njih ne razumije načine na koji funkcioniraju poretci u potpunosti, a da je dio njih navikao na postojeće sustave i misli da razvija osjećaj za smisao poredaka koji su već dugo u upotrebi.

Većina (87%) se igrača želi prijavljivati u sustav posebnim korisničkim računom kojeg ne žele povezivati s društvenim mrežama, a ostali ispitanici su spominjali Facebook i Google kao željene načine povezane prijave.

Više od polovice (17) ispitnika smatra da bi aplikacija trebala imati aktivan profil na društvenim mrežama, dok 13 ispitanika ne vidi društvene mreže kao presudan kanal za komunikaciju i interakciju. Kada bi se morali odlučiti za pojedinu društvenu mrežu, najviše (13) ih se odlučilo za Instagram, gotovo jednako (11) toliko ih se odlučilo za Facebook, a samo četiri ispitanika su glasala za Twitter.

Prema dobivenim podacima, u početnoj je pretpostavci precijenjen broj tjednog unosa utakmica te je nakon istraživanja s većom sigurnošću moguće odrediti minimalne zahtjeve poslužitelja na kojem će se izvoditi aplikacija i algoritam. Za ispitanike su parovi istog spola najčešća i najdraža vrsta utakmice odbojke na pijesku te će ostale vrste u početku biti u potpunosti zanemarene. Jedina značajka koja nije bila popularna među ispitanicima je forum, što je netipičan odgovor za dobnu skupinu ispitanika, ali je tipičan odgovor za prosječnog sportaša te dobi. Sve su druge značajke zauzele gotovo jednake važnosti, dok se najviše ističu kalendar, statistika i turniri. Najviše će cijeniti fleksibilnost korisničkog sučelja pri unosu i naknadnom pregledu rezultata, a komunikacija i interakcija su im najvažnije unutar same aplikacije, bez pretjerane integracije s drugim servisima i društvenim medijima.

Neformalno istraživanje kroz razgovor s predstavnicima ciljane skupine prije provođenja formalnog istraživanja bio je ključ za uspješne pretpostavke u izradi strategije.

## 4.2. Wireframe, workflow i korisničko sučelje

Kao novi koncept softvera i usluge, ovakva vrsta aplikacije nema već ustaljen vizualni uzorak korisničkog sučelja na kojeg su korisnici navikli. Čist i sugestivan dizajn sučelja pomoći će im u brzom učenju i prilagodbi na aplikaciju koju će koristiti. U usporedbi s poznatim web aplikacijama prema namjeni, nasličnije bi bile društvene mreže poput Facebooka. Zanimljivo je istaknuti kako je upravo Facebook u svom nastajanju bio u potpunosti novo rješenje koje nitko do tada nije znao koristiti, a danas se gotovo svaka osoba s pametnim uređajem brzo i jednostavno snalazi s njihovim sučeljima i koristi širok raspon mogućnosti i usluga mreže. Započevši s vodoravnom glavnom navigacijom na vrhu, okomitom sporednom navigacijom u lijevom rubu ekrana te glavnim i sporednim sadržajem nadesno od sporedne navigacije, Facebook je postavio svojevrstan standard i koncept vodoravnog redoslijeda i „top-down“ obavljanja radnji i snalaženja unutar sadržaja i funkcija aplikacije.

Slika 4.3 Korisnička sučelja prvih verzija društvene mreže Facebook

Svi zadaci, odnosno namjere korisnika poput unošenja nove utakmice ili provjere najnovijih rezultata moraju biti procesno povezani i nalaziti se na kratkim vizualnim udaljenostima i udaljenostima po koracima. Za fiksni je vizualni raspored naše aplikacije ključno da bude fleksibilan i lako razumljiv. Pritom na umu moramo imati pojedine scenarije u kojima će se korisnici nalaziti, i to na različitim vrstama uređaja, odnosno veličinama ekrana. Stoga ćemo pobrojati sve funkcije koje planiramo grafički prikazati kako bismo ih mogli grupirati u smislene cjeline pomoću sekcija.

Da bi se prijavio u sustav, korisnik će morati biti prethodno registriran i unijeti e-mail adresu i lozinku. Pri netočnom unosu podataka ostat će na istom ekranu i dobiti poruku upozorenja, a ako se uspješno prijavi, pozdravit ćemo ga osobnom porukom i prikazati glavnu stranicu odnosno kontrolnu ploču (engl. *dashboard*). Na kontrolnoj će se ploči isticati stanje novih unesenih utakmica koje se izravno tiču prijavljenog igrača. Ispod njih će biti prikazan kratki pregled o osobnim utakmicama u posljednjem mjesecu i godini kao i informacije o najboljim partnerima i članovima kluba te neke općenite informacije o sustavu poput broja unesenih utakmica. Novosti će se nalaziti na dnu ekrana, ali mogle bi pratiti korisnika i na ostalim stranicama ukoliko ih nije otvorio, pogotovo ako su od velikog značaja.

Kad korisnik poželi unijeti rezultate novoodigrane utakmice, ispunit će kratki formular u kojem će navesti koji su igrači odigrali utakmicu i tko je pobijedio. Pri unosu je moguće unijeti utakmice koje je odigrao prijavljen igrač, ali moguće je i unijeti informacije o utakmici potpuno drugih igrača. U potonjem će slučaju barem po jedan igrač iz oba tima morati vrednovati autentičnost utakmice na način da potvrdi ili odbije unos koji se tiče njihovog tima. Nepotvrđeni unosi će nakon tjedan dana nestati iz sustava.

Najizravnija osobna poveznica igrača s aplikacijom bit će njegova osobna stranica odnosno profil. U njemu će sportaši moći vidjeti osnovne podatke o sebi poput četveroznamenkastog broja odnosno svoje igračke oznake, klub kojem pripada i lokacija tog kluba, broj osobnih bodova, broj bodova najboljeg od vlastitih timova, vlastita pozicija i pozicija najboljeg vlastitog tima u poretku unutar kluba i svjetskom poretku, ukupni broj svih dosad odigranih utakmica, postotak pobjede i slične statistike sustava izravno povezane s igračem. Na istoj će kartici biti navedeni svi parovi u kojima je dosad igrao utakmice kao i njihovi bodovni rezultati te popis svih utakmica koje je igrač odigrao. Uz te osnovne radnje, igrači će poželjeti vidjeti statistike partnera i protivnika kroz njihove profile te isključivo unutar klubova, što će moći istražiti u ostalim, odvojenim dijelova aplikacije.

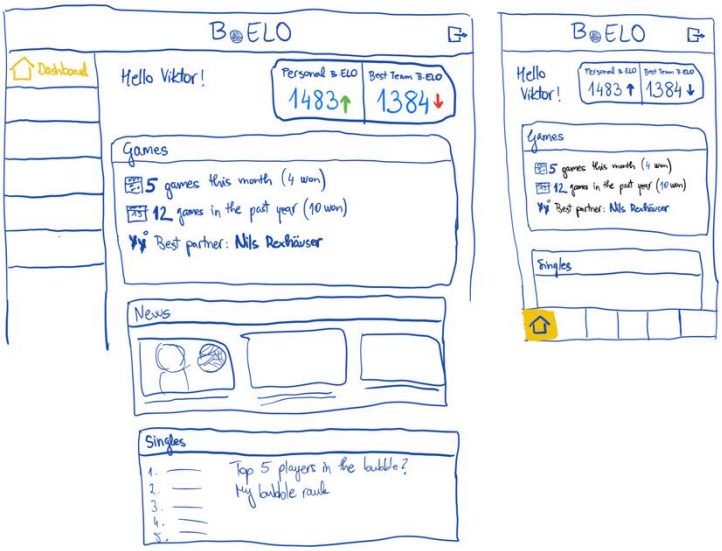
Pri dizajnu navigacije moramo imati na umu da uvijek jasno ističemo položaj unutar aplikacije. Držeći se isključivo glavne navigacije u početku, dodatno ćemo olakšati složenost organizacije informacija u sustavu. Kroz nju se prožima hijerahijski pristup strukture stabla na čijem je vrhu aplikacija i glavne stranice koje su „roditelji“ sekcija koje se u njima nalaze.[16] Valja predvidjeti poveznice između informacija na svakoj stranici, razmišljajući koje bi dodatne informacije igrači htjeli vidjeti neposredno nakon onih trenutno vidljivih i tiču li se iste teme, te hoće li detaljno istraživati sve informacije na aktivnoj kartici ili će brzo htjeti dalje se nastaviti kretati unutar aplikacije.[17]

U stadiju razvoja kostura (engl. *skeleton*) aplikacije, grupirat ćemo sve navedene informacije u sekcije. Na taj će način svaka stranica u navigaciji imati odvojene kartice s povezanim naslovima i sadržajem. Primjerice, na kontrolnoj će ploči igrači potvrđivati utakmice (kartica „Confirm matches“), pratiti vlastitu statistiku (kartica „Overview“), imati brzi pristup poretku unutar kluba (kartica „Best of the Bubble“) i čitati novosti (kartica „News“). Budući da u početku razvoja svaka od kartica neće biti prepuna podataka, kartice neće biti sklopive (engl. *collapsible*). Kroz daljnji će razvoj aplikacije biti praktično imati kartice kao glavne elemente sustava upravo radi brzog prikaza i skrivanja nepotrebnih informacija za korisnika.

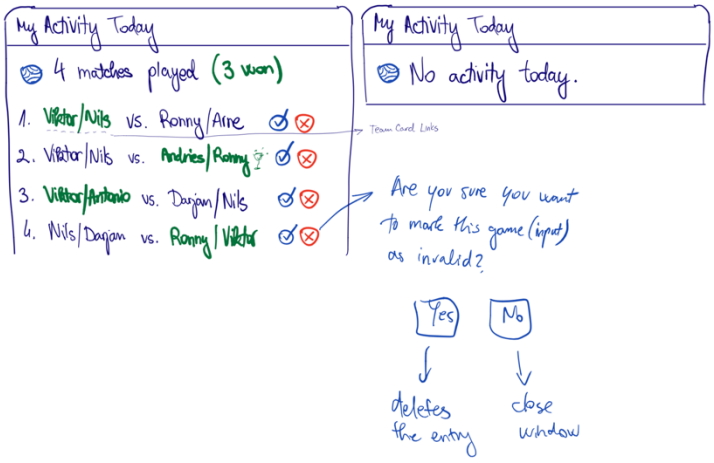
Poželjno je započeti detaljniju razradu pojedinih pregleda u unutrašnjosti aplikacije jer će ti uzorci biti najčešće korišteni i biti od najveće informacijske važnosti. Držeći se istog vizualnog uzorka bez izmjene položaja elemenata uspostavit ćemo osnovne odnose

[16] Garrett, J. J. The Elements of User Experience. Berkeley: New Riders, 2011., p. 93.
[17] Spencer, D. A Practical Guide to Information Architecture. Penarth: Five Simple Steps, 2010., p. 283.

interakcija i očekivanja.[18] Izbjegavajući zatrpavanje dodatnim prozorima i stranicama i ne odmičući se od takvog glavnog uzorka, nećemo zbunjivati korisnika i na taj ćemo način omogućiti bolje snalaženje.




Slika 4.4 Prototip niske vjernosti (low-fidelity) – navigacija i sekcije




Slika 4.5 Prototip niske vjernosti (low-fidelity) – sekcija potvrde autentičnosti utakmica

Prvim skicama postavljamo temeljne odnose između sadržaja i njegovog položaja u aplikaciji. Zatim na temelju razrađenih ideja stvaramo prototip srednje vjernosti (engl. *medium-fidelity wireframe prototype*) koji će nam poslužiti za otkrivanje najboljih rješenja najkraćim i najbržim mogućim putem, odnosno metodama. Izbjegavajući boje, crteže, fotografije i ostali vizualni sadržaj koji u ovoj fazi razvoja predstavlja nered i šum,

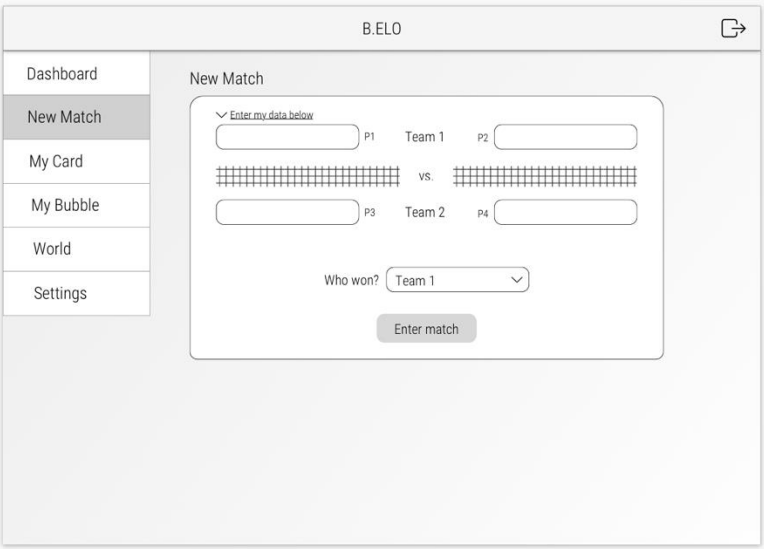
[18] Lynch, P. J., Horton, S. Web Style Guide: Foundations of User Experience Design. New Haven: Yale University Press, 2016., pp. 478-482.

usredotočit ćemo se na ispravno postavljene informacije u sustavu. Time smanjujemo utrošak vremena pri razvoju prvih ideja i ostavljamo prostor za naknadne odluke tek kada potvrdimo da smo zadovoljni s prethodnim korakom razvoja.

Na upravljačkoj je ploči zaglavlje prvi i jedini element koji se odmiče od strukture sekcija kako bi istaknuo najvažniji podatak za igrače: njihove bodovne rezultate.

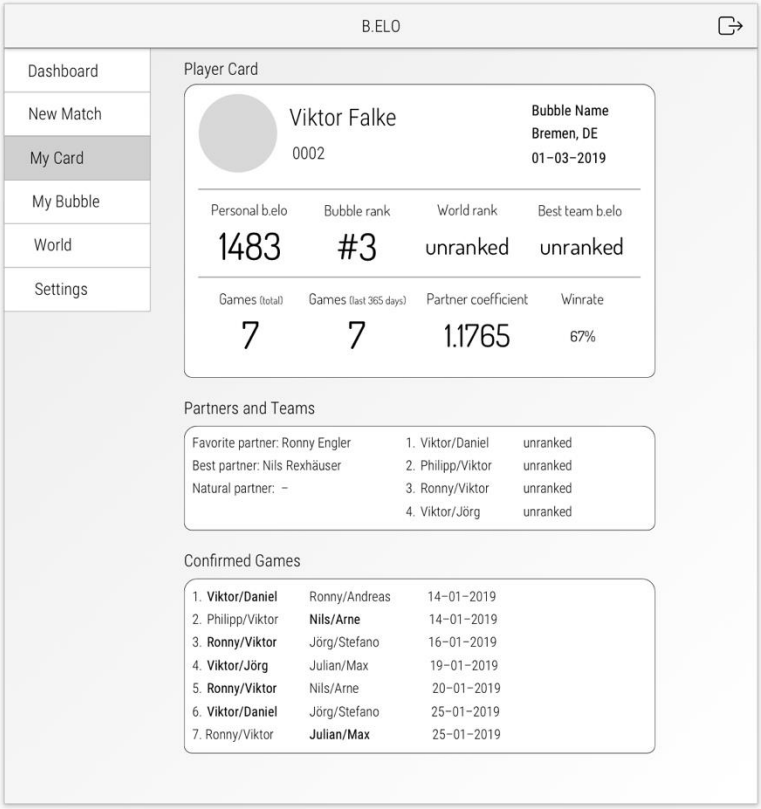
Slika 4.6 Prototip srednje vjernosti (medium-fidelity) – dashboard

Pri unosu nove utakmice jasno su odvojena polja za unos igrača, prateći vizualni izgled terena.




Slika 4.7 Prototip srednje vjernosti (medium-fidelity) – nova utakmica

Na stranici igračevog osobnog profila najviše su istaknuti glavni statistički podaci, dok se detaljnije informacije o timovima zajedno s popisom utakmica nalaze niže unutar stranice.




Slika 4.8 Prototip srednje vjernosti (medium-fidelity) – korisnički profil

Minimalnim fizičkim pravokutnicima kao sekcijama na sličnoj pozadini u široj perspektivi jasno se omogućuju neprimjetni i uniformni prijelazi između sadržaja. Na taj način umanjujemo mogućnost gubljenja konteksta nepotrebnim dodatnim odjeljivanjima.

## 4.3. Razvoj vizualnog sadržaja i izgleda

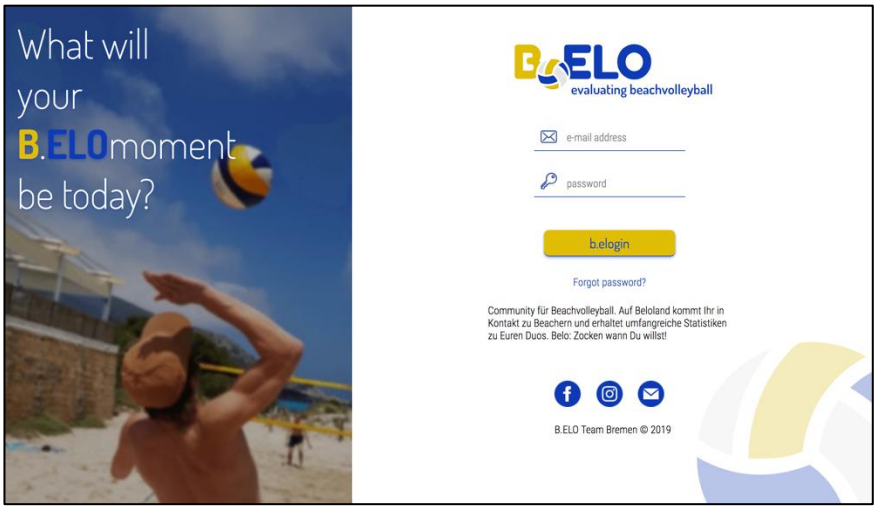
Konačnim definiranjem površinske (engl. *surface*) razine aplikacije uključujemo i pratimo B.ELO *branding* kojeg smo razradili ranije. Najvažniji je zadatak te razine upravo sugestivna potpora pisanim informacijama. U njoj ćemo kroz senzorne podražaje podsvjesno upućivati korisnika u ukupni smisao cjelina.

Glavne boje i dalje prate one karakteristične za odbojkaške lopte. Crna i bijela će biti osnovne, dok će plava boja konvencionalno označavati poveznice i interaktivne elemente. Žuta će boja služiti kao oznaka za *hover* ili kao potpora aktivnom stanju elemenata. Zelena i crvena boja će biti isključivo periferne boje kojima ćemo uobičajeno označavati pozitivan ishod ili uspjeh, odnosno negativan ishod ili greške.

Ikone će imati važnu ulogu u senzornom dizajnu. Njihova plošnost i jednostavnost linija podupirat će strukturni minimalizam. One će nam uštedjeti prostor kako ne bismo morali nepotrebno objašnjavati elemente navigacije u mobilnoj verziji, ali i pojedine dijelove prožete kroz cijelo sučelje. Što više slijedimo uobičajene simbole, brže će biti raspoznavanje ideje i svrhe iza ikone. Najbolji je test za svaku pojedinu ikonu u sučelju da korisnik razumije što ona predstavlja bez njene oznake ili podupirućeg teksta. U konačnici, estetska privlačnost posebno izrađenih ilustracija za ikone dodat će novu vrijednost aplikaciji.

Slika 4.9 Ilustracije najvažnijih ikona unutar sustava

Fotografije u aplikaciji neće biti korištene, osim na početnom ekranu prijave u sustav gdje ćemo kroz emocionalne apele i početne impresije poticati sportski duh i ponovni posjet terenu i aplikaciji. Jedine fotografije unutar aplikacije bit će korisničke slike profila.

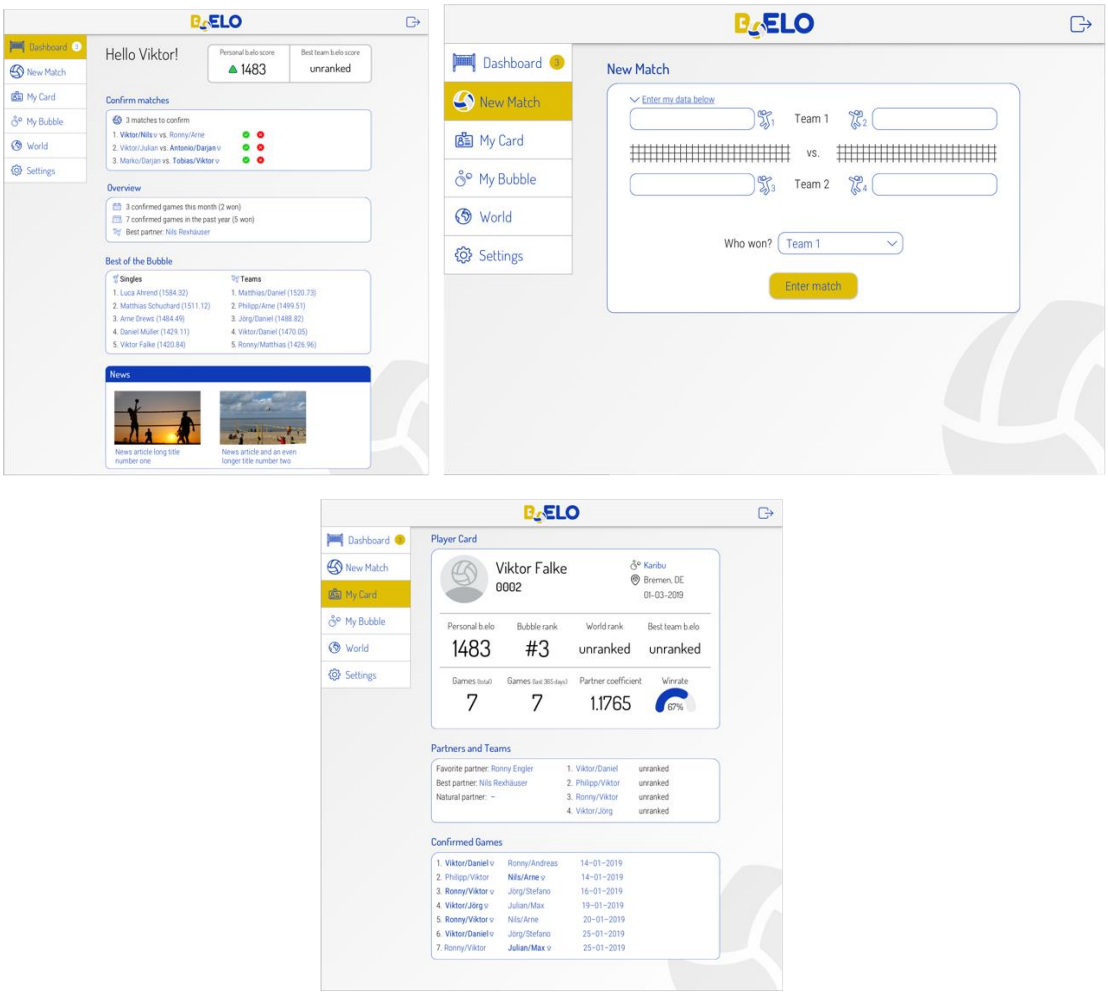



Slika 4.10 Početni ekran prijave u sustav

## 4.4. Prijedlog izgleda sučelja i analiza funkcionalnosti

### 4.4.1. Polazni prijedlog

Na temelju dosadašnjih promišljanja i odluka stvoreni su sljedeći vizualni koncepti prototipa pomoću kojih će se testirati prva verzija aplikacije.




Slika 4.11 Prototipovi visoke vrijednosti (high-fidelity) – polazni prijedlog

### 4.4.2. Testiranje upotrebljivosti prema polaznom prijedlogu

Testiranje upotrebljivosti proizvoda dio je intervjua s korisnicima u kojem ćemo koristiti prototip gotovo dovršenog proizvoda prije našeg konačnog programiranja i izrade aplikacije. Korištenjem prototipa će rješavati određene „zadatke“ koje ćemo im pripremiti i postaviti, čime ćemo saznati u kojoj je mjeri aplikacija prilagođena njihovim navikama i potrebama, ali i koliko je usklađena u vlastitim namjerama i odlukama o dizajnu. Tijekom tog procesa nastojat ćemo otkriti prepreke koje će ih sprječavati u rješavanju tih zadataka te logiku razmišljanja i podudarnost s organizacijom informacija u našem proizvodu.[19]

Kroz dobro pripremljeno istraživanje, već pet korisnika će otkriti gotovo sve mane i predložiti izmjene koje će odgovarati većini ljudi. Stoga je istraživanje provedeno s četiri

[19] Krug, S. Rocket Surgery Made Easy. Berkeley: New Riders, 2010., pp. 36-37.

predstavnika iz ciljane skupine (odbojkašima na pijesku) i jednom osobom koja nema izravnu poveznicu s ciljanom skupinom.

Prvi vizualni podražaj u testiranju bit će prazno sučelje *dashboarda* uz potpuni prikaz navigacije s lijeve strane. Uz njega će slijediti i prvo pitanje o sadržaju i podacima koje bi očekivali vidjeti upravo na glavnoj stranici aplikacije. Na taj ćemo način saznati više o njihovom ustaljenom načinu korištenja i prethodnim iskustvima. Nakon toga ćemo im prikazati jedan po jedan ekran i postavljati uvodna pitanja za razgovor i zadatke poput:

1. Za početak, molim te pogledaj ovu stranicu i odgovori mi kako ti se čini, što ti prvo privlači pažnju, što misliš da možeš ovdje napraviti, čemu služi…

2. Ako bi kliknuo samo na jedno mjesto, gdje bi prvo kliknuo i zašto?

3. Što očekuješ vidjeti ili dogoditi pri kliku na…

4. Kako bi unio novu utakmicu u sustav?

5. Kako bi postavio novu sliku na svome profilu?

6. Kako bi pozvao novog igrača u aplikaciju?

7. Gdje očekuješ vidjeti informacije o svim utakmicama koje si odigrao?

8. Kako bi pronašao profil drugog igrača? i sl.

Većina je očekivanja bez pregleda informacija na kontrolnoj ploči ispunjena. Korisnici žele vidjeti najnovije podatke u sustavu i one podatke koji se tiču njih samih. Izbor riječi pri naslovima sekcija ne upućuje na činjenicu očekuje li se određena akcija od korisnika ili ne. Jedan od primjera je prvi naslov u sekciji potvrđivanja utakmica, gdje naslov „Confirm matches“ (potvrdi utakmice) nije jednako jasan kao i „Open matches to confirm“ (otvorene utakmice za potvrdu). Svi korisnici žele prvo kliknuti na svoj profil i istražiti što će sve biti praćeno algoritmom te kako njihov unos utakmica utječe na isto.

Postoji niz oznaka i naslova ispisanih plavim slovima koji ne predstavljaju poveznicu i nisu klikabilni, što je nužno odvojiti od klikabilnog sadržaja. To nam ukazuje da važan element boja nije bio potpuno usklađen od početka, što ćemo imati na umu pri kasnijoj razradi.

Formulari predstavljaju specifičan izazov u dizajnu gdje je lako moguće iznijeti pretpostavke o terminologiji i kontekstu koje su radikalno drugačije od onoga što korisnici očekuju.[20] Tako je nagli i u stvarnome životu nepostojeći vizualni prekid odbojkaške mreže da bi se istaknula riječ „vs.“ (protiv) utjecao na pomutnje koje su se dogodile pri unosu podataka. Kako je formular za unos utakmice jedna od najkritičnijih točaka za sustav, taj dio mora biti najjasniji. Ključno je izbjeći takav dizajn u svrhu suvišnih dodatnih informacija koje su same po sebi jasne, ne narušavajući formu objekata koji su od velike važnosti. Jedina je iznimka oznaka „Unesi moje podatke niže“ koja je jasna svim korisnicima te su zadovoljni što im je olakšan unos najčešće informacije.

Podatak „Partner coefficient“ je novi matematički koncept unutar algoritma za koji je nužno uspostaviti značenje drugačijim prikazom. Tako će se uspostaviti odnosi i usporedbe unutar sustava bez potrebe za dodatnim objašnjenjima. Dio koji je izostavljen u početku, a većina korisnika je istaknula kao bitan jest vizualni prikaz povijesti napretka rezultata, pa će taj segment biti uključen među prvima od sljedećih ažuriranja aplikacije.

### 4.4.3. Prijedlog nakon unapređenja

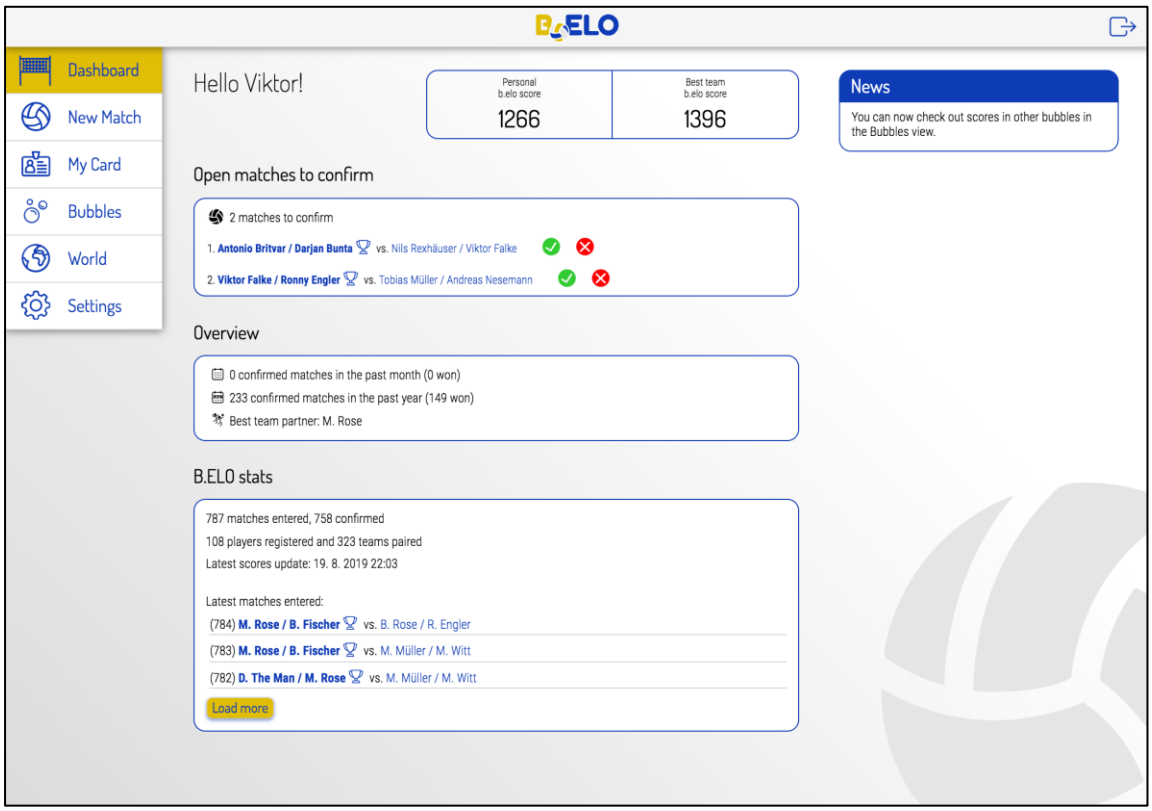
Statistike o klubu su izmještene s kontrolne ploče i zamijenjene najnovijim utakmicama unutar sustava. U cjelokupnoj strukturi prikaza, sama imena su dopunjena prezimenima, a klikom na partnere bit će prikazana timska kartica s prikazom osnovnih podataka o timu.

Jedine dodatne informacije koje bi se trebale nalaziti uz sami unos utakmice su neposredne povratne informacije o uspješnom unosu koji je u tijeku i za kojeg je nužno dobiti potvrdu od strane ostalih igrača kroz njihove profile. To je smješteno uz formular prvenstveno radi trenutne reakcije na unos koja kao signal korisniku jasno označava uspješan završetak interakcije u tom trenutku.[21]

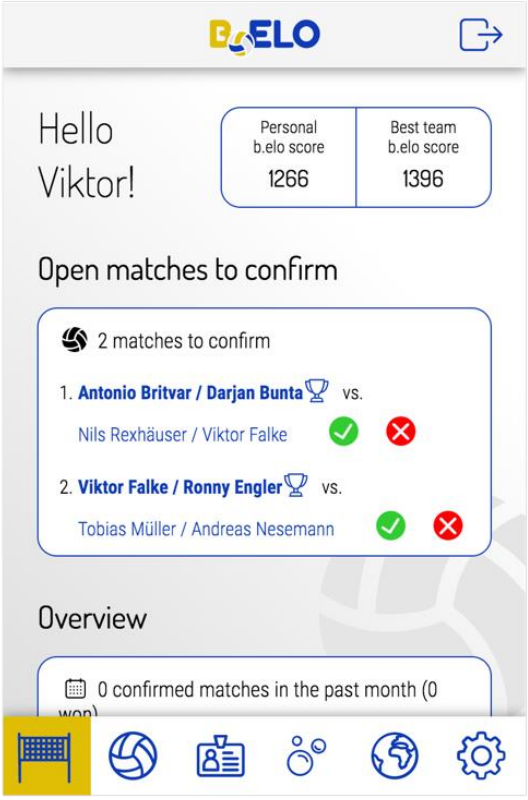
Time je zaključen konačni izgled prve verzije aplikacije za korisnike.

---

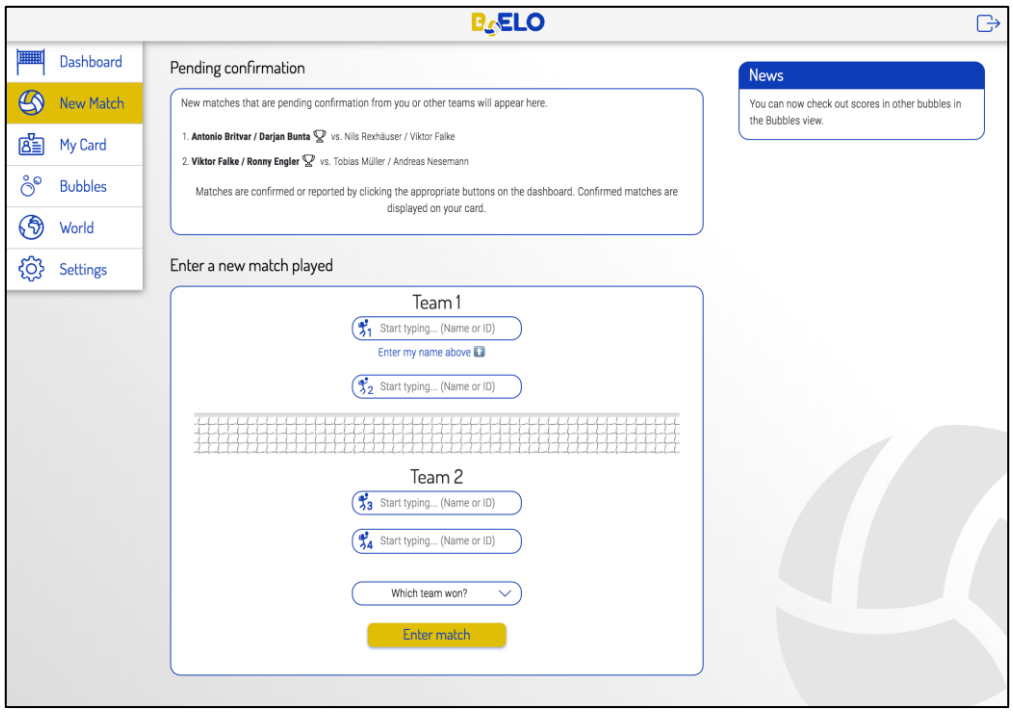
[20] Tidwell, J. Designing Interfaces: Patterns for Effective Interaction Design. Sebastopol: O'Reilly Media, 2010., p. 343.
[21] Cooper, A., Reimann, R. et al., About Face: The Essentials of Interaction Design. Indianapolis: Wiley, 2014., pp. 422-426.



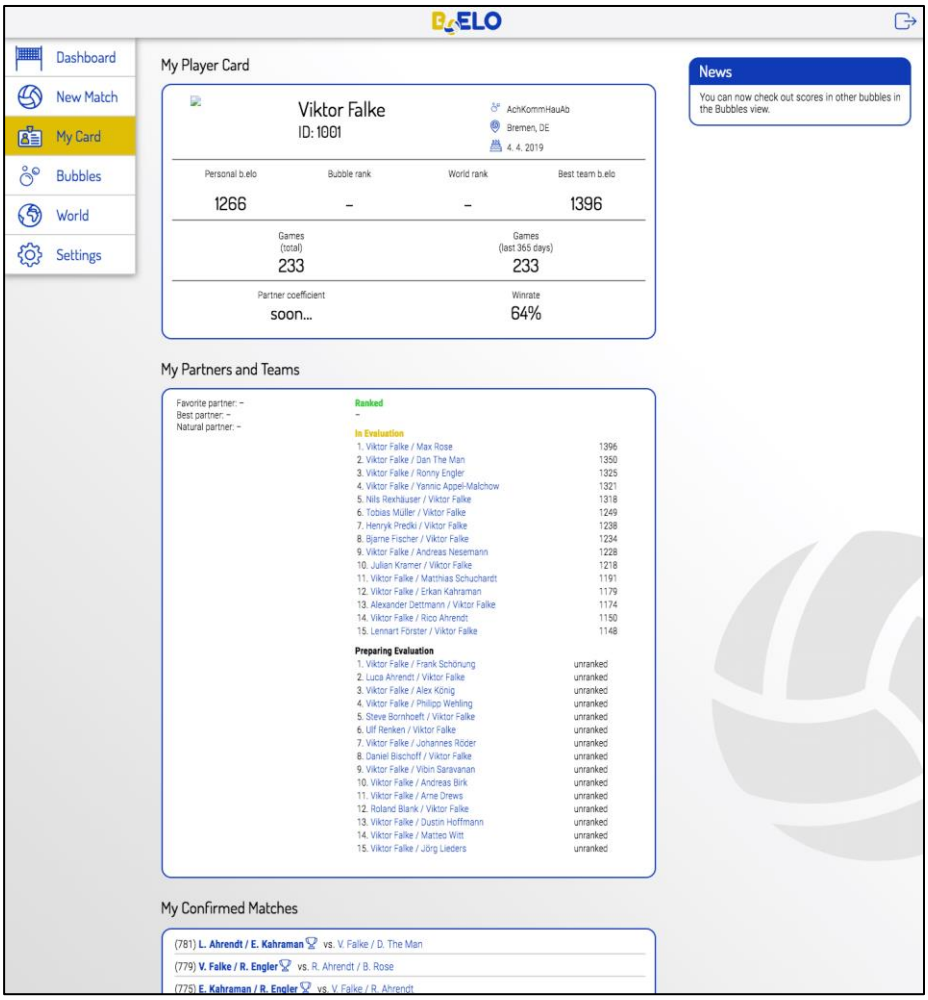

Slika 4.12 Konačni izgled aplikacije – dashboard – desktop



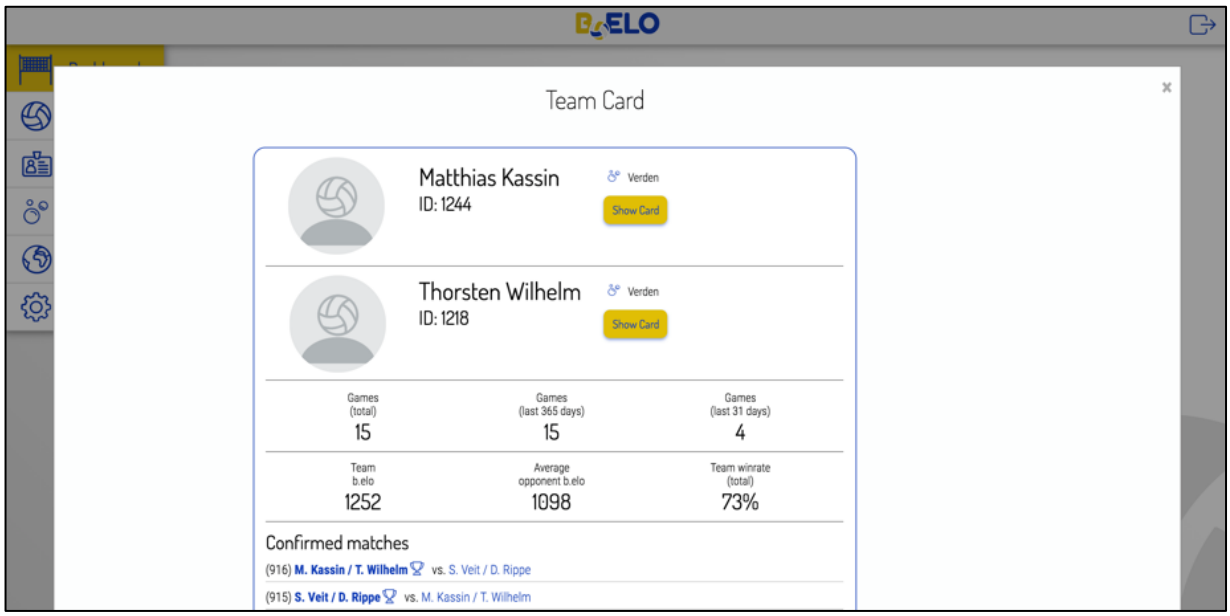

Slika 4.13 Konačni izgled aplikacije – dashboard – mobile



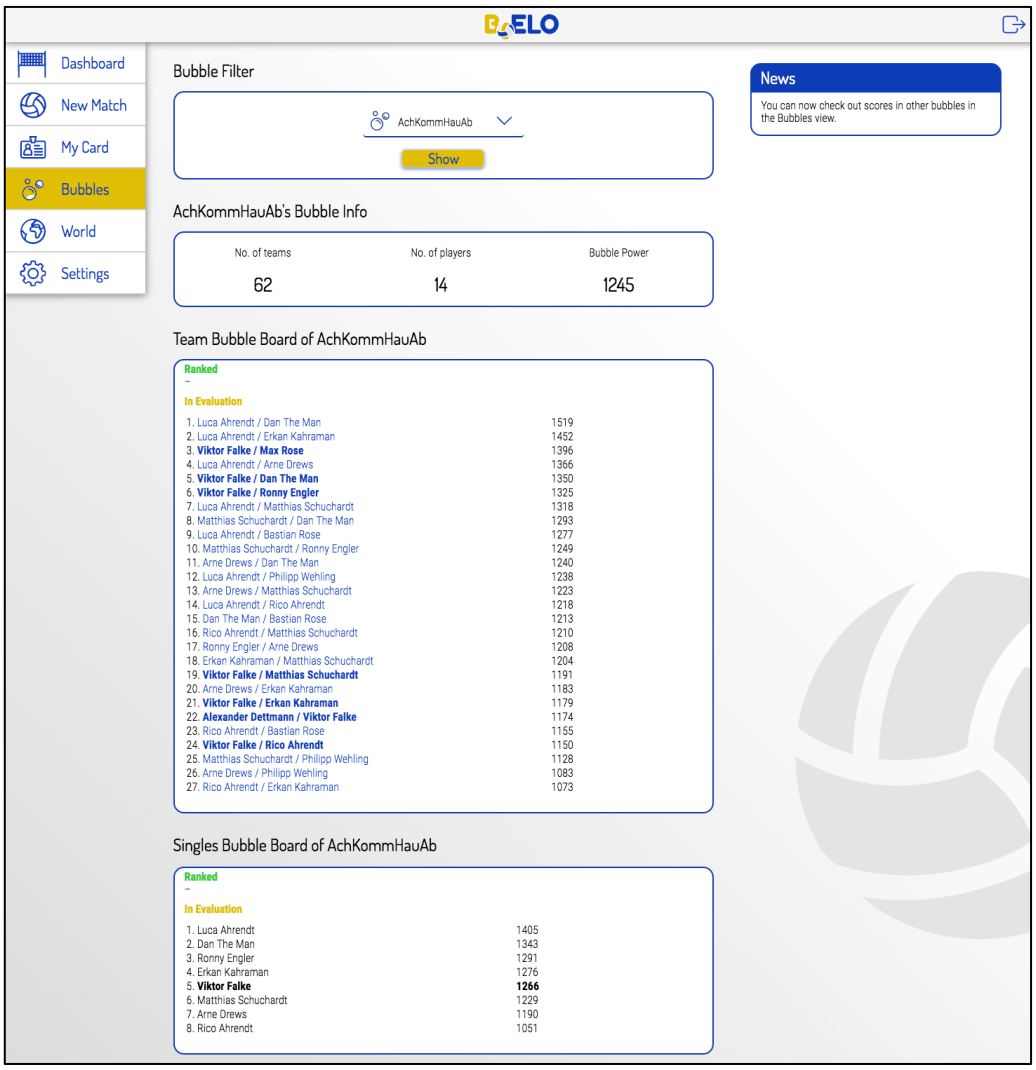

Slika 4.14 Konačni izgled aplikacije – nova utakmica



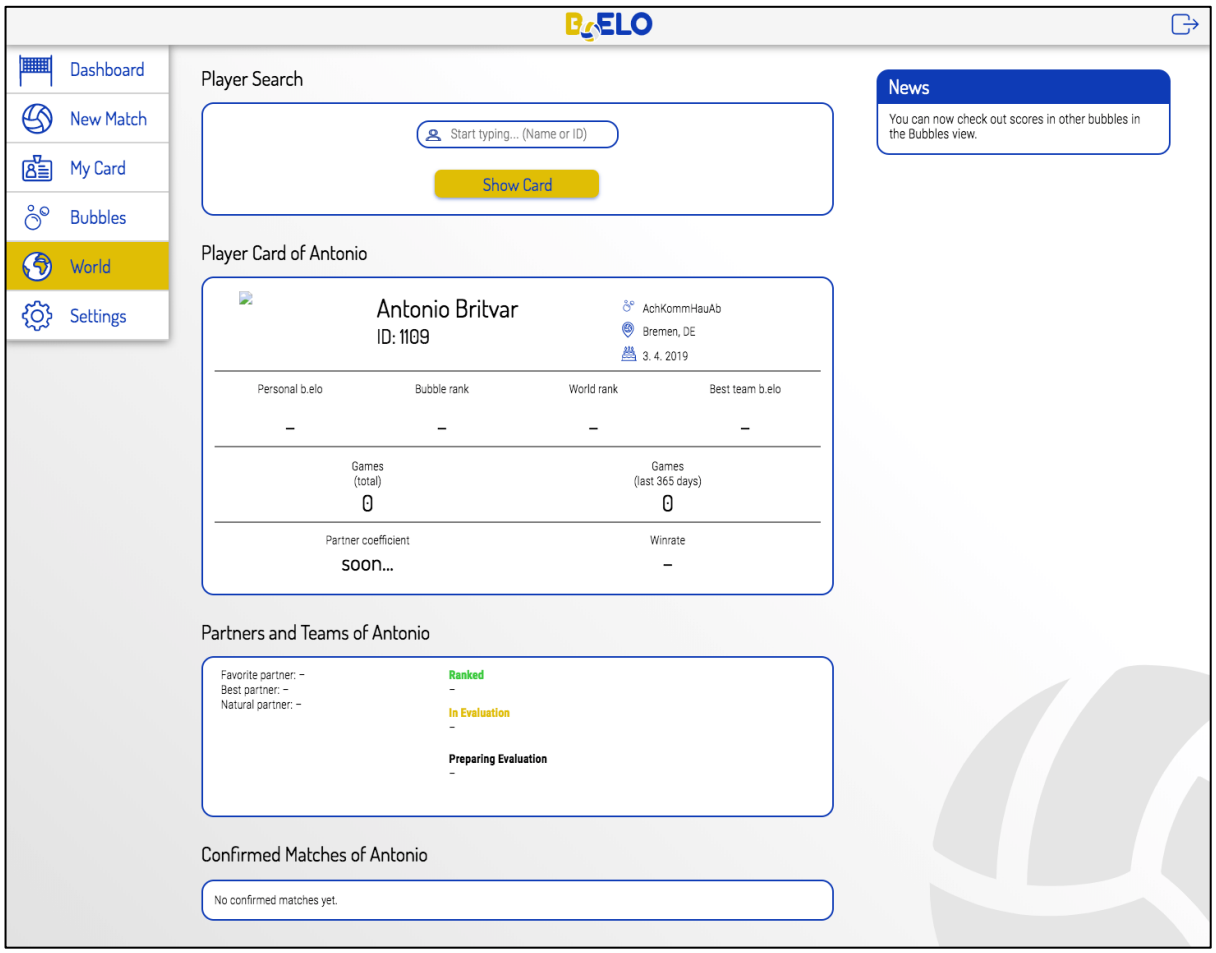

Slika 4.15 Konačni izgled aplikacije – moj profil



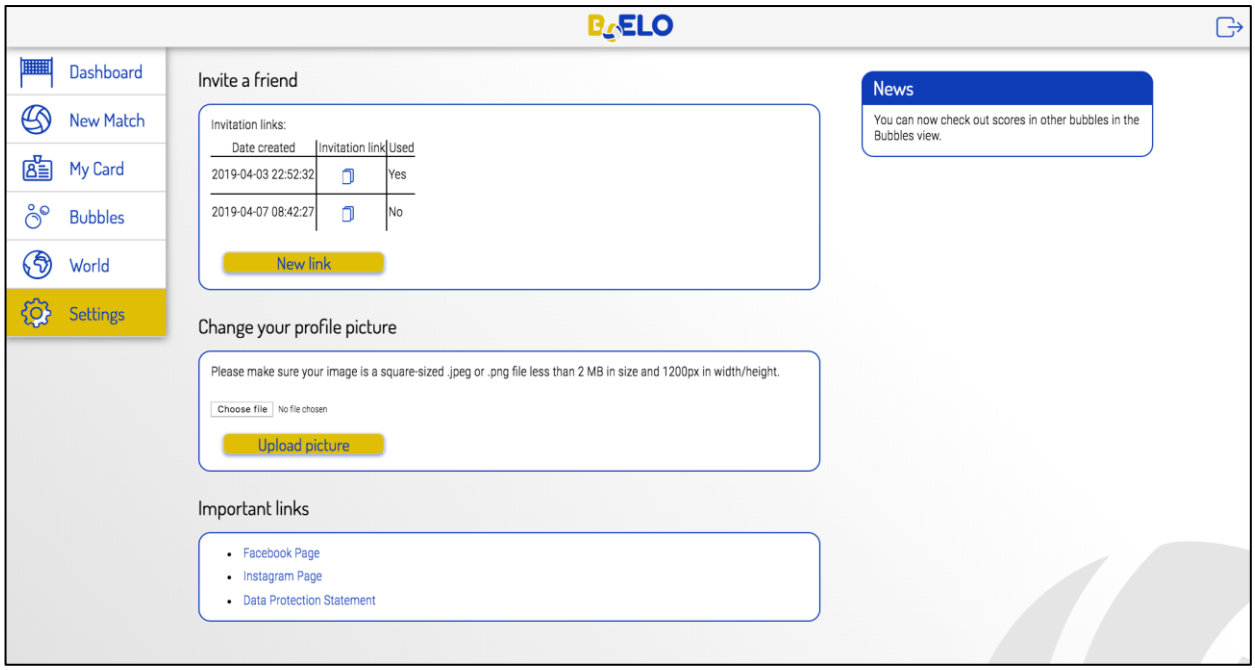

Slika 4.16 Konačni izgled aplikacije – overlay nove timske kartice




Slika 4.17 Konačni izgled aplikacije – klubovi




Slika 4.18 Konačni izgled aplikacije – svijet




Slika 4.19 Konačni izgled aplikacije – postavke

# 5. Razvoj aplikacije

Konačni je korak prema vizualnim predlošcima izraditi sustav, odnosno razviti funkcionalno sučelje gdje će se korisnici moći identificirati i služiti opisanom platformom kroz svoje račune. Takav je sustav moguće razviti koristeći u potpunosti vlastiti programski kôd. Osim što bi to odužilo i otežalo razvoj, bilo bi nužno ponovno pisati mnoge standardne dijelove aplikacije poput sustava registracije i prijave, middlewarea, gateova, automatiziranog slanja elektroničke pošte i sličnih akcija. Mnogo je učinkovitije takvu aplikaciju razviti koristeći gotovu programsku okosnicu (engl. *framework*) za PHP pomoću koje ćemo moći iskoristiti upravo te segmente, ali i neke dijelove kôda iz algoritma kojeg smo već razvili u PHP-u.

Koristeći framework u razvoju, ubrzat ćemo proces pisanja kôda za aplikaciju, struktura kôda će biti čišća i jasnija, sigurnost pri prijavi i korištenju će biti na globalno priznatoj i korištenoj razini, omogućit ćemo jednostavnost suradnje pri timskom razvoju kôda u budućnosti te pojednostavniti praćenje i održavanje sustava. Pored tih prednosti, najvažniji je nedostatak činjenica da bi framework koji djeluje poput sloja povrh PHP-a mogao utjecati na brzinu izvođenja kôda, ali već od početka ćemo to nadomjestiti izvođenjem aplikacije na poslužitelju prikladnih performansi.

U odabiru frameworka koji je besplatan i otvorenog izvora (engl. *open-source*) najvažniju ulogu imaju stabilnost i ažurnost, jednostavno i brzo postavljanje gotovog proizvoda (engl. *deployment*), ali i široka prihvaćenost u zajednici. Upravo taj najpoželjniji odabir, a pogotovo za potrebe ove aplikacije je Laravel.

Laravel koristi klasični Model-View-Controller obrazac softverske arhitekture u kojem se logika sučelja razdvaja na ta tri međusobno povezana segmenta: model je skup podataka koje dohvaćamo iz baze, pogled je dio s kojim korisnik vrši interakciju, dok upravitelj sadrži logiku obrade korisnikove interakcije. Taj će nam standard olakšati u prilagodbi aplikacije prema prototipu algoritma. Dodatni alati koje razvija njihov tim (engl. *first-party tools*) omogućuju dobru potporu pri lansiranju, održavanju, te primanju plaćanja i pretplata. Dvije značajke Laravela koje će najviše ubrzati razvoj aplikacije su Eloquent i Query Builder. Pomoću njih ćemo pri pisanju strukture upravitelja brzo i jednostavno stvarati, dohvaćati i upravljati objektima iz baze podataka. Osim što su jednostavniji od pisanja upita u SQL-u, imaju dobro razvijenu zaštitu od specifične vrste napada, tzv. *injection*.

Forge je Laravelova platforma za upravljanje poslužiteljima koju ćemo iskoristiti za postavljanje aplikacije na naš zakupljeni web poslužitelj kad ju napišemo i testiramo lokalno. Osim što ćemo moći pratiti status našeg poslužitelja, u njoj ćemo upravljati postavkama PHP-a i okruženjem, dok ćemo verzije koda osvježavati pomoću sustava po imenu Git (GitHub) koji će povezivati naše razvojno okruženje i poslužitelj. Poslužit će nam i za distribuciju i usklađenost pri pisanju i izmjenama kôda.

Kao što je planirano, u početku će sustav morati biti zatvoren za javnost. Novi se korisnici neće moći registrirati dok ne dobiju osobni pozivni kôd kojeg stvaraju postojeći korisnici unutar aplikacije. Tako se vanjski dio sastoji isključivo od prijave s početne stranice i registracije ukoliko je priložen ispravan i neiskorišten pozivni kôd, a unutarnji dio kojem se pristupa nakon uspješne autentifikacije sastojat će se od planiranih ekrana. Njemu će korisnici moći pristupiti tek nakon što potvrde svoju adresu e-pošte.

## 5.1. Sustav registracije, prijave i pohrane podataka

Laravel samostalno kreira stranice za prijavu i registraciju korisnika koje je nužno preinačiti na planirani izgled i funkcionalnost. Registraciju korisnika bez pozivnice spriječit ćemo na način da pri posjetu stranici /register mora biti zadan GET parametar i to s vrijednošću jednog od generiranih neiskorištenih kôdova. U suprotnom će se prikazati stranica s porukom greške i kratkom uputom kako je potrebno kliknuti na punu poveznicu u slučaju da su korisnici s pozivnicom kopirali nepotpunu adresu web stranice u svoj preglednik. U upravljaču registracija dio koji pokreće analizu i provjeru podataka za stvaranje korisnika imat će dio:

```
$invitationcode = $data["invitationcode"];
$statusCount = DB::table('userinvitation')->where('InvitationCode', $invitationcode)->where('Used', 0)->count();
```

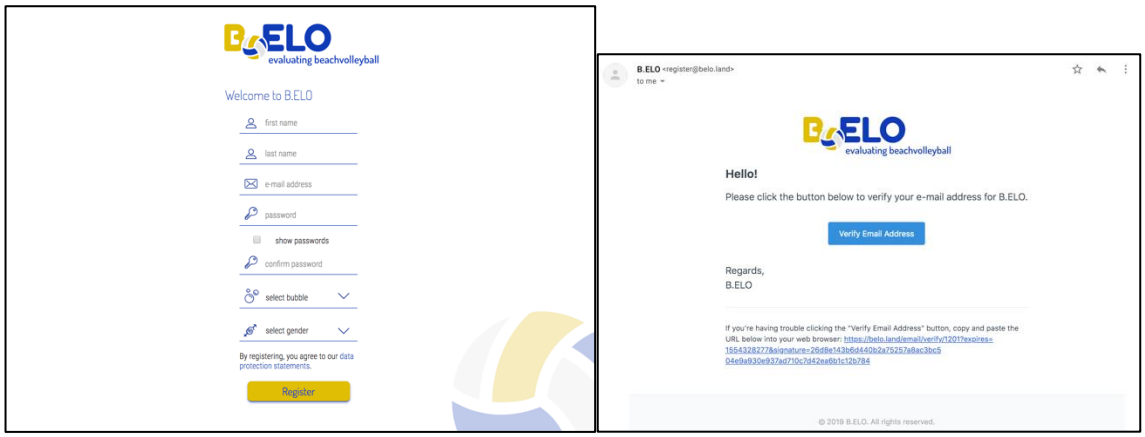
Kôd 5.1 Provjera pozivnog kôda i stvaranje novog korisnika – dio 1/2

gdje kroz varijablu `$statusCount` provjeravamo postoji li generiran kôd koji nije iskorišten, što je i primjer korištenja izražajnog i jednostavnog Query Buildera i objekta DB čiju je logiku u jednoj liniji koda iznimno lako iščitati. U prvoj je varijabli sadržana vrijednost kôda iz GET parametra koja je proslijeđena upravljaču metodom POST. U nastavku ćemo korisnika odbiti ukoliko kôd nije ispravan ili mu stvoriti novi račun:

```
  if($statusCount == 0) {
  dd('The invitation code you provided is either invalid or
expired.');
  } else {
        $user = User::create([
            'firstname' => $data['firstname'],
            'lastname' => $data['lastname'],
            …
            'invitationcode' => $data['invitationcode'],
        ]);

        $userBelo = UserBelo::create([
          'UserID' => $user->id,
        ]);

        $userBelo->save();
        $newStatus = UserInvitation::where('InvitationCode',
$invitationcode)->update(['Used' => 1]);
```

Kôd 5.2 Provjera pozivnog kôda i stvaranje novog korisnika – dio 2/2

Ako program upravljača nastavi s izvođenjem, stvorit će korisnika kroz objekt `User` i metodu `create()`. Kroz model i objekt `UserBelo` stvorit ćemo novi predefinirani početni zapis bodova za upravo stvorenog korisnika metodom `save()`. Na kraju ćemo zabilježiti i da je navedeni `UserInvitation` kôd iskorišten. To su primjeri korištenja Laravelovog Eloquent modela. Nadalje, upravljač će pokrenuti slanje e-pošte s poveznicom za validaciju korisničkog računa.




Slika 5.1 Prikaz registracije i validacije računa kroz e-poštu

Pri generiranju pozivnih kôdova u postavkama aplikacije moramo postaviti ograničenje na broj izdanih pozivnica novim korisnicima. To ćemo učiniti kroz varijablu u zapisu o korisniku unutar baze podataka gdje ćemo bilježiti najveći broj dozvoljenih pozivnica za izradu. Unutar upravljača to ćemo pratiti na sljedeći način:

```
$numberofinvitations = DB::table('userinvitation')->
where('UserIDOriginal', $UserID)->count();
$userinvitationlimit = Auth::user()->invites;
if($numberofinvitations > $userinvitationlimit) {
       $message = "You can't invite more players right now.";
} else {
    $userinvitation = UserInvitation::create([
        'UserIDOriginal' => $UserID,
        'InvitationCode' => $inviteCode,
        'Used' => 0,
    ]);  $userinvitation->save();
} return redirect('settings')
```

Kôd 5.3 Provjera ograničenja za pozivni kôd pri stvaranju istog

Za razliku od planiranih modela prototipa (slika 3.4), u glavnoj ćemo aplikaciji nužno morati dodati još neke podatke za pojedine objekte u bazi podataka, ali i potpuno nove objekte:

Tablica 5.1 Razlike u objektima baze podataka prototipa i konačne aplikacije

| **Naziv tablice** | **Razlike** |
|---|---|
| Bubble | dodaje se varijabla *City* za podatak o gradu kluba |
| Player (novo ime „Users“) | dodaju se varijable *email_verified_at* (za potvrdu validacije adrese e-pošte), *invitationcode* (za povezivanje s igračem koji je izdao pozivnicu za njega), *profpic* (za lokaciju odnosno naziv slike profila korisnika) i *invites* (kao najveći broj dopuštenih pozivnica) |
| Game (novo ime „GameTemp“) | dodaju se varijable *EntryStatusT1*, *EntryStatusT2* i *Stored* (za potrebu logike potvrde utakmica) te *EntryUser* i *UpdateUser* (za potrebu provjere unosa utakmica i poveznice s igračem koji ju je unio ili potvrdio/odbio) |
| UserInvitation (nova tablica) | sastoji se od *id*, *UserIDOriginal* (kao ID igrača koji je stvorio pozivnicu), *InvitationCode* te *Used* (kao status iskorištenosti) |
| News (nova tablica) | sastoji se od *id*, *title*, *content* te *image* |

Uz navedene promjene, Laravel u bazu podataka dodaje svoje objekte *migrations* te *password_resets* u kojima čuva podatke o promjena u bazi te zahtjevima za novo generiranje lozinke.

## 5.2. Primjena algoritma u web aplikaciji

Nova struktura, odnosno *pattern* redoslijeda izvođenja kôda zahtijevat će potpunu prilagodbu postojećeg sustava Laravelu. Prateći dosad zabilježene procese i tijek, prevodit ćemo ih u obrazac pomoću kojeg funkcionira MVC web aplikacija.

Jedna od najvažnijih datoteka aplikacije je **/routes/web.php**. Unutar nje definiramo pravce kretanja po aplikaciji kroz posjet stranicama i povezujemo ih s upravljačima. Pomoću *middlewarea* ćemo provjeravati je li korisnik autentificiran – ako nije, šaljemo ga na /login stranicu i ne dopuštamo mu pristup aplikaciji, a ako je, može posjećivati sljedeće rute (zahtijevati ih metodom GET):

- **/home** – upravljačka ploča
- **/newmatch** – novi zapis o utakmici
  (metodom POST na **/newmatch/enter** unosi se novi zapis)
- **/mycard** – vlastiti profil
- **/bubbles** – statistike klubova (metodom POST tražimo prikaz za klub s popisa)
- **/world** – pretraga profila ostalih igrača i svjetske statistike
  (metodom POST tražimo prikaz za odabranog igrača)
- **/settings** – postavke (metodom POST na **/settings/photo** i **/settings/invitationrequest** pokreću se zahtjevi za navedenim izmjenama)
- **/logout** – sigurni izlaz iz aplikacije
- rutama **/autocomplete** i **/team/{teamselected}** omogućujemo pretragu igrača i prikaz timske kartice u odvojenom oknu
- rutama **/match/{matchid}/confirm** i **/match/{matchid}/decline** upravljamo potvrđivanjem i odbijanjem utakmica, pri čemu moramo izvršiti provjeru pripada li ta utakmica prijavljenom igraču, odnosno smije li on činiti izmjene u tom zapisu

Rutu **/register** korisnik može posjetiti jedino ako nije prijavljen, kao i rute **/password/email** i **/password/reset**.

Za prikaz za većinu ruta možemo koristiti jednu datoteku glavnog rasporeda (engl. *layout*) u kojem će biti sadržano zaglavlje, navigacija, prikaz okna (engl. *modal*) i novosti te nacrtom za smještaj ostalog sadržaja specifičnog za stranicu koju korisnik posjećuje. To će nam omogućiti Laravelov Blade i njegove naredbe `@extends, @section i @yield`. Primjerice, tako možemo u datoteci newmatch.blade.php reći da ona `@extends('mainlayout', ['title' => 'New Match'])`, te da u `@section('content')` ima vlastiti sadržaj kojeg ćemo sastaviti, a u mainlayout.blade.php reći da ta datoteka ima dio u kojem je potrebno `@yield('content')` gdje ćemo ubaciti upravo taj pripadajući sadržaj ovisno o stranici koja se prikazuje. Koristeći ove jednostavne naredbe, brzo smo sastavili okosnicu vizualnog i logičkog rasporeda elemenata. Svi nacrti za preglede aplikacije nalaze se u /resources/views i /resources/views/layouts.

Nakon definiranja modela i pregleda valja opisati rad upravljača. Njihove datoteke nalaze se na lokaciji /app/Http/Controllers. Osim predefiniranih i uređenih Auth upravljača, potrebno je definirati i sljedeće:

- HomeController – glavni upravljač za upravljanje podacima za prikaz u pregledima

Unutar njega su sadržane funkcije index(), newMatch(), searchUsers(), searchTeams(), mycard(), bubbles(), world() i settings(). Zbog jednostavnosti organizacije, sve su funkcije za pripremu podataka za vizualni ispis na ekranu unutar jednog upravljača. S daljnjim razvojem bi ih radi preglednosti bilo uputno razdvojiti. Ono što povezuje upravljač i pregled je upravo već spomenuti /routes/web.php. Primjerice,

```
Route::get('/mycard','HomeController@mycard')->name('mycard')->
middleware('verified');
```

označava da će pri posjetu stranici /mycard za pripremu podataka biti zadužen upravljač HomeController, i to specifično funkcija mycard(), s time da će korisnik morati biti autentificiran prije nego što će moći pristupiti toj stranici.

- SettingsController – odvojeni upravljač za postavljanje slike profila i izdavanje novih kôdova pozivnica u sustav
- MatchController – upravljač za unos nove utakmice, provjeru postojanja tima u bazi podataka, izradu novog tima i dohvat postojećeg tima
- HomeMatchController – upravljač za upravljanje potvrdom ili odbijanjem utakmica na upravljačkoj ploči

- GameTempController – upravljač za pokretanje izračuna poretka parova i pojedinaca za nove potvrđene igre

Pored izračuna poretka, najzahtjevniji isječak programske logike ne odnosi se na unos utakmice ili postavljanje fotografije profila, već na prikaz nepotvrđenih utakmica i na samu logiku potvrđivanja ili odbijanja istih koji su sadržani u HomeMatch upravljaču. On se sastoji od dvaju funkcija; `confirmMatch($matchid)` i `declineMatch($matchid)`. Ako igrač potvrdi utakmicu klikom na zeleni gum potvrde i posjećujući /match/{matchid}/confirm pokreće se sljedeća logika:

```
$userloggedin = Auth::user()->id;
$usersTeams = DB::table('team')->where('UserID1',
$userloggedin)->orWhere('UserID2', $userloggedin)->get();
$teamIDs = array();
      foreach($usersTeams as $usersTeam) {
        $teamIDs[] = $usersTeam->id;
      }
      $gameSubmitted = GameTemp::where('id', $matchid)
                                ->first();
      $gameSubmittedT1 = $gameSubmitted->TeamID1;
      $gameSubmittedT2 = $gameSubmitted->TeamID2;
      $entryStatusT1 = $gameSubmitted->EntryStatusT1;
      $entryStatusT2 = $gameSubmitted->EntryStatusT2;
      $validityT1 = in_array($gameSubmittedT1, $teamIDs);
      $validityT2 = in_array($gameSubmittedT2, $teamIDs);

      if ($validityT1 == 1 && $entryStatusT1 == 0) {
        $gameSubmitted->EntryStatusT1 = 1;
        $gameSubmitted->save();
      } if ($validityT2 == 1 && $entryStatusT2 == 0) {
        $gameSubmitted->EntryStatusT2 = 1;
        $gameSubmitted->save();
      } if ($validityT1 == 0 && $validityT2 == 0) {
        dd('You can't modify a game you're not a part of.');
      } return redirect('home');
```

Kôd 5.4 Program za potvrđivanje utakmice na upravljačkoj ploči

Ista logika vrijedi i za odbijanje utakmice, pri čemu se vrijednost `$gameSubmitted->EntryStatusT1/T2` postavlja na -1 i kao takva sprema u bazu podataka. Važno je

provjeriti ima li korisnik pravo na izmjene za navedeni objekt i odbiti njegov zahtjev u slučaju da je neosnovan.

Značajan dodatak vanjskog izvora unutar aplikacije zapravo je javno dostupna JavaScript knjižnica (engl. *library*) imena typeahead.js koja nam pomaže u stvaranju popisa igrača iz baze podataka pri unosu u formularu za novi zapis utakmice. Važno je imati na umu da u funkciji za pretragu igrača uključimo uvjet da podatak o vremenu verifikacije adrese e-pošte ima zapis, odnosno nije prazan, kako bismo odbacili nepotvrđene igrače.

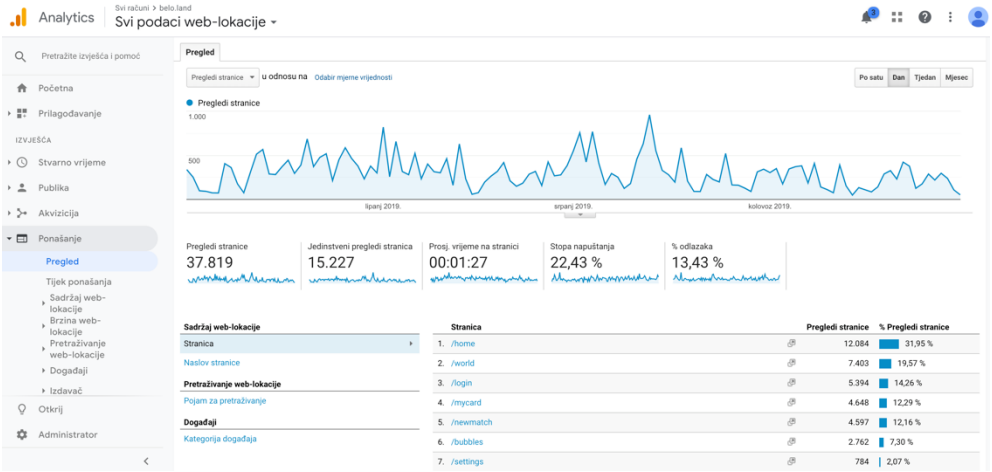
Kako se dvije glavne funkcije za izračun poretka nalaze unutar GameTemp upravljača, moguće ih je pokrenuti kroz putanju u /routes/web.php. Ni jedna ni druga ne zahtijevaju posebne, varijabilne ulazno-izlazne podatke, već svoje podatke za izračun uzimaju iz baze. Važno je jedino u obzir uzeti one utakmice gdje su EntryStatus varijable jednake 1 i gdje je Stored varijabla jednaka 0, odnosno one utakmice koje su potvrđene od oba tima, a nisu već unesene u sustav, te nakon obrade tu Stored varijablu promijeniti u 1. Tako razlikujemo četiri glavne vrste unosa utakmica u tablici GameTemp: nepotvrđene, odbijene, potvrđene te obrađene. Idealno, funkcije za izračun se izvode unutar sustava samostalno barem jednom dnevno, a moguće ih je i ručno pokrenuti, pogotovo za vrijeme turnira s mnogo utakmica. Automatizaciju je moguće izvesti pomoću Laravelovog task schedulinga i artisan naredbi.

U konačnici, na primjeru sustava s oko tisuću unesenih igara i dvjesto korisnika, Laravel dohvaća, obrađuje i pohranjuje podatke dovoljno brzo za predviđen rad aplikacije i sustava na manjem poslužitelju.

## 5.3. Dodatne mogućnosti, primjene i skaliranje

Ovaj se algoritam i struktura aplikacije najlakše mogu primijeniti u srodnim okruženjima poput sportova i aktivnosti gdje se par igrača natječe s drugim parom poput stolnog tenisa, belota, računalnih igara i sličnim primjerima gdje bi vizualne i tehničke promjene bile minimalne. Koristeći navedene principe, algoritam se može dalje razvijati za više ljudi u timu, ali najzanimljivija bi primjena bila za nefiksan broj ljudi u timu. Takva bi fleksibilnost osigurala dugoročnu primjenu, pogotovo u područjima gdje se odabirom pridaje značaj određenim izborima i na taj način precizno određuje mnijenje. Pritom je najvažnije pomno analizirati dodatne parametre koje bi trebali bilježiti, a da su relevatni za algoritam i da trebaju biti uključeni u izračun.

Zanimljivo je u postojećoj strukturi promotriti predviđeno ponašanje korisnika kroz Google Analytics platformu. Nakon tri mjeseca aktivnog rada aplikacije, daleko se najviše koriste upravljačka ploča i pretraga tuđih profila i statistika, kao i onih vlastitih. Najmanje se zadržava pri unosu nove utakmice. Prikaz kluba je objavljen samo nekoliko tjedana prije kraja tromjesečnog uzorkovanja, a ako se u obzir uzme prikaz isključivo tog perioda, upravo taj prikaz ima daleko najviše posjeta iza upravljačke ploče.



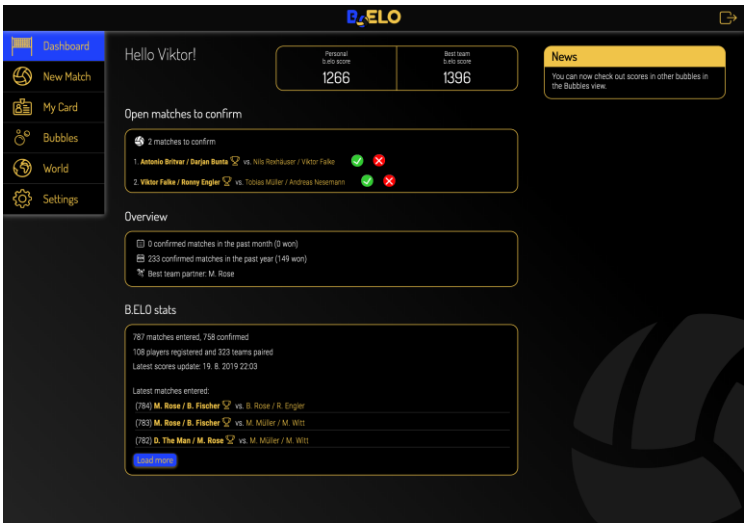

Slika 5.2 Prikaz aktivnosti korisnika unutar Google Analytics platforme

Ovi podaci prije svega potvrđuju ispravno predviđenu informacijsku arhitekturu i vizualno planiranje sučelja.

Od dodatnih funkcija za razvoj, ističe se nekoliko pitanja i novih smjerova. Za algoritam je važno postaviti pitanje koliko dugo će utakmice ostati relevantnima za izračun bodova unutar sustava te kako isplanirati prikaz nakon prvih 365 dana rada, kao i treba li najviše pažnje obraćati isključivo na aktualnu godinu. Način upravljanja najstarijim podacima bit će poseban izazov za stabilnost, brzinu i relevatnost sustava i podataka.

Posebna funkcija „izazivanja“ para na utakmicu, odnosno okršaj dodatno bi pojačala interakcije unutar sustava i interes igrača, kao i društvenu komponentu aplikacije. Planiranjem i projektiranjem integracije s ostalim platformama mogli bi produbiti interakcije kroz notifikacije o novoj unesenoj utakmici, o potvrđenim utakmicama, o promjeni stanja bodova i sličnim događajima.

Aplikaciju je nužno doraditi i vizualno, pri čemu se ističe mod za vanjski teren (engl. *outdoor court mode*) gdje bi igračima moglo biti korisno imati sučelje s visokim kontrastom i bez bijele boje ekrana zbog sunca i ostalih vizualnih smetnji.




Slika 5.3 Primjer vizualnog predloška boja u modu za teren

Najjača struktura poretka, ona svjetska, ima mnogo uvjeta poput pokazatelja razlike bodova prosječnog protivnika tima i bodova tog tima, što govori o razlici njihovih vještina. Jesu li igrači koji najbolje procjenjuju vještine svoje okoline najbolji i treba li njih najviše nagraditi? U aplikaciju je također nužno uključiti igračice, odnosno ženski spol, kao i „miješane“ parove.

# Zaključak

Sustav za izradu poretka gradio se definiranjem i rastavljanjem cjelokupnog koncepta na dijelove različite strukture. Krenuvši od najmanje jedinice odnosno jedne utakmice, nastavljajući do igračeve i protivničke prošlosti, povezujući ih i pridavajući im različite važnosti, stvorena je nova platforma koja napredno pozicionira igrače na temelju odabranih parametara. Budući da se sustav temelji na iznimno jednostavnim varijablama i podacima, nužno je postavljati sve zahtjevnije uvjete igračima za precizno rangiranje u svjetskoj strukturi. Kroz prikupljanje i analizu podataka o utakmicama u sustavu, bit će jednostavnije odrediti kvalitetnije parametre za nove verzije algoritma koje će spriječiti nepravednu dodjelu bodova. Pored tih informacija, prateći korisničko ponašanje unutar aplikacije i kroz komunikaciju putem ostalih kanala pomnije ćemo odrediti njima najzanimljive značajke za daljnji razvoj. Na živom je primjeru analizom ponašanja korisnika moguće zaključiti da su opseg i način korištenja uspješno planirani, pri čemu je veliku zaslugu nužno pripisati učestalom testiranju i iteriranju. Time je kroz jedinstveno rješenje web aplikacije realiziran i pokriven širok spektar primjene i očekivanja igrača, od kojih su najbitnije pristupačnost i brzina.

Nakon nekoliko mjeseci rada, nekoliko stotina korisnika i tisuću unosa, aplikacija i dalje nije ni približno opterećena podacima i izračunima. Ovakav princip smanjivanja tražene količine podataka i stvaranja novih relevatnih i povezanih zaključaka iz njih mogao bi se primijeniti na mnoge druge sustave. Time bi uvelike pridonio općenitom smanjenju količine i pohrane podataka, redundanciji i resursima za obradu, ali i smanjio uvjete tražene od korisnika poput prijave na turnire i detaljnije praćenje rezultata što predstavlja dugotrajan i iscrpan proces.

# Popis kratica

| | | |
|---|---|---|
| FIVB | *Fédération Internationale de Volleyball* | Međunarodna odbojkaška federacija |
| HTML | *Hypertext Markup Language* | Jezik za označavanje hiperteksta |
| CSS | *Cascading Style Sheets* | Kaskadni stilski jezik |
| JS | *JavaScript* | Skriptni programski jezik |
| PHP | *Hypertext Preprocessor* | Hipertekstualni pretprocesor |
| SQL | *Structured Query Language* | Strukturni upitni jezik |
| MVC | *Model-View-Controller* | Model-Pogled-Upravitelj |
| CSV | *Comma-separated Values* | Vrijednosti odvojene zarezima |
| URL | *Uniform Resource Locator* | Usklađeni lokator resursa (web adresa) |

# Popis slika

Slika 2.1 Odabrane boje i fontovi ..... 5

Slika 2.2 Logo branda B.ELO (autor D. Bunta) ..... 6

Slika 2.3 Branding B.ELO-a u primjeni u početku ..... 6

Slika 2.4 Prikaz količine podataka unutar algoritma prema procesima ..... 13

Slika 3.1 Glavni tijek izvođenja unosa podataka ..... 17

Slika 3.2 Prikaz strukture polja za unos igrača u novoj utakmici ..... 18

Slika 3.3 Prikaz algoritma za izračun bodova timova ..... 27

Slika 3.4 Nacrt objekata u bazi podataka prototipa i njihovih odnosa ..... 32

Slika 3.5 Vizualna struktura pregleda u prototipu ..... 33

Slika 3.6 Primjer strukture CSV datoteke za unos podataka ..... 35

Slika 3.7 Razlike u bodovima parova pri različitim načinima unosa utakmica ..... 37

Slika 4.1 Rezultati istraživanja – najvažnije značajke buduće aplikacije ..... 41

Slika 4.2 Rezultati istraživanja – stavovi o utakmicama i poretku ..... 42

Slika 4.3 Korisnička sučelja prvih verzija društvene mreže Facebook ..... 44

Slika 4.4 Prototip niske vjernosti (low-fidelity) – navigacija i sekcije ..... 46

Slika 4.5 Prototip niske vjernosti (low-fidelity) – sekcija potvrde autentičnosti utakmica. 46

Slika 4.6 Prototip srednje vjernosti (medium-fidelity) – dashboard ..... 47

Slika 4.7 Prototip srednje vjernosti (medium-fidelity) – nova utakmica ..... 47

Slika 4.8 Prototip srednje vjernosti (medium-fidelity) – korisnički profil ..... 48

Slika 4.9 Ilustracije najvažnijih ikona unutar sustava ..... 49

Slika 4.10 Početni ekran prijave u sustav ..... 49

Slika 4.11 Prototipovi visoke vrijednosti (high-fidelity) – polazni prijedlog ..... 50

Slika 4.12 Konačni izgled aplikacije – dashboard – desktop ..... 53

Slika 4.13 Konačni izgled aplikacije – dashboard – mobile ..... 53

Slika 4.14 Konačni izgled aplikacije – nova utakmica ..... 54

Slika 4.15 Konačni izgled aplikacije – moj profil ..... 54

Slika 4.16 Konačni izgled aplikacije – overlay nove timske kartice ..... 55

Slika 4.17 Konačni izgled aplikacije – klubovi ..... 55

Slika 4.18 Konačni izgled aplikacije – svijet ..... 56

Slika 4.19 Konačni izgled aplikacije – postavke ..... 56

Slika 5.1 Prikaz registracije i validacije računa kroz e-poštu ..... 59

Slika 5.2 Prikaz aktivnosti korisnika unutar Google Analytics platforme ..... 65

Slika 5.3 Primjer vizualnog predloška boja u modu za teren ..... 66

# Popis tablica

Tablica 2.1 Persone prvih korisnika aplikacije ..... 8

Tablica 3.1 Vrijeme izvođenja i veličina baze pri različitim načinima unosa utakmica ..... 36

Tablica 5.1 Razlike u objektima baze podataka prototipa i konačne aplikacije ..... 60

# Popis kôdova

Kôd 3.1 Program za unos kluba u bazu podataka ..... 18

Kôd 3.2 Program za pretragu i dohvaćanje podataka o igračima ..... 19

Kôd 3.3 Program za prikaz dohvaćenih podataka o igračima ..... 19

Kôd 3.4 Program za filtriranje identifikacijskog broja iz unosa ..... 20

Kôd 3.5 Program za provjeru dvostrukih vrijednosti unesenih igrača ..... 20

Kôd 3.6 Logika za provjeru postojanja prvog od timova u bazi podataka ..... 21

Kôd 3.7 Upit za provjeru postojanja tima u bazi podataka u funkciji teamCheck() ..... 21

Kôd 3.8 Upit za dohvaćanje tima iz baze podataka u funkciji teamGet() ..... 21

Kôd 3.9 Program za odbacivanje decimala broja ..... 22

Kôd 3.10 Program za identifikaciju dosadašnjih protivnika tima ..... 23

Kôd 3.11 Program za prebrojavanje dosadašnjih utakmica s protivnicima tima ..... 24

Kôd 3.12 Program za pripremu polja bodova dosadašnjih protivnika ..... 24

Kôd 3.13 Program za izračun medijana vrijednosti u polju ..... 25

Kôd 3.14 Program za izračun konačnih bodova timova u Elovom algoritmu ..... 26

Kôd 3.15 Upit za dohvaćanje isključivo igrača s odigranim utakmicama ..... 29

Kôd 5.1 Provjera pozivnog kôda i stvaranje novog korisnika – dio 1/2 ..... 58

Kôd 5.2 Provjera pozivnog kôda i stvaranje novog korisnika – dio 2/2 ..... 59

Kôd 5.3 Provjera ograničenja za pozivni kôd pri stvaranju istog ..... 60

Kôd 5.4 Program za potvrđivanje utakmice na upravljačkoj ploči ..... 63

# Literatura


[1] HOPPENBROUWER, J., WINKELS, M. *Sport Ratings*. Amsterdam, 2013.

[2] GLICKMAN, M. E., HENNESSY, J., BENT, A. *A Comparison of Rating Systems for Competitive Women's Beach Volleyball*. Cambridge, MA, 2016.

[3] HALL, E. *Just Enough Research*. New York: A Book Apart, 2013.

[4] ENDERS, J. *Designing UX: Forms*. Colingwood: Sitepoint, 2016.

[5] GARRETT, J. J. *The Elements of User Experience*. Berkeley: New Riders, 2011.

[6] BREZNIK, K., BAGATELJ, V. *FIDE Chess Network.* Ljubljana, 2011.

[7] SORENSEN, S. P. *An Overview of Some Methods for Ranking Sports Teams.* Tennessee, 1999.

[8] KRUG, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. San Francisco: New Riders, 2014.

[9] PHP priručnik: *BCMath*, dostupno: https://web.archive.org/web/20190503142420/https://www.php.net/manual/en/book.bc.php [pristupano: 3. svibnja 2019.]

[10] HALL, E. *Just Enough Research*. New York: A Book Apart, 2013.

[11] SHARON, T. *It's Our Research: Getting Stakeholder Buy-in for User Experience Research Projects*. Waltham: Morgan Kaufmann, 2012.

[12] SPENCER, D. *A Practical Guide to Information Architecture*. Penarth: Five Simple Steps, 2010.

[13] LYNCH, P. J., HORTON, S. *Web Style Guide: Foundations of User Experience Design*. New Haven: Yale University Press, 2016.

[14] KRUG, S. *Rocket Surgery Made Easy*. Berkeley: New Riders, 2010.

[15] TIDWELL, J. *Designing Interfaces: Patterns for Effective Interaction Design*. Sebastopol: O'Reilly Media, 2010.

[16] COOPER, A., REIMANN, R. et al., *About Face: The Essentials of Interaction Design*. Indianapolis: Wiley, 2014.

[17] Službena Laravel dokumentacija, dostupno: https://web.archive.org/web/20190901221055/https://laravel.com/docs/5.8/ [pristupano: 1. rujna 2019.]

# Prilozi

1) Prototip aplikacije za fino ugađanje parametara algoritma zajedno s testnom SQL bazom podataka s primjerima izračuna i testnim CSV datotekama za unos parova i utakmica

2) Vizualni prototipovi (wireframes) u Sketchu

3) Aplikacija izrađena u Laravel frameworku

ALGEBRA
VISOKO UČILIŠTE

**DIZAJN I IZRADA SUSTAVA**
**ZA POREDAK PAROVA I POJEDINACA**
**U TIMSKIM SPORTOVIMA**

Pristupnik: Antonio Britvar, 0321005881

Mentor: Predrag Šuka, pred.