

Oblikovanje programske potpore

Ak. god. 2015./2016.

Virtualna liga - rukomet

Dokumentacija, Rev. 2

Grupa: *gladijatori*

Voditelj: *Goran Brlas*

Datum predaje: *21. siječnja 2016.*

Nastavnik: *Danko Ivošević*

Popis članova grupe i zaduženja

Goran Brlas – voditelj projekta, koordinacija članova projekta, kontrola kvalitete, upravljanje ljudskim i projektnim resursima, dizajn UML dijagrama

Filip Džidić – dizajn UML dijagrama, arhitektura sustava, baza podataka, implementacija web-aplikacije

Antonio Ivčec – dizajn UML dijagrama, arhitektura sustava

Vinko Kolobara – dizajn UML dijagrama, arhitektura sustava, implementacija web-aplikacije, testiranje

Vladimir Dragutin Livaja – dizajn UML dijagrama, arhitektura sustava

Llore Torbarina – dizajn UML dijagrama, arhitektura sustava, baza podataka, implementacija web-aplikacije, testiranje

Lovro Vidović – dizajn UML dijagrama, arhitektura sustava, implementacija web-aplikacije, testiranje

Sadržaj

1.	Dnevnik promjena dokumentacije	4
2.	Opis projektnog zadatka	5
3.	Pojmovnik	7
4.	Funkcionalni zahtjevi	8
5.	Ostali zahtjevi	27
6.	Arhitektura i dizajn sustava	28
6.1.	Svrha, opći prioriteti i skica sustava	28
6.2.	Dijagram razreda s opisom	34
6.3.	Dijagram objekata	38
6.4.	Ostali UML dijagrami	41
7.	Implementacija i korisničko sučelje	48
7.1.	Dijagram razmještaja	48
7.2.	Korištene tehnologije i alati	48
7.3.	Isječak programskog koda vezan za temeljnu funkcionalnost	50
7.4.	Ispitivanje programskog rješenja	55
7.5.	Upute za instalaciju	71
7.6.	Korisničke upute	73
8.	Zaključak i budući rad	83
9.	Popis literature	85
	Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)	86
	Dodatak B: Dnevnik sastajanja	89
	Dodatak C: Prikaz aktivnosti grupe	92
	Dodatak D: Plan rada	94
	Dodatak E: Apache Derby	95
	Dodatak F: Apache Tomcat	100

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak. Dodan opis projektnog zadatka. Započeo analizu funkcionalnih zahtjeva.	Brlas	26.10.2015.
0.2	Dodan Use Case dijagram.	Brlas	26.10.2015.
0.3	Dodan sekvencijski dijagram.	Vidović	1.11.2015.
0.4	Dodani ostali zahtjevi.	Brlas	10.11.2015.
0.5	Dodan dijagram razreda.	Kolobara	12.11.2015.
0.6	Izmjenjena UC slika, dodan riječnik pojmova.	Brlas	12.11.2015.
0.7	Dodan dijagram ER baze podataka i skica sustava.	Džidić, Torbarina	16.11.2015.
0.8	Dodan dijagram objekata.	Livaja, Ivčec	18.11.2015.
1.0	Dodan zaključak, popis literature te dnevnik sastajanja.	Brlas	18.11.2015.
1.2	Dodan dijagram aktivnosti, dnevnik sastajanja.	Brlas	10.1.2016.
1.6	Dodane upute za instalaciju, korisničke upute, zaključak, update-an prikaz aktivnosti grupe, dodatci E i F.	Brlas	16.1.2016.
1.7	Dodan komunikacijski dijagram, dijagram stanja, dijagram komponenti i dijagram razmještaja.	Livaja, Torbarina, Ivčec	19.1.2016.
1.9	Ispavljeni potrebni dijagrami iz rev. 1.	Kolobara	19.1.2016.
2.0	Uređen tekst, dodan indeks slika, ispravljene gramatičke pogreške.	Brlas	20.1.2016.

2. Opis projektnog zadatka

Cilj projekta jest razviti web aplikaciju koja će služiti za praćenje rezultata međunarodne rukometne lige, internetski prijenos svih utakmica te nagradne igre za posjetitelje portala, u obliku virtualne lige koja bi koristila igrače iz postojeće baze. Funkcionalnosti koje se želi postići su jednostavna i minimalistička interakcija korisnika sa sustavom koji omogućuje jednostavno pregledavanje rezultata, kao i sudjelovanje u virtualnoj ligi.

Predviđa se izrada portala koji bi se sastojao od web mjesta na kojem bi se objavljivali rezultati utakmica te bi bila ostvarena programska potpora za prijavu i igranje virtualne lige. Na njemu bi bilo moguće upisati rezultate utakmica zajedno s ostalim, za virtuelnu ligu bitnim, događajima. Svi upisani podaci o utakmicama javno bi se objavljivali na portalu udruženja, a dio programskog sustava koji se odnosi na virtualnu ligu to bi pravilno evidentirao kroz odgovarajuće bodovanje izabranih igrača, što će se odraziti na uspješnost prijavljenih natjecatelja virtualne lige.

Osim ručnog upisivanja podataka bit će ostvaren i simulator koji će za prijavljene sastave rukometnih ekipa simulirati odigravanje utakmica, odnosno generirati rezultate i za virtualnu ligu bitne događaje koji će se automatski upisati u sustav te odraziti na bodove natjecatelja.

Kao sustav rukometnog natjecanja pretpostavljen je raspored odigravanja utakmica po sustavu *Bergerovih tablica* (https://hr.wikipedia.org/wiki/Bergerove_tablice) .

Sve nepredviđene ili krive akcije korisnika također su pokrivene unutar sustava. U sustavu se, osim prijavljenih natjecatelja, razlikuje još nekoliko vrsta korisnika.

Posjetitelji portala koji nisu registrirani natjecatelji mogu samo pratiti rezultate i poredak stvarne rukometne lige.

Službena osoba definira proračun natjecatelja, cijene igrača za virtualnu ligu, pravila bodovanja unutar virtualne lige te upisuje događaje utakmica koje se odigravaju.

Administrator se brine za učitavanje konfiguracije stvorene od strane službene osobe, samo učitavanje podataka, te provodi simulaciju dijela ili cijelog rukometnog turnira.

Posebna *tehnička komisija* proglašava najboljeg igrača svake utakmice, a njihov ga predstavnik upisuje u sustav.

Natjecatelji virtualne lige registriraju se u sustav navodeći svoje ime, prezime, e-mail adresu, korisničko ime, naziv svoje virtualne ekipe, državu iz koje dolazi te ime rukometne ekipe koju podupire. Nakon toga biraju 7 igrača u svoju virtualnu ekipu unutar dozvoljenog proračuna i po zadanim pozicijama. Nakon svake odigrane utakmice imaju uvid u stanje bodova i bodovanje te prate svoj napredak na ukupnoj ljestvici. Osim ukupne, globalne ljestvice svih natjecatelja, sustav ujedno generira i ljestvice virtualnih ekipa po državama iz koje natjecatelji dolaze i po rukometnim ekipama koje natjecatelji podupiru.

3. Pojmovnik

Apache Derby: open source relacijska baza podataka izrađena u Javi.

Apache Tomcat: open source aplikacijski web poslužitelj za pokretanje i prikaz web stranica.

CSS (Cascading Style Sheets): stilski jezik, rabi se za opis prezentacije dokumenta napisanog pomoću markup (HTML) jezika.

DAO (Data Access Object): objekt koji pruža apstraktno sučelje za rad s bazom podataka.

Eclipse: programska razvojna okolina (IDE) pisana u Javi, a može se koristiti za razvoj aplikacija u raznim programskim jezicima kao što su Java, Ada, C, C++, COBOL, Perl, PHP, Python, R, Ruby, itd. Razvojna okolina (IDE) često se naziva Eclipse ADT (Ada Development Toolkit) za Adu, Eclipse CDT za C/C++, Eclipse JDT za Javu i Eclipse PDT za PHP.

Git: distribuirani sustav za upravljanje izvornim kodom.

Hibernate: objektno/relacijska tehnologija koja povezuje objekte iz Jave s relacijskim tablicama baze podataka.

HTML (HyperText Markup Language): prezentacijski jezik za izradu web stranica.

Java: Objektno orijentirani programski jezik koji se može iskoristiti za izradu web aplikacija.

JSP (JavaServer Pages): tehnologija koja se koristi za izradu dinamički generiranih web staranica baziranih na HTML, XML ili drugim tipovima dokumenata. Sličan je PHP-u i ASP-u, ali je zasnovan na Javi.

Servlet: mali Java program koji se izvršava na web serveru. Obraduje zahtjeve koje dobiva od web klijenata, najčešće preko HTTP-a.

SQL: Programski jezik za manipulaciju bazom podataka.

UML (Unified/Universal Modeling Language): standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

4. Funkcionalni zahtjevi

Dionici:

- Posjetitelj
- Natjecatelj
- Službena osoba
- Predstavnik tehničke komisije
- Administrator baze podataka

Aktori i njihovi funkcionalni zahtjevi:

- Posjetitelj, inicijator
 - Prati rezultate utakmica
 - Prati poredak rukometne lige
 - Može se registrirati i postati natjecatelj
- Natjecatelj, inicijator
 - Registrira se navodeći potrebne podatke
 - Bira 7 igrača u svoju virtualnu ekipu unutar svog proračuna
 - Ostvaruje bodove ovisno o tome kako izabrani igrači igraju
 - Uvid u ukupno stanje bodova, ljestvicu virtualne lige i ostale statističke podatke
- Službena osoba, inicijator
 - Definira proračun i cijene igrača za virtualnu ligu
 - Definira pravila bodovanja unutar virtualne lige
 - Upisuje događaje utakmica koji su relevantni za virtualnu ligu
- Predstavnik tehničke komisije, inicijator
 - Upisuje najboljeg igrača svake utakmice

- Administrator, inicijator

- Učitava konfiguraciju stvorenu od strane službene osobe
- Učitava podatke vezane uz ligu
- Provodi simulaciju utakmica i cijelog turnira

- Baza podataka, sudionik

- Čuva podatke o svim registriranim korisnicima
- Čuva podatke o rukometnoj ligi
- Čuva podatke o virtualnoj ligi

Opis obrazaca uporabe:

- UC1 – PogledajStvarnoStanje

- **Glavni sudionik:** Posjetitelj, Natjecatelj.
- **Cilj:** Pogledati trenutno stanje u rukometnoj ligi, pojedinačne rezultate utakmica, ostale statističke podatke
- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema.
- **Rezultat:** Prikaz svih trenutnih stanja u ligi te pojedinačnih utakmica po potrebi.
- **Željeni scenarij:**

1. Tablica trenutnog stanja je prikazana prilikom učitavanja stranice.
2. Ako korisnik želi pogledati pojedinačne rezultate utakmica, treba odabrati tu opciju.
3. Ako korisnik želi, može pogledati pojedinačni učinak igrača koje je izabrao za virtualnu ligu
4. Ako korisnik želi, može pregledati sve događaje bitne za virtualnu ligu (strijelce, isključenja i dr.)

5. Sustav vrši pretraživanje baze podataka i prikazuje rezultate utakmica u kronološkom redu, počevši od najnovijih rezultata.
6. Sustav vrši pretraživanje i prikazuje statističke podatke

- **UC2 – RegistrirajNatjecatelja**

- **Glavni sudionik:** Posjetitelj.
- **Cilj:** Registrirati novog natjecatelja u bazu podataka.
- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema.
- **Rezultat:** Registracija novog korisnika i slanje elektroničke pošte s potvrdom korisniku.
- **Željeni scenarij:**
 1. Korisnik na vrhu stranice odabire poveznicu za prijavu novog korisnika te se otvara stranica za prijavu.
 2. Korisnik unosi sve potrebne podatke te odabire tipku za potvrdu unosa.
 3. Sustav provjerava ispravnost podataka.
 4. Sustav prihvaca podatke i u bazu podataka upisuje podatke o novom natjecatelju.
 5. Sustav šalje e-mail korisniku sadržaja o uspješnoj registraciji, kao i potvrdu o odabranim podacima.
- **Mogući drugi scenarij:**
 - Baza podataka ne može upisati novog natjecatelja, zato što potrebni podaci nisu dobro uneseni ili je unesena već postojeća e-mail adresa, naziv virtualne ekipe ili korisničko ime.
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu sa obavijesti o kojem tipu greške se radi.
 2. Korisnik modificira potrebne podatke te odabire tipku za završetak unosa ili odustaje od registracije.

- **UC3 – SastaviEkipu**

- **Glavni sudionik:** Natjecatelj.
- **Cilj:** Sastaviti virtualnu ekipu koja se sastoji od 7 igrača.
- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema. Korisnik već nema izabranu ekipu.
- **Rezultat:** Natjecatelj sastavlja vlastitu ekipu s kojom se onda natječe u virtualnoj ligi.
- **Željeni scenarij:**
 1. Korisnik po poziciji iz padajućeg izbornika odabire igrače ili pretražuje igrače po imenu ili prezimenu, vodeći računa o svom proračunu.
 2. Nakon što su svi igrači odabrani, sustav pohranjuje korisnikovu ekipu u bazu podataka.
- **Mogući drugi scenarij:**
 3. Ukoliko je korisnik prekoračio svoj proračun, sustav ga obavještava te ne prihvaca ekipu.
 4. Ukoliko je korisnik izabrao istog igrača na više pozicija, sustav ga obavještava te ne prihvaca ekipu.
 5. Ukoliko je korisnik izabrao više igrača za istu poziciju, sustav ga obavještava te ne prihvaca ekipu.

- **UC4 – PogledajStanjeVirtualneLige**

- **Glavni sudionik:** Natjecatelj.
- **Cilj:** Dobiti uvid u trenutno stanje virtualne lige.
- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema.
- **Rezultat:** Natjecatelj dobiva uvid u trenutno stanje virtualne lige.

- **Željeni scenarij:**

1. Natjecatelj u svakom trenutku ima pogled na ukupnu virtualnu ljestvicu.
2. Korisnik može odabratи drukčiji pogled na tablicu:
 - a. Može odabratи koliko bodova je od početka lige do sad osvojio pojedini igrač u vlastitoj ekipi.
 - b. Može odabratи poredak svih natjecatelja koji dolaze iz iste države.
 - c. Može odabratи poredak svih natjecatelja koji podržavaju istu ekipu.
3. Sustav korisniku prikazuje željeni odabir.

- **UC5 – DefinirajPočetnuKonfiguraciju**

- **Glavni sudionik:** Službena osoba.
- **Cilj:** Definirati proračun natjecatelja i cijene igrača za virtualnu ligu te pravila bodovanja unutar virtualne lige.
- **Sudionici:** Administrator.
- **Preduvjeti:** Liga još nije počela.
- **Rezultat:** Definiran proračun, cijene te pravila bodovanja unutar virtualne lige.
- **Željeni scenarij:**

1. Službena osoba definira proračun, cijene te pravila bodovanja unutar virtualne lige.
2. Definirani podaci izvode se u odgovarajućem formatu.
3. Definirani podaci učitavaju se kao početna konfiguracija od strane administratora.

- **UC6 – UpišiDogađaje**

- **Glavni sudionik:** Službena osoba.
- **Cilj:** Upisati događaje utakmica koje se odigravaju.

- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema.
- **Rezultat:** Upisani događaji utakmica koje se odigravaju.
- **Željeni scenarij:**
 1. Službena osoba odabire utakmicu za koju želi upisati događaje.
 2. U bazu podataka se upisuju događaji relevantni za virtulanu ligu za odabranu utakmicu te se automatski osvježava stanje u tablici virtualne lige.

- **UC7 – ProglasiMVP**

- **Glavni sudionik:** Predstavnik tehničke komisije.
- **Cilj:** Upisati igrača utakmice za odabranu utakmicu.
- **Sudionici:** Baza podataka.
- **Preduvjeti:** Nema.
- **Rezultat:** Upisan igrač utakmice za odabranu utakmicu u bazu podataka.
- **Željeni scenarij:**
 1. Predstavnik tehničke komisije odabire utakmicu za koju želi proglašiti glavnog igrača ukoliko on nije automatski proglašen od strane simulatora.
 2. U bazu podataka se upisuje igrač utakmice te se automatski osvježava stanje na tablici virtualne lige.

- **UC8 – AdministrirajBazuPodataka**

- **Glavni sudionik:** Administrator.
- **Cilj:** Učitati podatke ili početnu konfiguraciju u bazu podataka.
- **Sudionici:** Baza podataka.

- **Preduvjet:** Ako se želi učitati početna konfiguracija, ona prvo mora biti definirana od strane službene osobe.
- **Rezultat:** Učitani podatci ili početna konfiguracija.
- **Željeni scenarij:**
 1. Administrator iz izbornika odabire želi li učitati početnu konfiguraciju ili podatke vezane uz rukometnu ligu ili natjecatelje.
 2. Željeni podatci ili konfiguracija učitavaju se u bazu podataka.

- **UC9 – Simuliraj**

- **Glavni sudionik:** Administrator.
- **Cilj:** Simulirati jednu utakmicu ili cijeli turnir.
- **Sudionici:** Baza podataka.
- **Preduvjet:** Nema.
- **Rezultat:** Simulirana jedna utakmica ili cijeli turnir.
- **Željeni scenarij:**
 1. Administrator iz izbornika odabire želi li simulirati jednu utakmicu ili cijeli turnir.
 2. Ako simulira jednu utakmicu, onda odabire točnu utakmicu iz izbornika te pokreće simulator. Ako se simulira cijeli turnir, odabere se ta opcija i pokreće simulator.
 3. Rezultat simulacije se unosi u bazu podataka i automatski se osvježava stanje na tablici rukometnih ekipa, kao i stanje tablice virtualne lige.

- **UC10 – UlogirajSe**

- **Glavni sudionik:** Natjecatelj, Službena osoba, Predstavnik tehničke komisije, Administrator.
- **Cilj:** Ulogirati se u sustav.
- **Sudionici:** Baza podataka.

- **Preduvjet:** Nema.

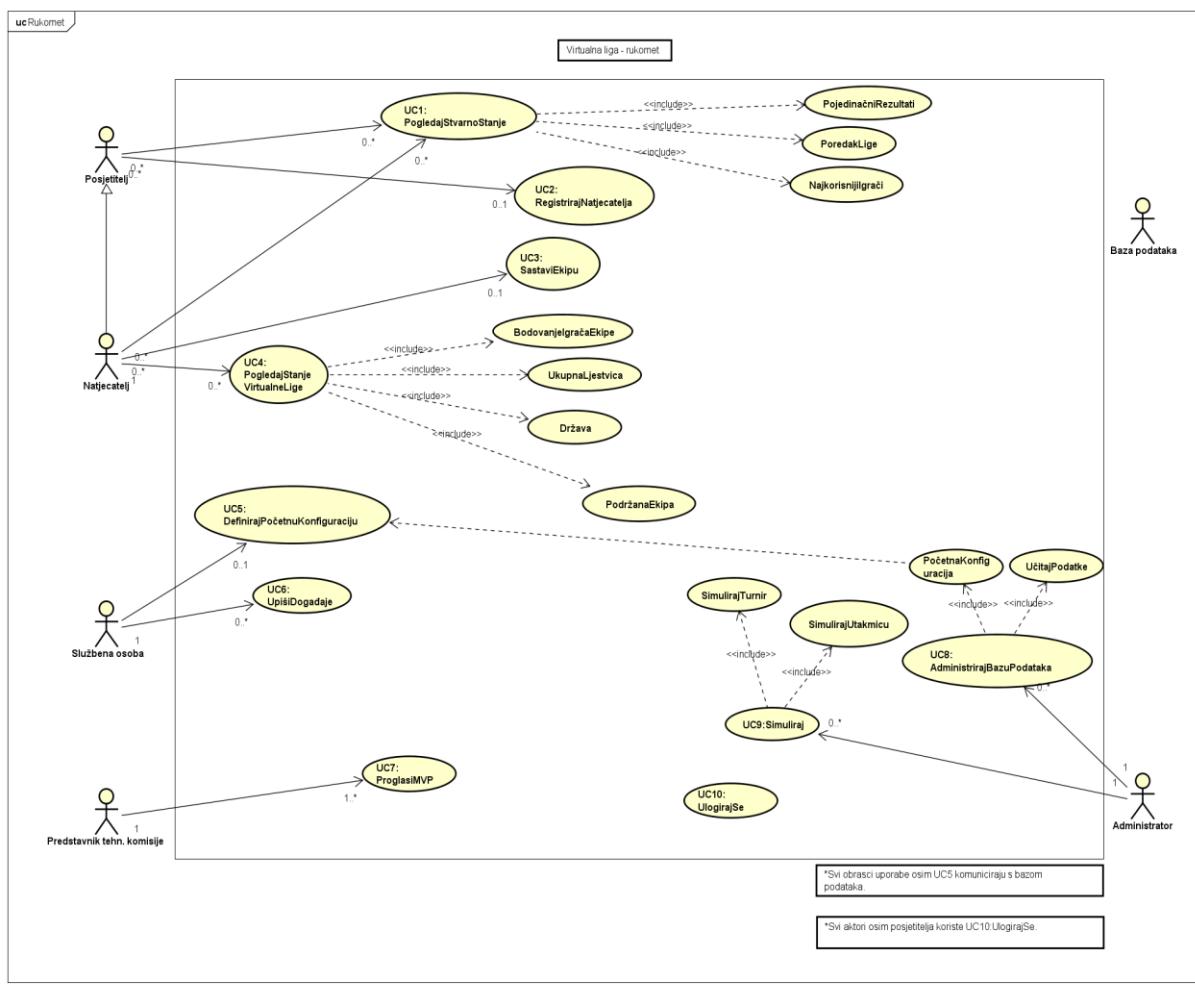
- **Rezultat:** Korisnik ulogiran u sustav te su mu dodijeljena prava sukladna sa njegovim ovlastima.

- **Željeni scenarij:**

1. Korisnik na vrhu stranice odabire poveznicu za prijavu novog korisnika te se otvara stranica za prijavu.
2. Korisnik unosi potrebne podatke.
3. Korisnik se ulogira u sustav.

- **Mogući drugi scenarij:**

1. Ukoliko je korisnik unio pogrešne podatke, sustav mu to javlja odgovarajućom porukom.



Slika 4.1 Sekvencijski dijagram za UC10 (UlogirajSe)

Sekvencijski dijagrami

Obrazac uporabe UC1

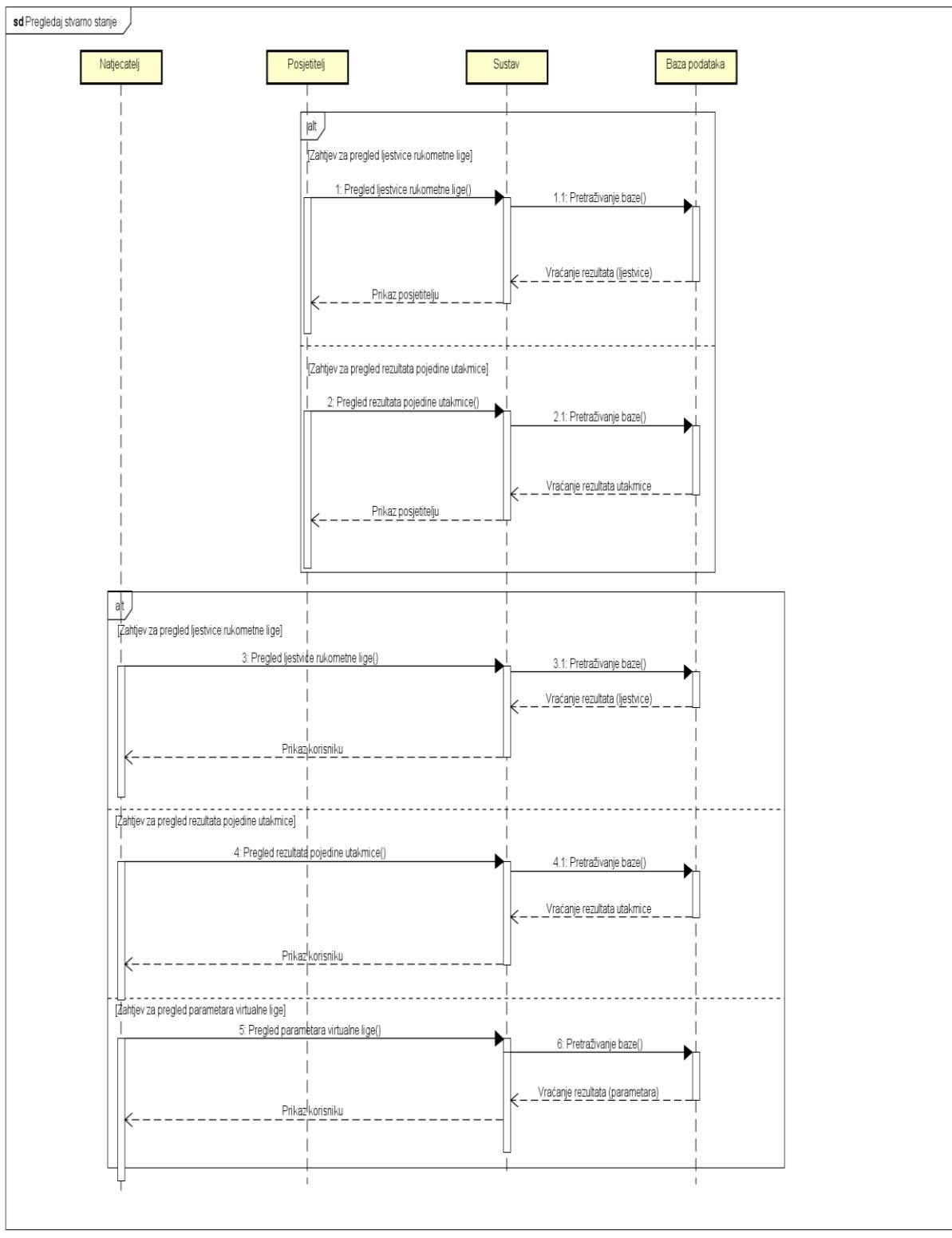
Pregled stvarnog stanja rukometne lige može biti izvršen od strane korisnika virtualne lige (registrirane osobe koja je ujedno i natjecatelj) te od strane posjetitelja portala (neregistrirane osobe).

Korisnik, uz praćenje rezultata stvarne rukometne lige, ima mogućnost praćenja parametara potrebnih za ravnopravno natjecanje u virtualnoj ligi: stanja na ljestvici virtualne lige te učinka igrača koje je izabrao za sudjelovanje u virtualnoj ligi.

Korisnik (natjecatelj) i posjetitelj imaju mogućnost gledanja rezultata svake utakmice odabirom potrebne opcije koja će omogućiti prikaz rezultata.

Korisnik učitava stranicu te mu se nudi prikaz tablice rukometne lige, rezultata utakmice, statističkih podataka, ljestvice stvarne rukometne lige te tablice najkorisnijih igrača.

Prikaz svih potrebnih parametara za sudjelovanje u virtualnoj ligi vrši se uz pomoć sustava i baze podataka. Sustav vrši pretraživanje baze podataka i prikazuje rezultate utakmica u kronološkom redu, počevši od najnovijih rezultata. Na isti se način prikazuju i statistički podatci vezani za pojedinu utakmicu.

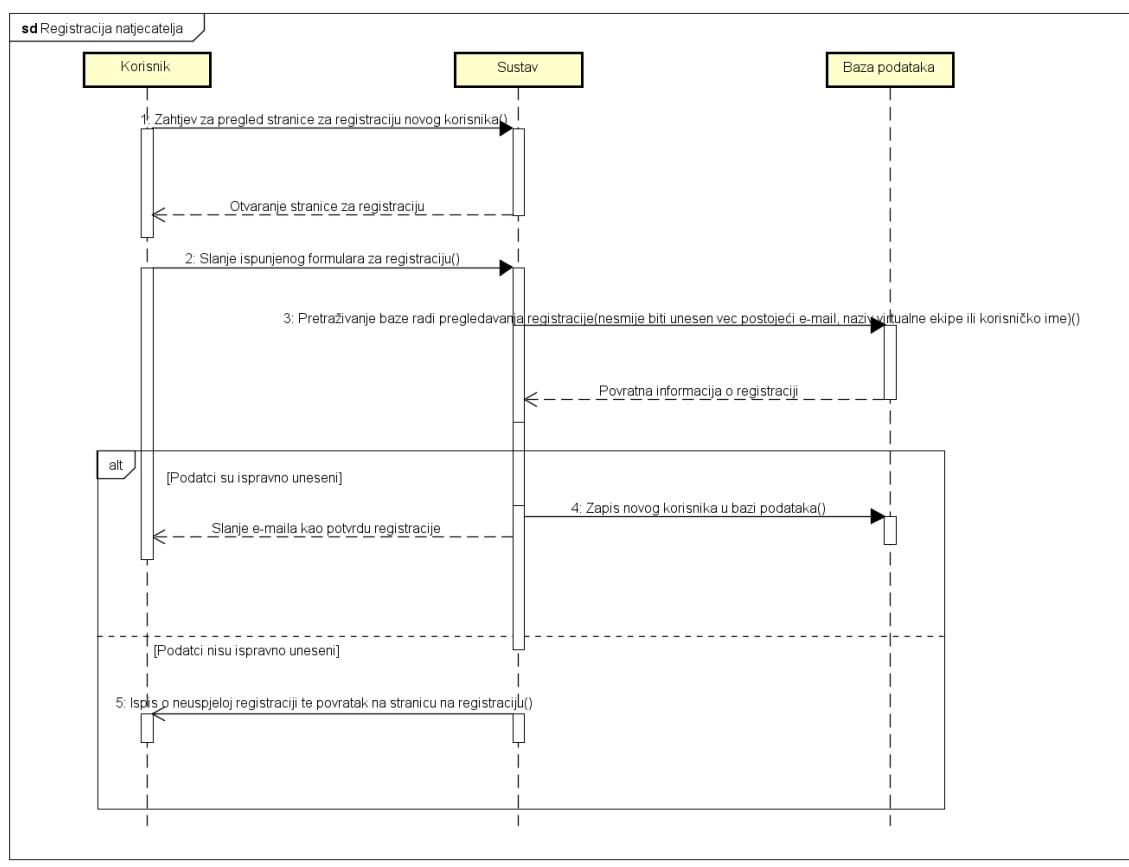
**Slika 4.2 Sekvencijski dijagram za UC1 (PogledajStvarnoStanje)**

Obrazac uporabe UC2

Korisnik na vrhu stranice odabire poveznicu za prijavu novog korisnika koja mu nudi mogućnost registriranja. Stranica sadrži formular za registraciju. Za samu registraciju nužno je upisati: ime, prezime, e-mail adresu, korisničko ime, šifru, naziv svoje virtualne ekipe, državu iz koje dolazi te ime rukometne ekipe koju podupire.

Korisnik unosi potrebne podatke te potvrđuje unos podataka. Sustav provjerava ispravnost unesenih podataka te, ako su svi podaci dobro uneseni, šalje e-mail korisniku kojim se potvrđuje uspješna registracija te se korisnik upisuje u bazu podataka.

Ako uneseni podatci nisu ispravno uneseni ili je unesena već postojeća e-mail adresa, naziv virtualne ekipe ili korisničko ime, sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu sa obavijesti o kojem tipu greške se radi.



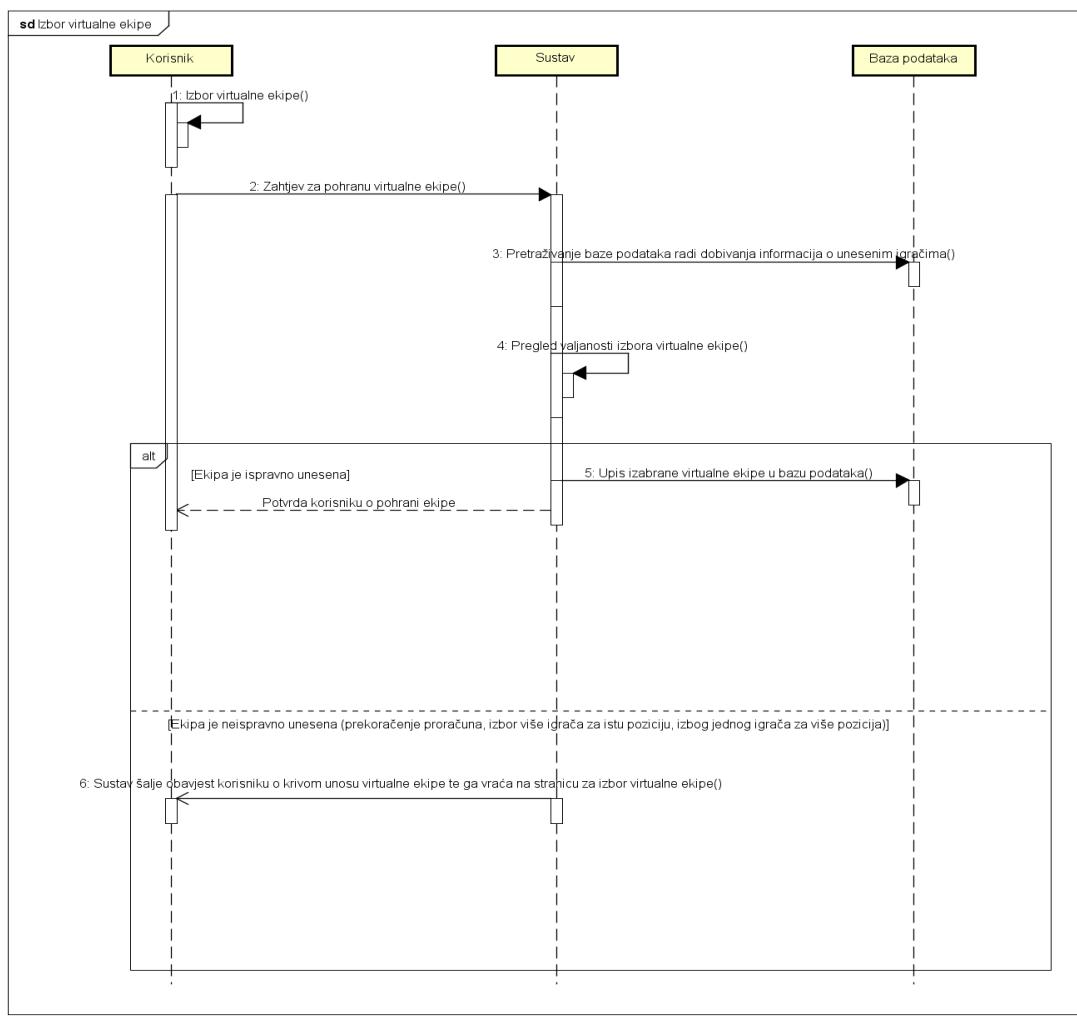
Slika 4.3 Sekvencijski dijagram za UC2 (RegistrirajNatjecatelja)

Obrazac uporabe UC3

Nakon uspješne registracije korisnik je proslijeđen na stranicu koja nudi mogućnost izbora vlastite ekipe s kojom se natječe u virtualnoj ligi. Korisnik odabire igrače pretražujući po imenu i prezimenu željenog igrača.

Nakon što su svi korisnikovi igrači odabrani, sustav čita podatke iz baze podataka za odabranu ekipu. U slučaju da je korisnik prilikom izbora virtualne ekipe prekoračio proračun, izabrao istog igrača na više pozicija ili više igrača za istu poziciju, sustav ga obavještava o grešci te ga vraća na stranicu koja nudi mogućnost izbora ekipe za sudjelovanje u virtualnoj ligi.

Ukoliko je korisnik izabrao vlastitu ekipu na pravilan način, ekipa se zapisuje u bazu podataka.

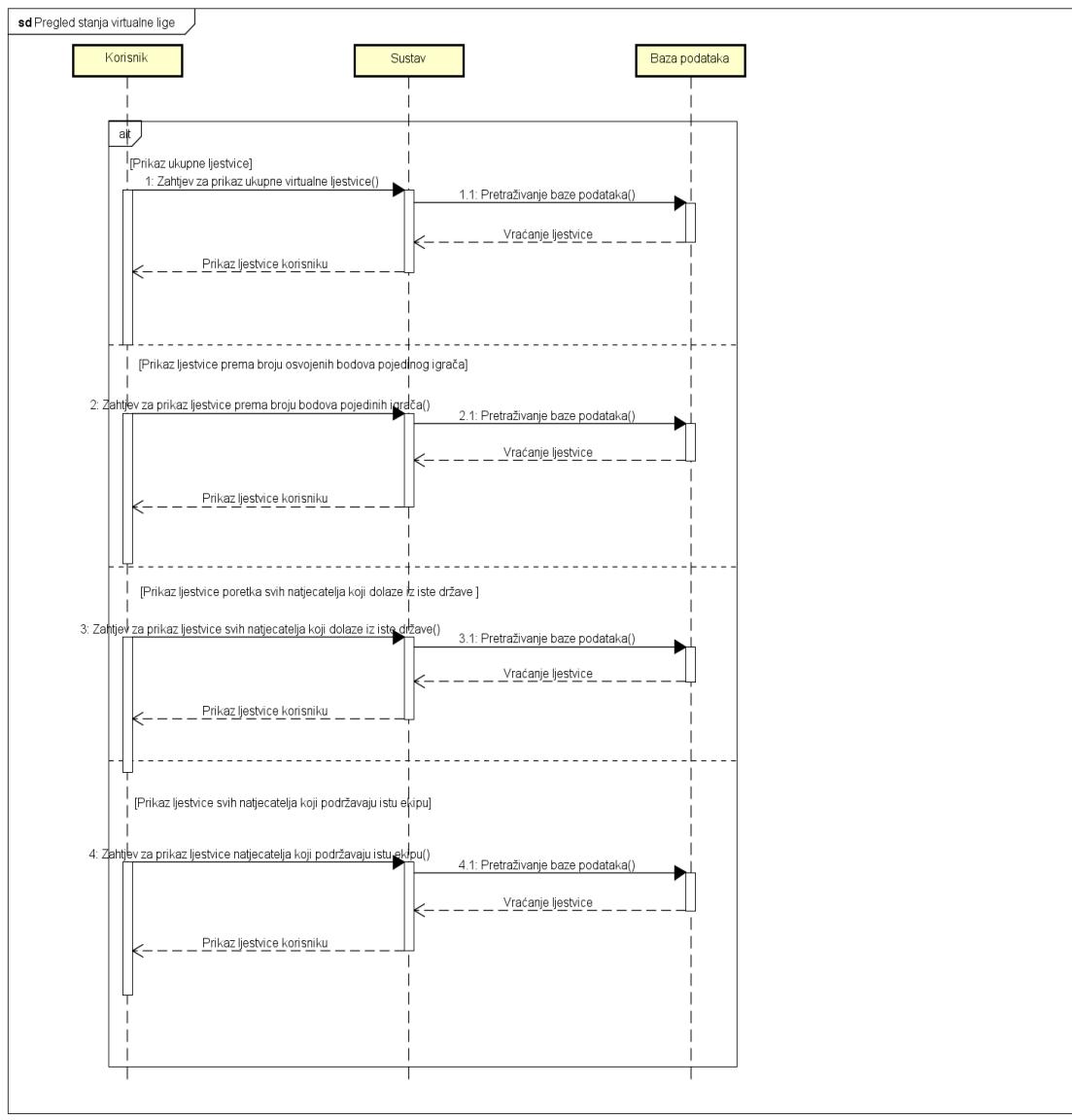


Slika 4.4 Sekvencijski dijagram za UC3 (SastaviEkipu)

Obrazac uporabe UC4

Sustav nudi mogućnost prikaza stanja na ukupnoj ljestvici svakom natjecatelju u bilo kojem trenutku (sustav šalje upit bazi podataka koja vraća ukupnu ljestvicu). Također, korisnik može izabrati i drugačiji pogled na tablicu. Može odabratи pogled stanja bodova igrača u vlastitoj ekipi, poredak svih natjecatelja koji dolaze iz iste države te poredak svih natjecatelja koji podržavaju istu ekipu.

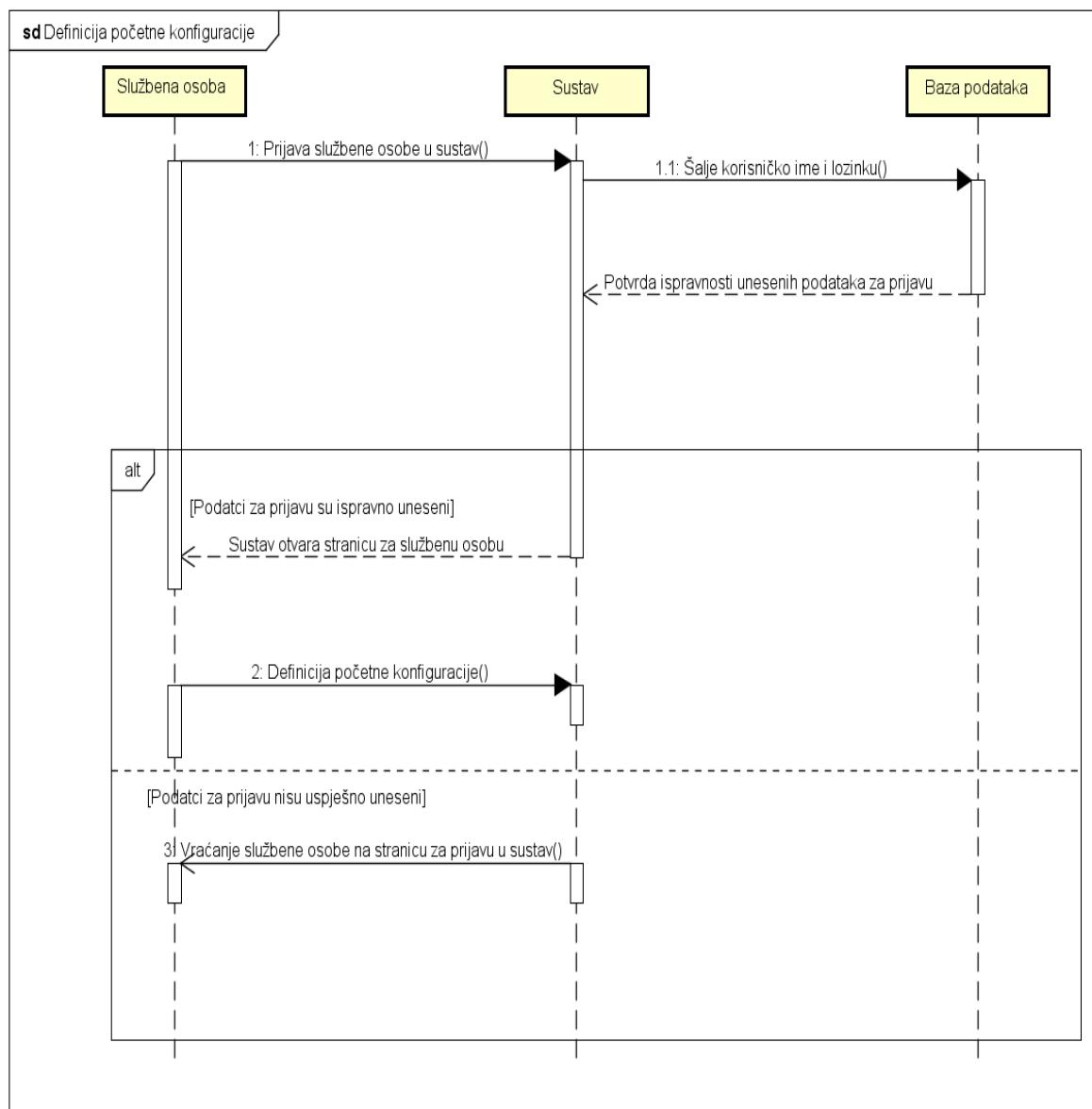
Nakon odabira željenog prikaza tablice sustav šalje upit bazi podataka za prikaz specifičnog oblika tablice korisniku te se korisniku prikazuje tražena tablica.



Slika 4.5 Sekvencijski dijagram za UC4 (PogledajStanjeVirtualneLige)

Obrazac uporabe UC5

Službena osoba prijavljuje se na sustav i odlazi na posebnu stranicu za definiranje parametara važnih za sudjelovanje u virtualnoj ligi. Definira proračun svakog natjecatelja, cijene igrača te pravila bodovanja unutar virtualne lige popunjavanjem odgovarajućih polja na stranici. Stvara se poveznica za izvoz konfiguracije virtualne lige u određenom formatu. Ukoliko je krivo unesena lozinka ili korisničko ime službena osoba se vraća na početnu stranicu za prijavu u sustav.

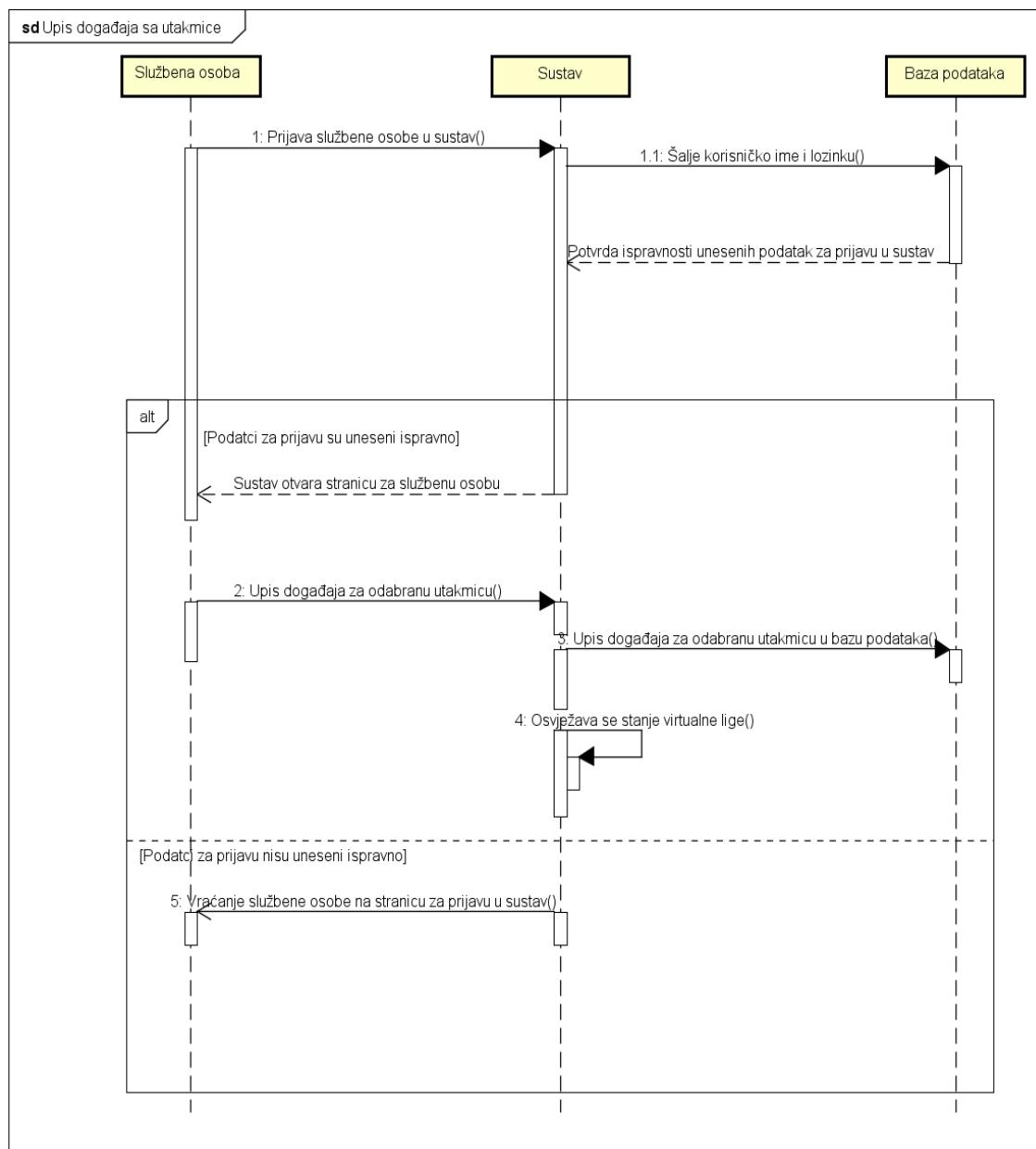


powered by Astah

Slika 4.6 Sekvencijski dijagram za UC5 (DefinirajPočetnuKonfiguraciju)

Obrazac oprabe UC6

Službena osoba prijavljuje se u sustav. U slučaju pogrešnog unosa lozinke ili korisničkog imena sustav vraća službenu osobu na stranicu za prijavu u sustav. Nakon uspješne prijave u sustav službena osoba bira utakmicu za koju želi upisati događaje. U bazu podataka se upisuju događaji relevantni za virtuelanu ligu za odabranu utakmicu te se automatski osvježava stanje u tablici virtualne lige.



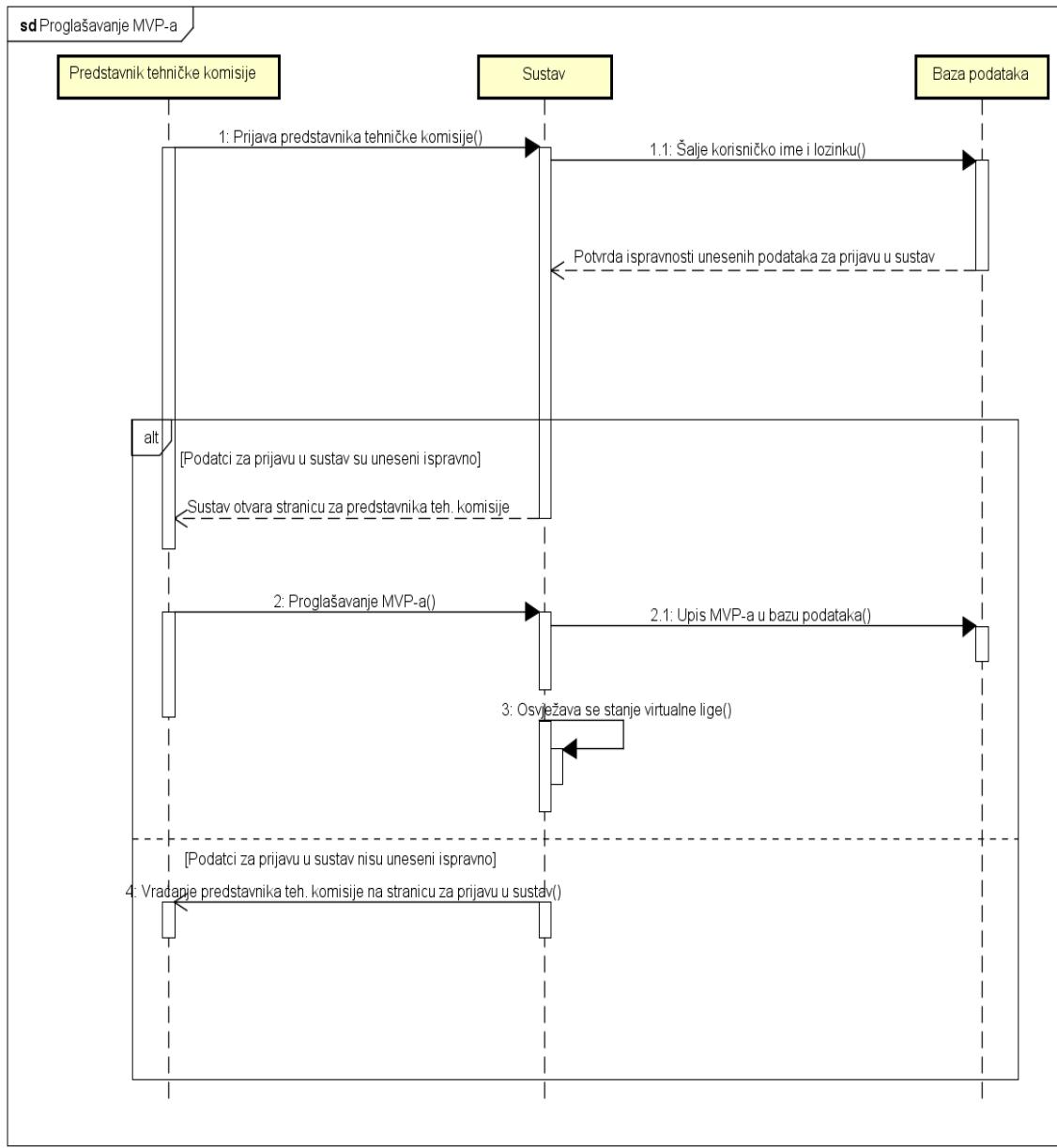
powered by Astah

Slika 4.7 Sekvencijski dijagram za UC6 (UpisiDogađaje)

Obrazac uporabe UC7

U sustav se prijavljuje predstavnik tehničke komisije. U slučaju krivog unosa korisničkog imena ili lozinke korisnik se prosljeđuje na stranicu za prijavu u sustav.

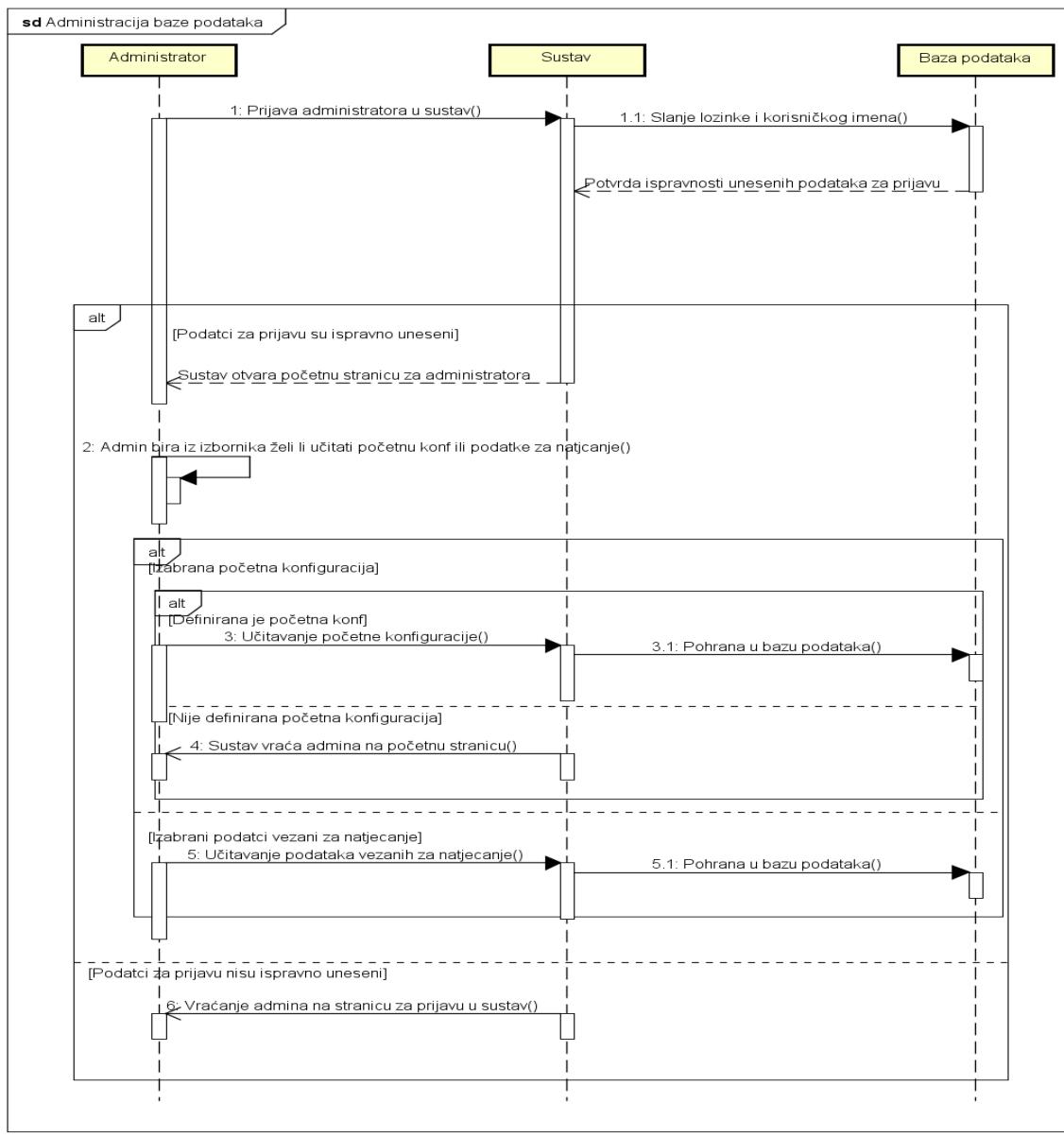
Predstavnik tehničke komisije odabire utakmicu za koju želi proglašiti glavnog igrača ukoliko on nije automatski proglašen od strane simulatora. U bazu podataka upisuje se igrač utakmice te se automatski osvježava stanje na tablici virtualne lige.



Slika 4.8 Sekvenčni dijagram za UC7 (ProglasiMVP)

Obrazac uporabe UC8

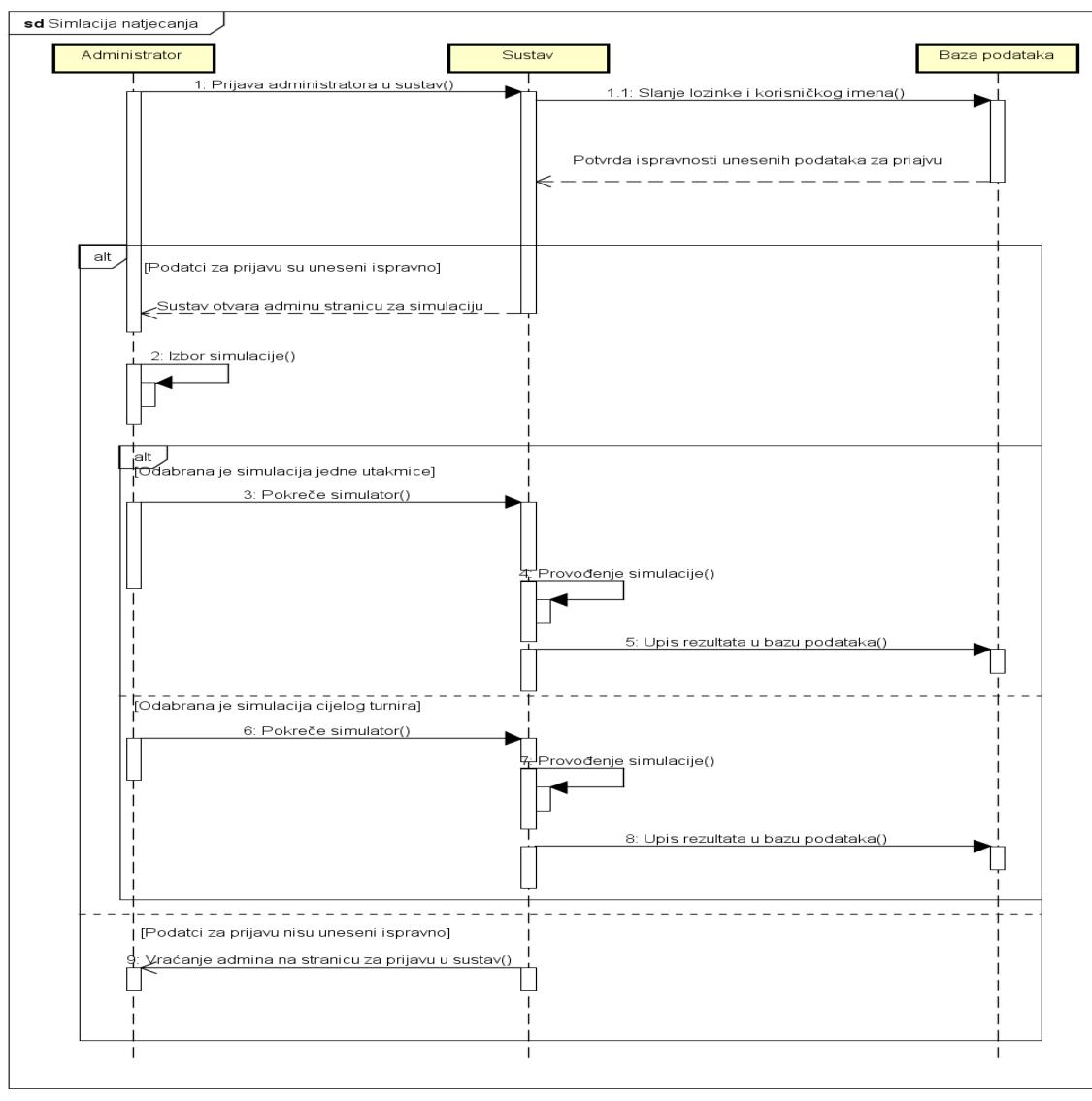
Administrator se prijavljuje u sustav. U slučaju krivog unosa korisničkog imena ili lozinke korisnik se prosljeđuje na stranicu za prijavu u sustav. Nakon uspješne prijave u sustav administrator iz izbornika odabire želi li učitati početnu konfiguraciju ili podatke vezane uz rukometnu ligu ili natjecatelje. Željeni podaci ili konfiguracija učitavaju se u bazu podataka.



Slika 4.9 Sekvencijski dijagram za UC8 (AdministrirajBazuPodataka)

Obrazac uporabe UC9

Administrator se prijavljuje u sustav. U slučaju krivog unosa korisničkog imena ili lozinke proslijeđuje se na stranicu za prijavu u sustav. Nakon uspješne prijave u sustav administrator iz izbornika odabire želi li simulirati jednu utakmicu ili cijeli turnir. Ako simulira jednu utakmicu, onda odabire točnu utakmicu iz izbornika te pokreće simulator. Ako se simulira cijeli turnir, odabere se ta opcija i pokreće simulator. Rezultat simulacije se unosi u bazu podataka i automatski se osvježava stanje na tablici rukometnih ekipa, kao i stanje tablice virtualne lige.



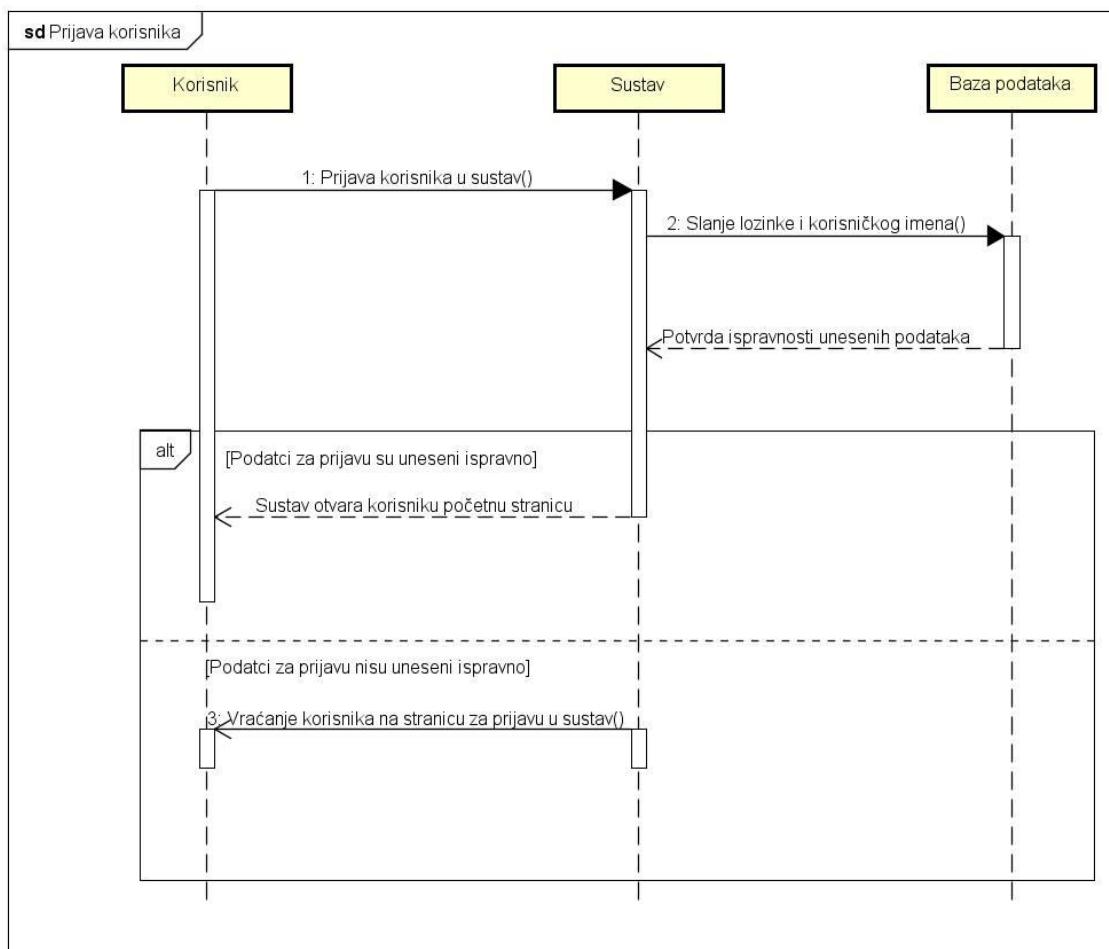
Slika 4.10 Sekvencijski dijagram za UC9 (Simuliraj)

Obrazac uporabe UC10:

Korisnik (natjecatelj, službena osoba, predstavnik tehničke komisije, administrator) na vrhu stranice odabire poveznicu za prijavu novog korisnika te se otvara stranica za prijavu. Korisnik unosi potrebne podatke za prijavu u sustav (korisničko ime i lozinku) te stisne gumb za prijavu u sustav.

Ukoliko je dobro unesena lozinka i korisničko ime korisnik je uspješno prijavljen u sustav i sustav otvara početnu stranicu.

Ukoliko je korisnik prilikom prijave napisao pogrešno korisničko ime ili lozinku sustav mu javlja informaciju o grešci te ga usmjerava na stranicu za prijavu u sustav.



powered by Astah

Slika 4.11 Sekvencijski dijagram za UC10 (UlogirajSe)

5. Ostali zahtjevi

- Sustav mora podržavati paralelni rad više korisnika,
- Korisničko sučelje i sustav trebaju podržavati hrvatske znakove,
- Sustav mora biti realiziran tako da omogućuje jednostavno korištenje bez potrebe za detaljnim uputama,
- Pretraga baze podataka koju inicira korisnik ne smije trajati duže od 10 sekundi,
- Korisničko sučelje mora omogućiti više različitih razina dostupnosti ovisno o dodijeljenim korisničkim ovlastima,
- Opisana aplikacija mora biti prilagođena izvedbi na Webu,
- Nepravilno i nepredviđeno ponašanje korisnika ne smije narušiti funkcionalnost i rad sustava,
- Nadogradnja ili uvođenje potpuno novih funkcionalnosti ne smije narušiti postojeće funkcionalnosti kao ni sustav u cjelini,
- Cjelokupni sustav i priložena dokumentacija moraju biti izvedeni do siječnja 2016. godine.

6. Arhitektura i dizajn sustava

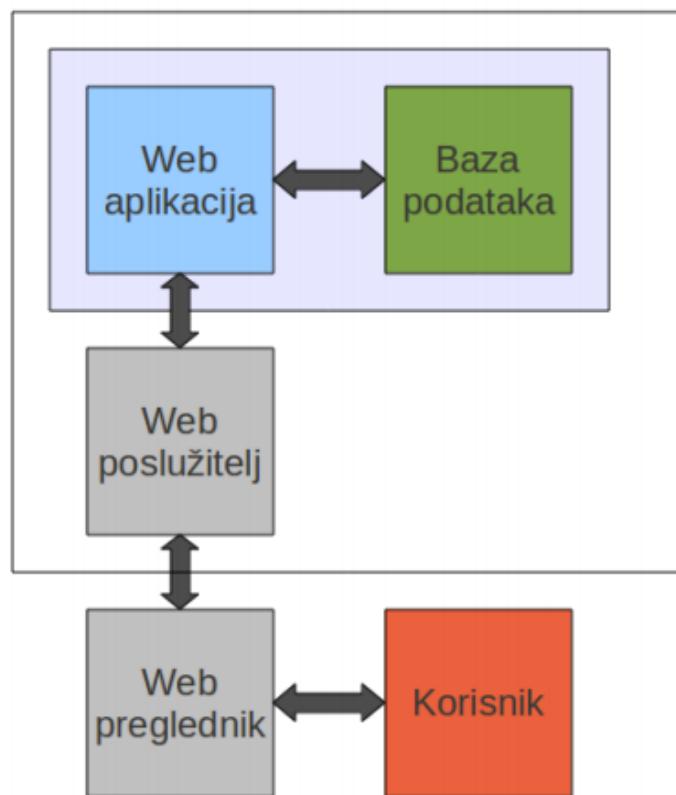
6.1. Svrha, opći prioriteti i skica sustava

Najidealnija arhitektura našeg sustava jest **web aplikacija**, jer ne zahtjeva od korisnika nikakvu instalaciju posebne programske podrške osim standardnog web preglednika kojeg većina korisnika ima instaliranog na svojim računalima. Dakle, naša će aplikacija morati zadovoljavati standarde današnjih web poslužitelja.

Za rad aplikacije bit će bitna baza podataka u kojoj ćemo bilježiti i čuvati sve važne podatke potrebne za rad i funkcioniranje sustava. Aplikacija će koristiti prikladni DAO za pristup bazi podataka.

Najvažniji podsustavi arhitekture našeg sustava su:

1. Web poslužitelj
2. Web aplikacija
3. Baza podataka



Slika 6.1 Podsustavi arhitekture sustava

Web poslužitelj

Web poslužitelj temeljni je dio bilo kojeg sustava na arhitekturi web aplikacije. Njegova je osnovna zadaća procesiranje zahtjeva korisnika kroz HTTP, osnovni mrežni protokol kojim se dijele informacije na World Wide Web-u.

Danas postoje razne kvalitetne programske implementacije web poslužitelja. Naša aplikacija konkretno će se izvoditi na *Apache Tomcat* web poslužitelju.

Upute za instalaciju i korištenje nalazit će se u prilogu napisanom nakon implementacije.

Web aplikacija

Web aplikacija je prozor kroz koji korisnik komunicira sa sustavom. Ona korisničke zahtjeve oblikuje u operacije sustava. Aplikacija kroz bazu podataka upravlja sa svim podacima potrebnim za rad sustava. Web aplikacija sama stvara i puni bazu podataka po potrebi.

Još jedna zadaća web aplikacija je vizualna prezentacija sustava korisniku. Za to se oslanja na web preglednik. Sama aplikacija priprema vizualne informacije u obliku HTML dokumenata dok web preglednik prikazuje te HTML dokumente u obliku web stranica.

Baza podataka

Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu.

Naša baza podataka sastoji se od sljedećih entiteta:

- Utakmica
- Događaj
- Igrači
- Pozicija
- Tip događaja

- Virtualne ekipe
- Korisnik
- Klubovi
- Ogranak
- Poredak lige
- Kolo
- Poredak virtualne lige

Utakmica – Ovaj entitet sadržava sve važne informacije o utakmicama koje će se odigravati u ligi. Sadrži atribute: Šifra domaćina, Šifra gosta, Datum održavanja utakmice, Šifra najboljeg igrača u utakmici, Šifra utakmice te šifre više događaja koji su se odigrali u utakmici.

Ovaj entitet je u vezi One-to-One s entitetom **Igrači** preko atributa šifre najboljeg igrača te u vezi One-to-Many s entitetom **Događaj** preko šifre događaja.

Događaj – Ovaj entitet sadržava informacije o odigranim događajima u utakmicama. Sadrži atribute: Šifra događaja, Ime događaja, Šifra tipa događaja te Šifra igrača koji je izvršio taj događaj.

Ovaj entitet je u vezi Many-to-One s entitetom **Utakmica** preko šifre događaja, One-to-One s entitetom **Tip Događaja** preko šifre tipa i u vezi One-to-One s entitetom **Igrači** preko šifre igrača koji je sudjelovao u događaju.

Igrači – Ovaj entitet sadržava osnovne informacije o igračima u rukometnim klubovima. Svaki igrač ima atribute: Šifra igrača, Ime igrača, Prezime igrača, Šifra kluba u kojem igra, Starost igrača, Cijena igrača, Šifra pozicije te ostvareni bodovi.

Ovaj entitet je u One-to-One vezama s entitetima **Utakmica**, **Događaj** i **Pozicija** preko atributa šifre igrača i šifre pozicije. Također je u vezi Many-To-One s entitetom **Klubovi** preko šifre igrača.

Pozicija – Ovaj entitet sadržava sve moguće pozicije u kojima igrač može igrati u rukometu. Pozicija ima atribute Šifre pozicije i Ime pozicije.

Ovaj entitet je u vezi One-To-One s entitetom **Igrači** preko atributa šifre pozicije.

Tip događaja – Ovaj entitet sadržava sve moguće tipove događaja koje se mogu dogoditi u jednoj utakmici. Tip događaja sadrži atribute Šifre tipa i Ime tipa.

U vezi je One-To-One s entitetom **Događaj** preko atributa šifre tipa.

Virtualne ekipe – Predstavlja virtualne ekipe koje svaki korisnik može stvoriti da bi se natjecao u virtualnoj ligi. Sadrži atribute: Šifra ekipe, Naziv ekipe, Trenutni proračun.

Preko atributa šifre ekipe entitet je u vezi One-To-One s entitetom **Korisnik**.

Korisnik - Entitet korisnik služi za pohranjivanje i čuvanje podataka koje opisuju korisnika u našem opisanom sustavu web aplikacije za rukometnu ligu. Sadrži sljedeće atribute: korisničko ime, korisnička lozinka, šifra korisnika, ime, prezime, e-mail, država, šifra ekipe, šifra podržanog kluba, tip korisnika i šifra virtualne ekipe.

Entitet je u vezi One-To-One s entitetom **Virtualna Ekipa** preko atributa šifra ekipe, u vezi Many-To-One s entitetom **Klubovi** te u vezi Many-To-One s entitetom **Poredak virtualne lige**.

Klubovi - Entitet Klubovi služi nam za pohranjivanje i čuvanje podataka koje opisuju klub u našoj rukometnoj ligi. Sadrži atribute: Šifra kluba, šifra ogranka, ime kluba, vrijednost kluba, broj igrača, ostvareni bodovi.

Entitet je u vezi One-To-Many s entitetom **Korisnik** preko atributa šifre kluba, Many-To-One s entitetom **Ogranak** preko šifre ogranka i Many-To-One vezi s entitetom **Poredak Lige** preko šifre kluba.

Ogranak - Entitet ogranak predstavljanja ogranak rukometne lige. Sadrži dva atributa: Šifra ogranka i Ime ogranka.

Preko atributa Šifra ogranka entitet je u vezi One-To-Many s entitetom **Klubovi**.

Poredak lige - Entitet Poredak Lige predstavlja poredak rukometne lige. Sastoje se od dva atributa: Šifra kluba i Šifra kola. Preko atributa Šifra kluba entitet je u vezi One-To-Many s entitetom **Klubovi**, a preko atributa šifre kola je u vezi One-To-Many s entitetom **Kolo**.

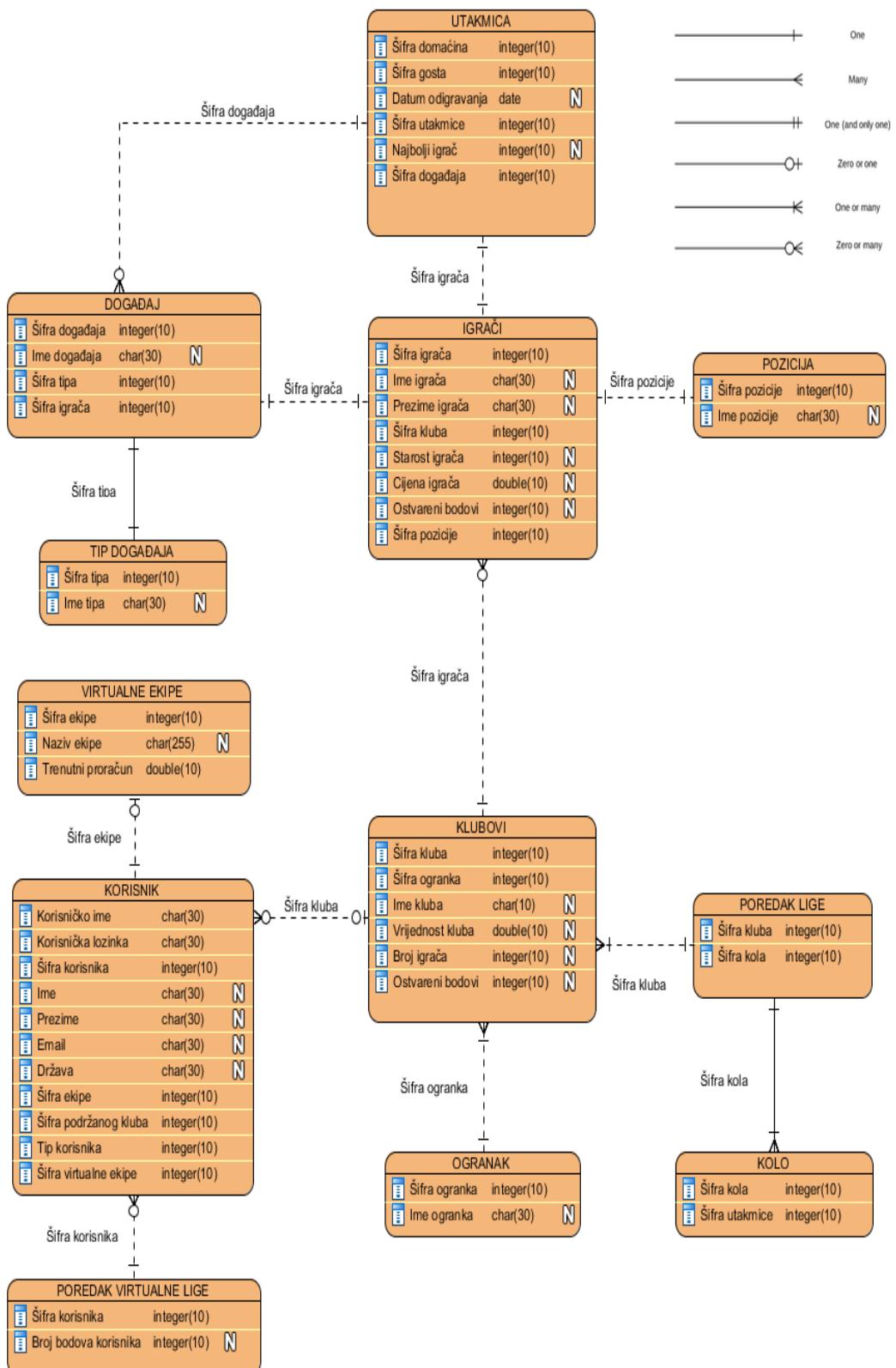
Kolo - Entitet Kolo predstavlja jedno kolo odigravanja utakmica rukometne lige.

Sastoje se od dva atributa: Šifra kola i Šifra utakmice.

Preko atributa šifre koja je u vezi Many-To-One s entitetom **Poredak lige**.

Poredak virtualne lige - Entitet Poredak virtualne lige predstavlja poredak virtualnih ekipa virtualne lige. Sastoje se od dva atributa: Šifra korisnika i Broj bodova korisnika.

Preko atributa šifre korisnika entitet je u vezi One-To-Many s entitetom **Korisnik**.



Slika 6.2 ER dijagram baze podataka

6.2. Dijagram razreda s opisom

Razredi su podijeljeni u tri velike cjeline: podpaket **modeli**, podpaket **dao** i podpaket **simulator**. U podpaketu **modeli** nalaze se svi potrebni modeli podataka koji se koriste u izradi aplikacije, u podpaketu **dao** nalaze razredi zaduženi za komunikaciju s bazom podataka, a u podpaketu **simulator** nalazi se sve potrebno za simulaciju utakmica.

Podpaket **modeli** je podijeljen također u dvije skupine.

Prva skupina zove se **Liga** i sadrži sve razrede koji se odnose na samu implementaciju virtualne i prave rukometne lige.

Liga je osnovni razred koji predstavlja stvarnu rukometnu ligu. Sadrži klubove koji sudjeluju, te listu kola.

Kolo predstavlja jedno kolo lige. Sadrži listu utakmica koje se održavaju u tom kolu kao i **Ligu** u kojoj se nalazi.

Utakmica predstavlja jednu utakmicu. Sadrži **Klub** domaćina, **Klub** gosta, datum odigravanja utakmice, listu **Događaja** na utakmici te najboljeg **Igrača** (mvp u dijagramu) i **Kolo** u kojem se odigrava.

Klub predstavlja jedan od klubova koji se natječu u stvarnoj ligi. Sadrži naziv kluba, ogrank u kojem se nalazi, vrijednost kluba i listu **Igrača** koji nastupaju za njega.

Igrač predstavlja jednog igrača koji sadrži ime, prezime, godine, vrijednost i **Poziciju** na kojoj igra. Uz to, ima i **Klub** kojem pripada.

Pozicija je posebni enum koji se odnosi na moguće različite pozicije koje igrač može igrati.

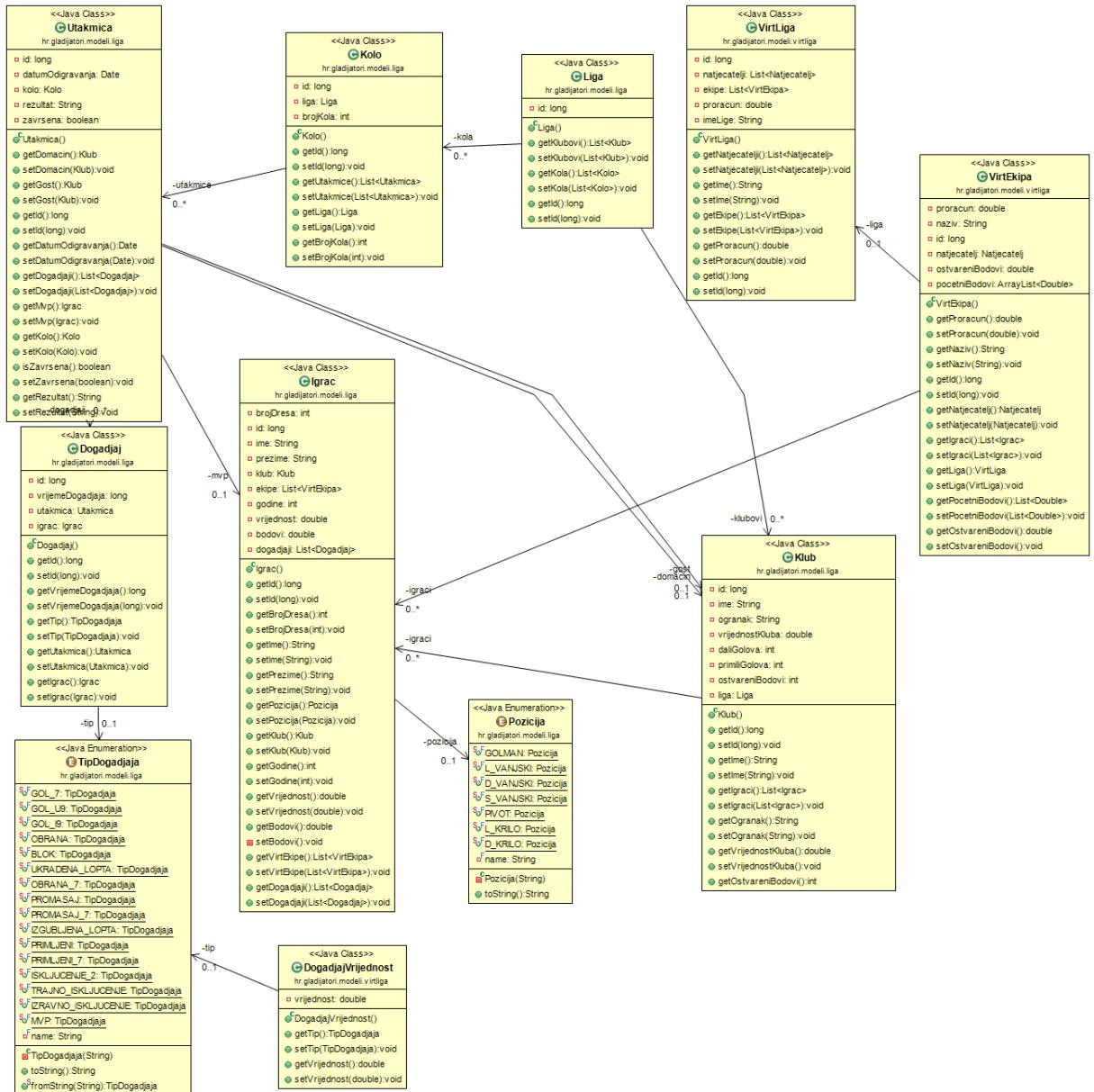
Dogadjaj se odnosi na pojedini događaj koji se može dogoditi tijekom utakmice. Sadrži vrijeme događaja (sekundu u kojoj se dogodio), **TipDogađaja** koji određuje koja se vrsta događaja dogodila i **Utakmicu** na kojoj se dogodio.

TipDogadjaja je enum koji sadrži sve moguće događaje koji se mogu dogoditi na utakmici.

VirtLiga predstavlja virtualnu ligu u kojoj se **Natjecatelji** natječu. Sadrži listu **Natjecatelja** koji sudjeluju, te predefinirani proračun za svaku **VirtEkipu**.

VirtEkipa predstavlja jednu ekipu koja sudjeluje u virtualnoj ligi. Sastoji se od trenutnog proračuna, naziva, te liste **Igrača** koji su prijavljeni za tu ekipu i **VirtLige** kojoj pripada.

DogadjajVrijednost predstavlja učitanu vrijednost jednog događaja, tj. koji broj bodova igrač dobija ili gubi u odnosu na određeni **TipDogadjaja**.



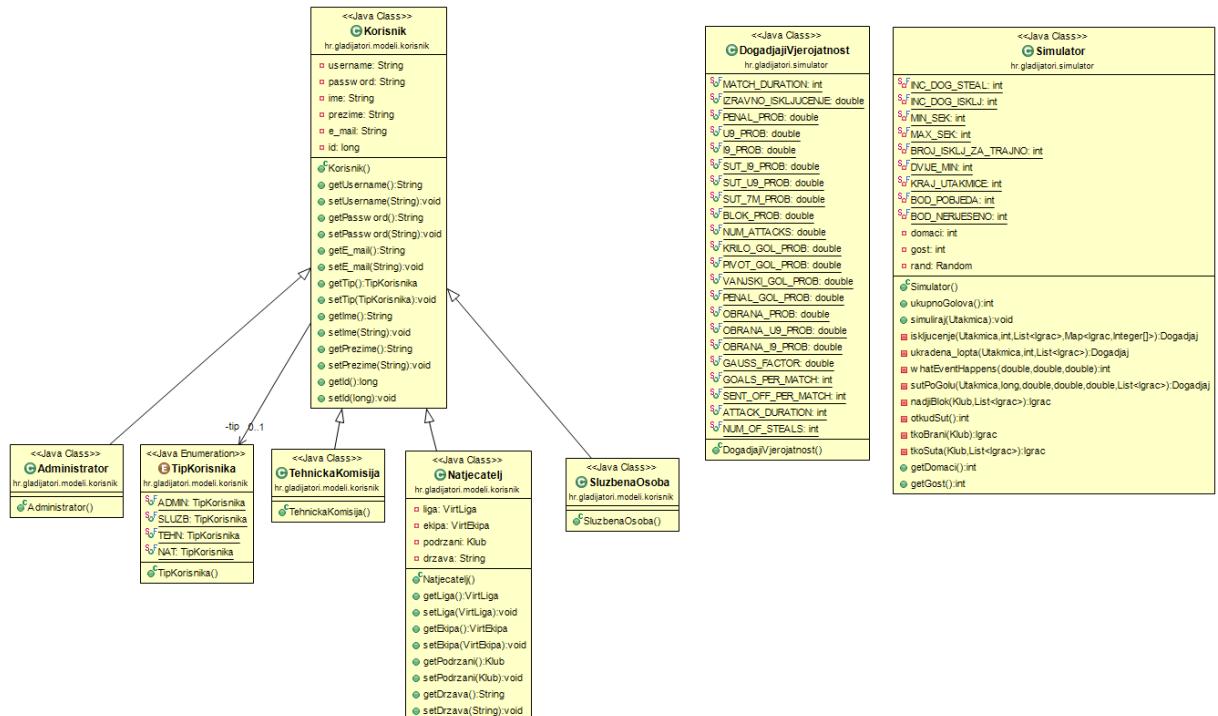
Slika 6.3 Dijagram razreda podpaketa modeli.liga

Druga skupina modela se odnosi na korisnike. Sadrži izvorni razred **Korisnik** iz kojeg su izvedeni podrazredi koji se odnose na pojedinu vrstu korisnika koje razlikujemo. Svaki od izvedenih korisnika ima posebne metode i razinu dozvola u

aplikaciji. Uveden je i poseban **TipKorisnika** koji služi za jednostavno razlikovanje različitih korisnika.

Korisnik kao izvorni razred sadrži sve zajedničke atribute (korisničko ime, lozinku, ime, prezime, e-mail) korisnika i nudi metode za postavljanje i dohvaćanje istih.

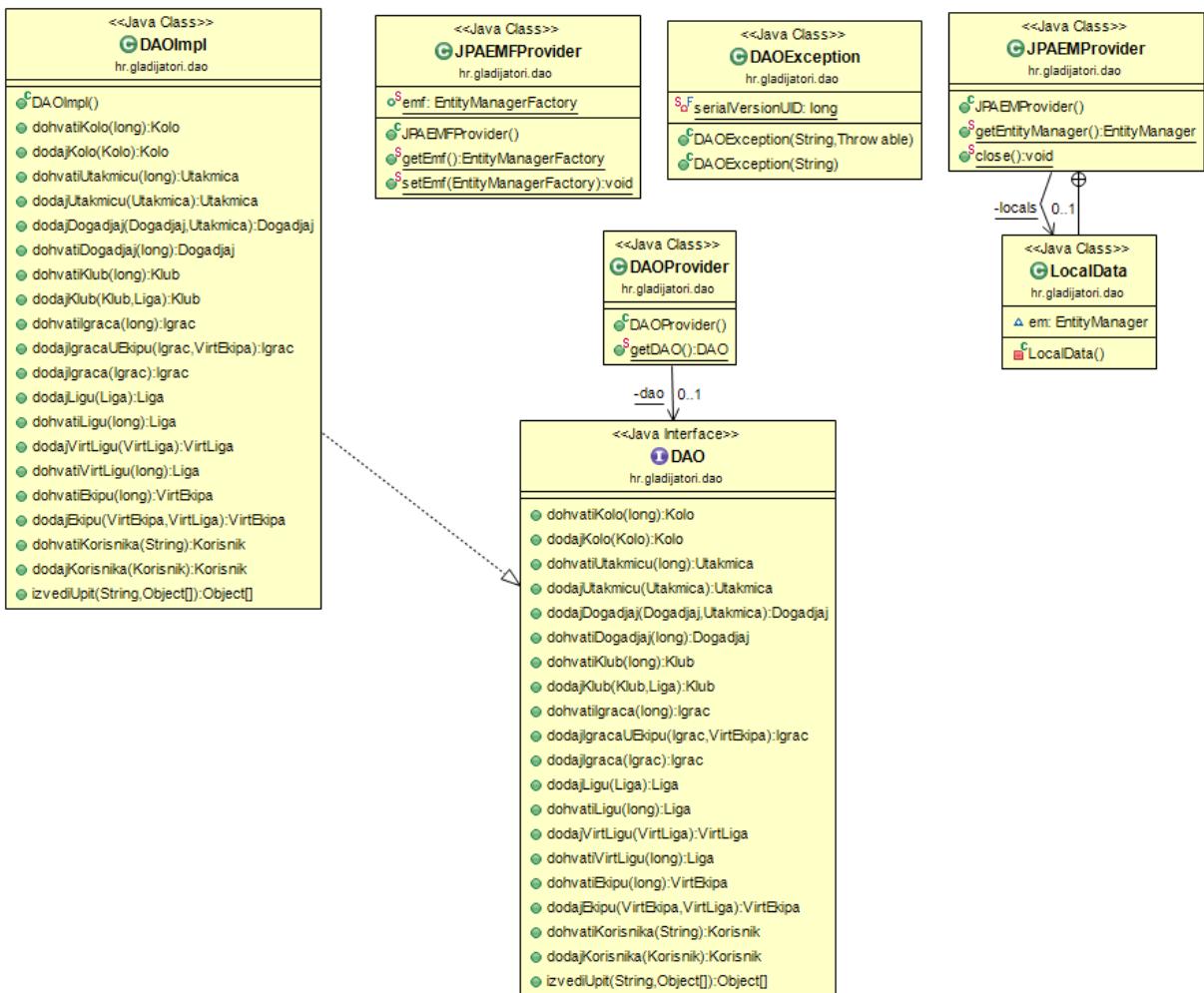
Ostali razredi korisnika se odnose na pojedine tipove korisnika koji se mogu pojaviti u aplikaciji, i dodijeljena im je prikladna razina pristupa aplikaciji.



Slika 6.4 Dijagram razreda podpaketa modeli.korisnik i simulator

Druga velika cjelina cijele aplikacije je **dao**, odnosno dio za upravljanje podatcima i povezivanje s bazom podataka. Ključno sučelje je **DAO** koje sadrži sve potrebne metode za upravljanje bazom podataka. Te se metode odnose na dodavanje objekata u bazu, dohvaćanje objekata u bazu, ali i mogućnost izvođenja korisničkog SQL upita. **DAOImpl** je razred koji implementira sve potrebne metode u skladu s korištenom bazom podataka.

Cijeli **DAO** sloj zamišljen je da radi kao Singleton, postojala bi samo jedna instanca **DAOImpl** koja se može dohvatiti iz bilo kojeg dijela programa, a taj objekt bi pružao **DAOProvider**. Ostali razredi služe za inicijalizaciju i prekid veze s bazom podataka.



Slika 6.5 Dijagram razreda podpaketa DAO

Treća i zadnja cjelina je **simulator**. Sastoji se od potrebnih razreda za simulaciju utakmice, a to su: **Simulator** i **DogadjajVjerojatnost**.

DogadjajVjerojatnost je razred koji sadrži sve potrebne konstante potrebne za slučajno odigravanje utakmica kao što su: prosječaj broj golova, prosječan broj isključenja, vjerojatnost da će golman obraniti, vjerojatnost da će igrač dati gol (i koji) i tome slično.

U **Simulatoru** se nalazi ostvarenje algoritma koji slučajno generira događaje na utakmici. Sadrži razne privatne metode koje služe za odabir koji će igrač šutati, koji golman brani, tko je ukrao loptu, tko je izgubio, je li bilo isključenje, postavlja

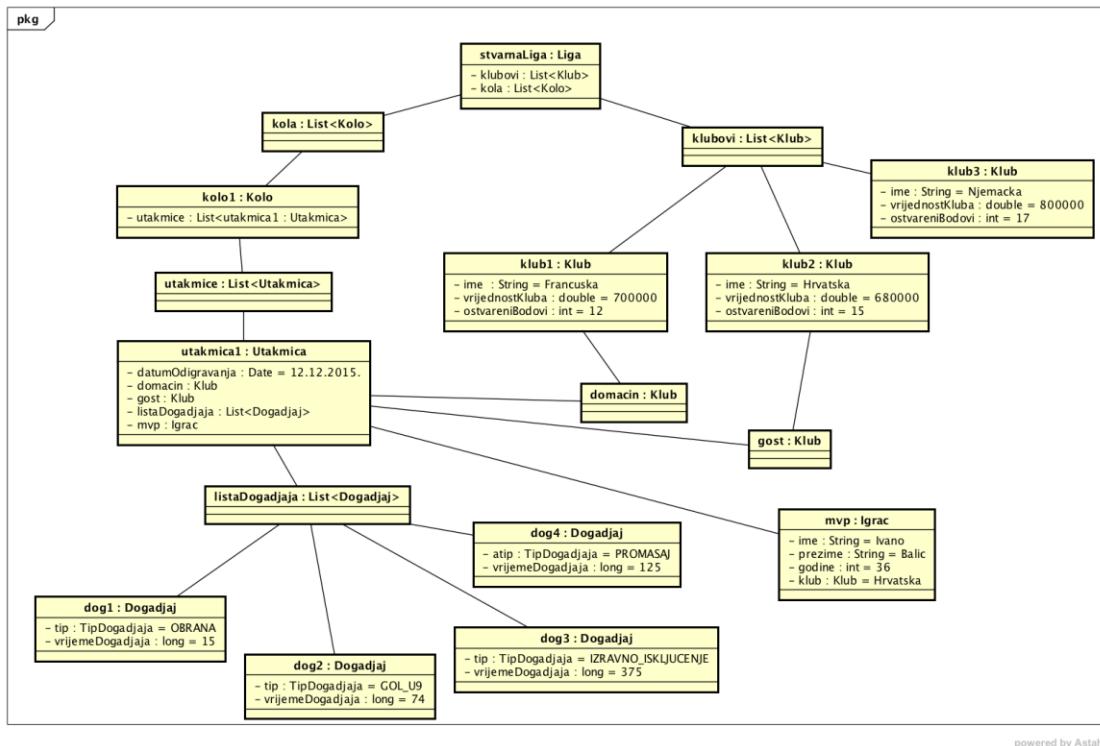
odgovarajuće vjerojatnosti (pojačava ili smanjuje u ovisnosti o vrijednosti igrača i vrijednosti ekipa).

6.3. Dijagram objekata

Kako bismo prikazali ponašanje sustava u stvarnom vremenu, prikazat ćemo kako naš razredni model izgleda u dvije specifične situacije. Za prvi primjer uzeli smo jednu stvarnu ligu koja se sastoji od liste **Klubova** i liste **Kola**. Lista **Kola** sadrži instance razreda **Kolo**, a između ostalog i prikazani objekt **kolo1**. Svako kolo ima listu utakmica koje sadrži, i tako **kolo1** sadrži instancu razreda **Utakmica**, **utakmica1**. **Utakmica1** se sastoji od datuma održavanja, dva **Kluba** (domaćin i gost), liste **Dogadjaja** utakmice te najboljeg igrača. Lista instanci **Dogadjaj** utakmice, puni se dok se utakmica odvija, a svaki **Dogadjaj** ima svoj određeni tip(GOL, OBRANA, PROMASAJ) te varijablu vrijemeDogadjaja koja označava u kojoj sekundi utakmice se događaj dogodio. Varijabla **mvp** označava najboljeg igrača utakmice i pokazuje na instancu razreda **Igrac** koja sadrži igračevu ime, prezime, njegove godine te njegov klub.

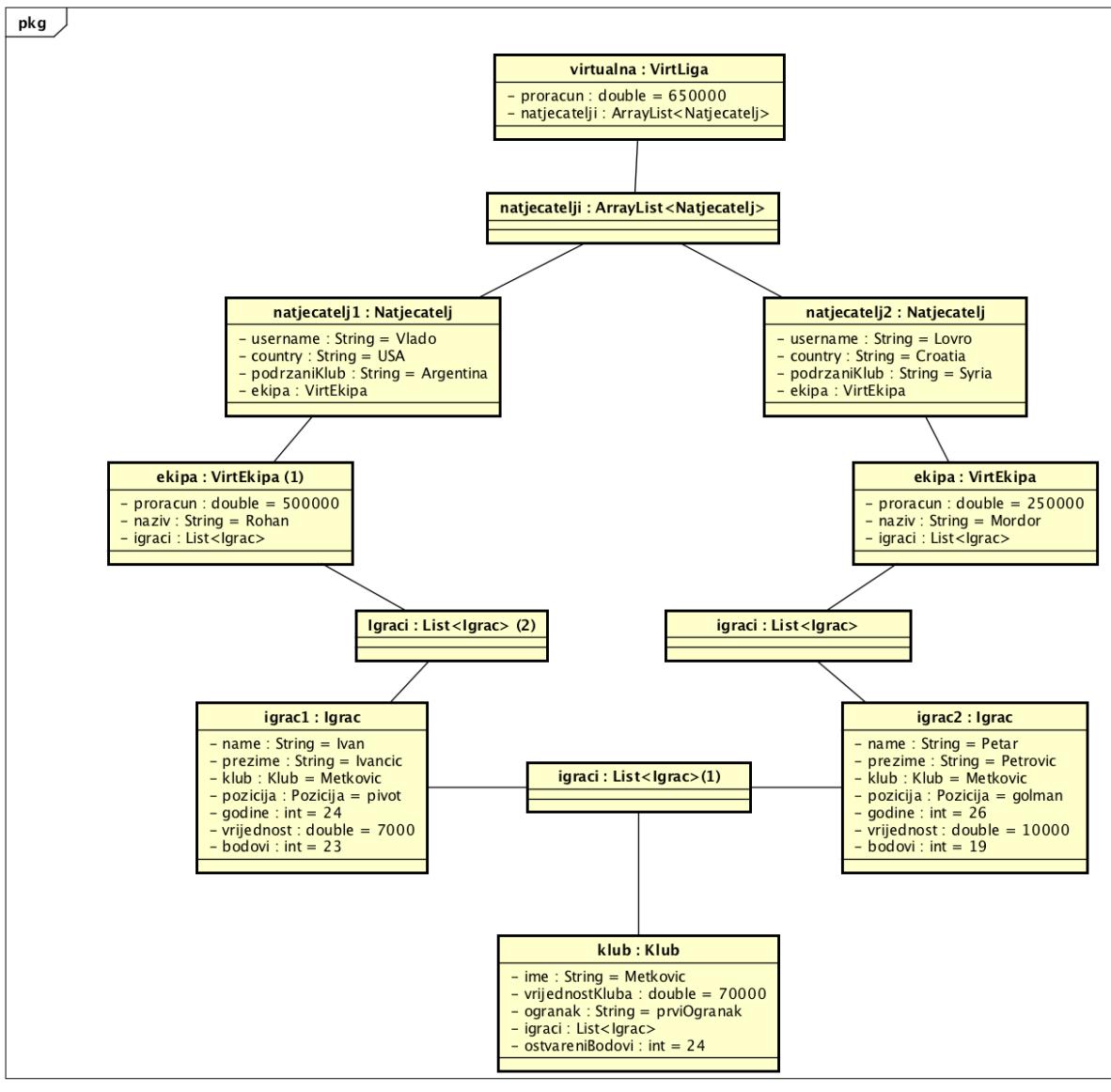
Uz to, svaka liga ima i listu instanci razreda **Klub**, u kojoj se nalaze svi klubovi koji trenutno igraju u toj ligi, sa svojim podacima: ime, vrijednostKluba, ostvareniBodovi.

Ovi objekti interno sadrže još neke varijable (npr. Liga, Klub, Utakmica, Igrac i Dogadjaj imaju varijablu id koja služi za rukovanje podacima iz baze podataka) koje, zbog sažetosti prikaza, nismo ovdje prikazali.



Slika 6.6 Dijagram objekata Stvarna liga

Kao drugi primjer, prikazat ćemo jednu virtualnu ligu. Za razliku od stvarne, virtualna liga se izvodi između registriranih **Korisnika** na sustavu, koji slažu svoje **Ekipe**, stvarnim **Igracima**, čije performanse rastu ili padaju ovisno o njihovom učinku u stvarnoj **Ligi**. Tako i prikazana **VirtLiga**, virtualna, ima svoj proračun te sadrži listu **Natjecatelja** koji u njoj sudjeluju. Natjecatelji su korisnici, koji se prijavljuju na sustav. Klasa **Natjecatelj** sadrži osobne podatke korisnika (username, country, podrzaniKlub te još neke koje na dijagramu nisu prikazane poput password, e-mail) i ekipu koju je Natjecatelj odabrao. Ekipa natjecatelja je instanca razreda **VirtEkipa**.



powered by Astah

Slika 6.7 Dijagram objekata Virtualna Liga

Ona sadrži proračun i naziv ekipe te listu **Igraca**. Svaki od **Igraca** ima svoje ime, prezime, stvarni **Klub** u kojem igra, svoju poziciju na kojoj igra, vrijednost (cijenu), te bodove koje je do sada ostvario u stvarnoj **Ligi**. Na dijagramu smo prikazali slučaj kada dva različita **Natjecatelja** odaberu **Igrače** iz istog stvarnog **Kluba**.

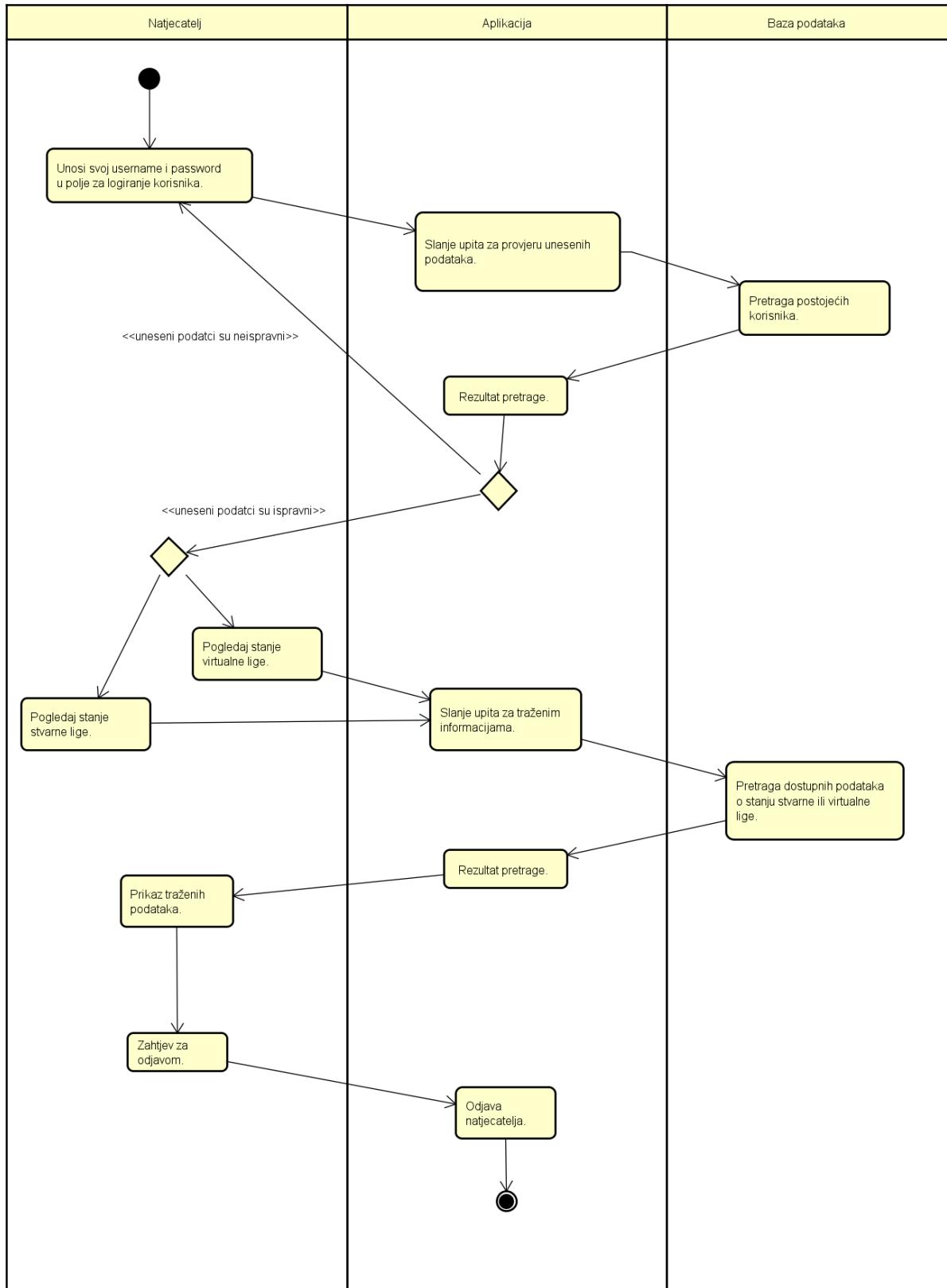
6.4. Ostali UML dijagrami

Dijagram aktivnosti

Dijagram aktivnosti služi za prikaz aktivnosti registriranog **Natjecatelja**.

On unosi svoje korisničko ime i lozinku te je nakon uspješne provjere unesenih podataka usmjeren na stranicu gdje ima mogućnosti pregleda stanja stvarne lige (**UC1: PogledajStvarnoStanje**) te pregleda stanja virtualne lige (**UC4: PogledajStanjeVirtualneLige**).

Nakon pregleda željenih podataka, natjecatelj se odjavljuje iz aplikacije.



powered by Astah

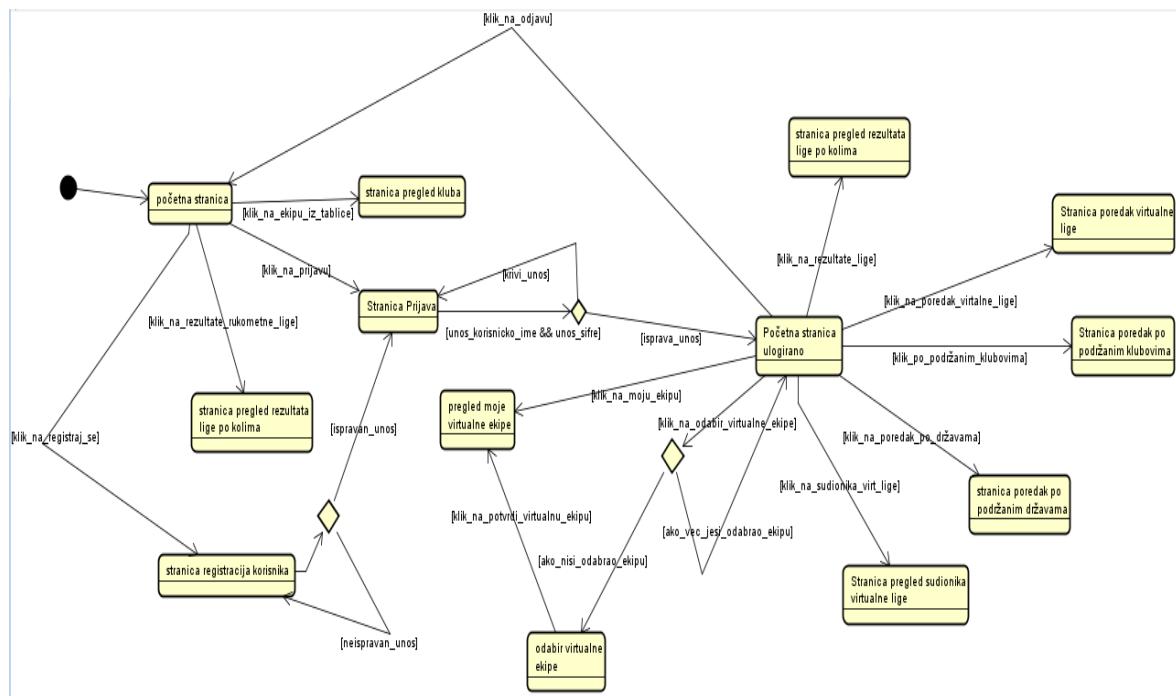
Slika 6.8 Dijagram aktivnosti registriranog natjecatelja

Dijagram stanja

Dijagramom stanja modeliramo ponašanje sustava koje je upravljanu događajima. Na Slici 6.9 korisnik se prijavljuje u sustav dolaskom na početnu stranicu i birajući stranicu prijava („Stranica prijava“). Korisnik pritom unosi svoje korisničko ime i lozinku te, uz uvjet da je pravilno napisao podatke, ulazi na početnu stranicu u prijavljenom stanju („Početna stranica Ulogirano“).

Pritom korisniku stoji na raspolaganju mnogo opcija; može pregledati rezultate lige po kolima („Stranica pregled rezultata lige po kolima“), poredak virtualne lige („Stranica poredak virtualne lige“), poredak korisnika po podržanim klubovima („Stranica poredak po podržanim klubovima“), sudionike virtualne lige („Stranica pregled sudionika virtualne lige“), svoju vlastitu virtualnu ekipu („Pregled moje virtualne ekipe“). Također, u slučaju da korisnik već nije, može složiti svoju virtualnu ekipu („Odabir virtualne ekipe“) te se može odjaviti, što ga ponovno odvodi na početnu stranicu („Početna stranica“).

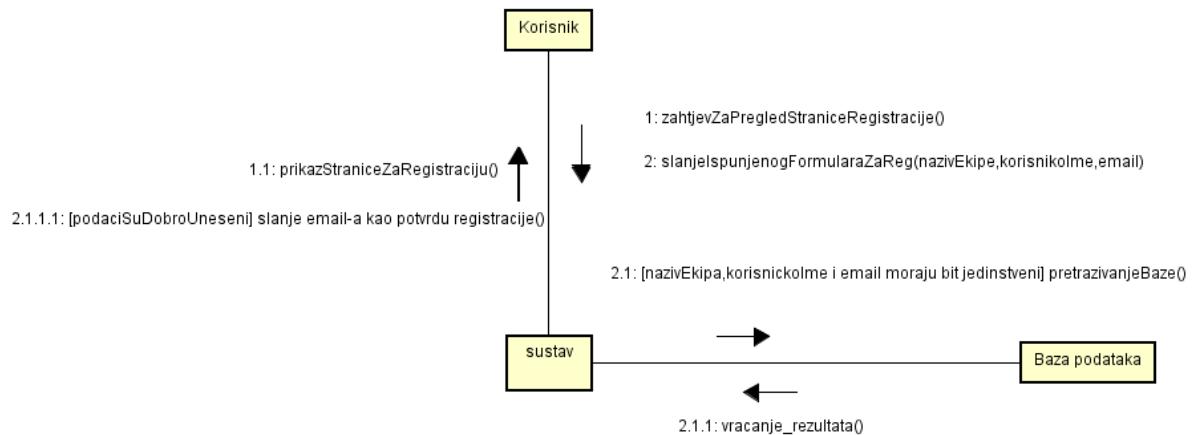
Ukoliko se korisnik ne prijavi, ostaje mu na raspolaganju samo pregled tablice rukometne lige („Početna stranica“) i rezultati lige po kolima („Stranica pregled rezultata lige po kolima“). Također, može se i registrirati („Stranica registracija korisnika“). U stanju „Stranica registracija korisnika“ korisnik bira svoje korisničko ime i lozinku te, u slučaju da korisničko ime već ne postoji u bazi podataka, se registracija prihvata i korisnika vodi dalje do stranice za prijavu korisnika („Stranica Prijava“). U slučaju da unos nije bio dobar, ispisat će se da njegovo željeno korisničko ime već postoji i ostavit će ga i dalje na istoj stranici.



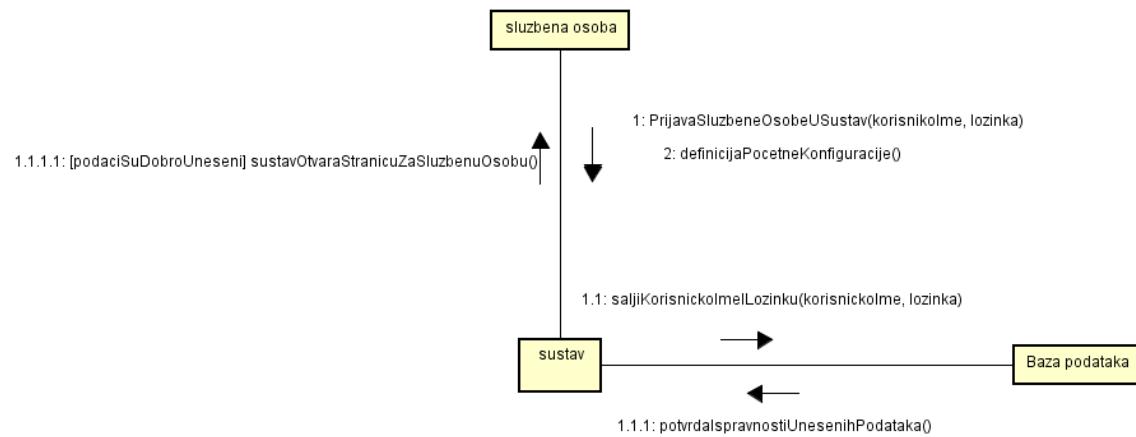
Slika 6.9 Dijagram stanja registriranog natjecatelja

Komunikacijski dijagram

Komunikacijski dijagram na slici 6.10 prikazuje registraciju posjetitelja. Posjetitelj učitava stranicu za registraciju te ispunjava tražene informacije (korisničko ime i lozinku). Sustav te podatke proslijeđuje bazi podataka za provjeru, a ona vraća rezultat te provjere. Sustav pritom korisniku daje informaciju o stanju njegove registracije. U slučaju dobre registracije, korisnik dobiva email kao potvrdu registracije.

**Slika 6.10 Komunikacijski dijagram registriranje korisnika**

Drugi komunikacijski dijagram na slici 6.11 prikazuje uvođenje početne konfiguracije od strane službene osobe. Prvo se službena osoba prijavljuje sa svojim korisničkim podacima koje sustav prima i prosljeđuje dalje u bazu podataka za provjeru ispravnosti podataka. Baza podataka vraća potvrdu o ispravnosti unesenih podataka i sustav na osnovu te potvrde, u slučaju da je pozitivna, otvara službenoj osobi stranicu za službenu osobu. Konačno, službena osoba definira početnu konfiguraciju i time završava učitavanje početne konfiguracije.

**Slika 6.11 Komunikacijski dijagram unošenje početne konfiguracije**

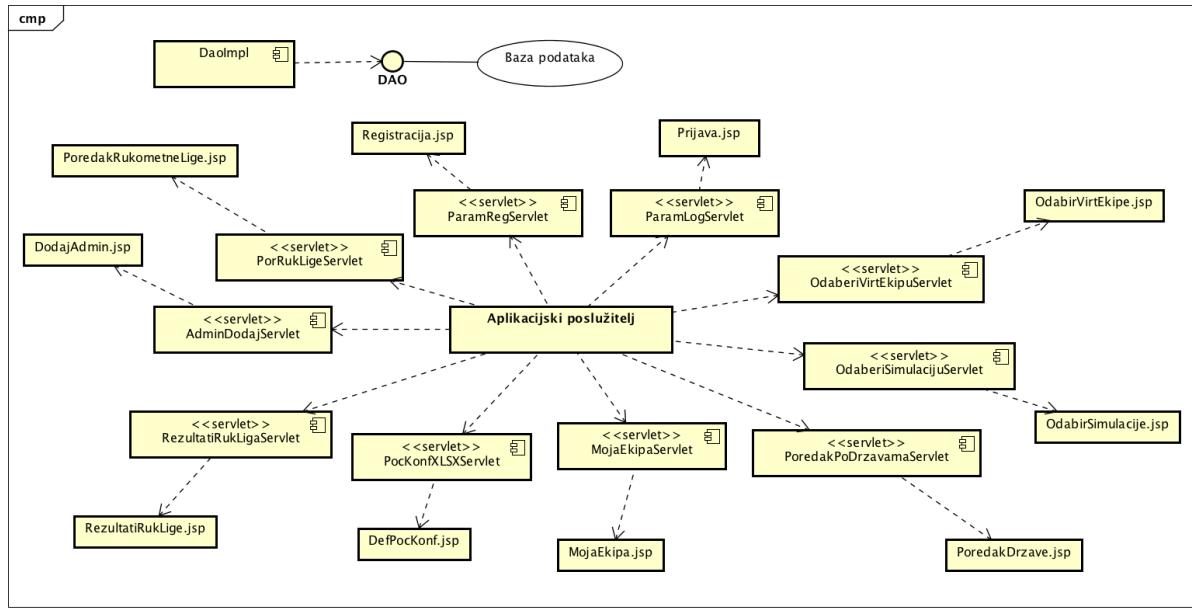
Dijagram komponenti

Dijagram komponenti prikazuje strukturu sustava pomoću komponenti od kojih je sustav sastavljen. Naš sustav ima osnovnu strukturu web aplikacije izrađene u Javi. Korisnik unosom URL-a u preglednik ili pritiskom na jednu od poveznica na web stranici web aplikacije ukazuje web serveru koji servlet treba obraditi. Servlet tada obavlja posao vezan uz obradu podataka iz baze ili slično, te rezultirajuće podatke prosljeđuje .jsp datoteci pomoću koje server generira odgovarajuću .html datoteku koju šalje klijentskom pregledniku koji koristi .html datoteku za prikaz web stranice koju klijent očekuje. Daolmpl je naša implementacija sučelja DAO, koju servleti koriste za rad s bazom podataka. Naša aplikacija sastoji se od preko 40 servleta, stoga smo na dijagramu naveli samo jedan manji podskup servleta i .jsp dokumenata.

Ovdje su funkcije prikazanih servleta:

- **AdminDodajServlet** – dodavanje korisničkog računa tehničke komisije ili službene osobe od strane administrator,
- **ParamRegServlet** – omogućuje registriranje novog korisnika,
- **ParamLogServlet** – omogućuje prijavljivanje postojećeg korisnika,
- **OdaberiVirtEkipuServlet** – odabir virtualne ekipe trenutnog natjecatelja,
- **MojaEkipaServlet** – pregled sastava virtualne ekipe trenutno prijavljenog natjecatelja,
- **OdaberiSimulacijuServlet** – odabir utakmica koje će se simulirati pomoću administratorskog računa,
- **PorRukLigeServlet** – pregled poretku stvarne rukometne lige,
- **PoredakPoDrzavamaServlet** – rangiranje natjecatelja po pripadnim državama,
- **RezultatiRukLigaServlet** – pregled rezultata odigranih utakmica rukometne lige,

- **PocKonfXLSXServlet** – definiranje početne konfiguracije od strane službene osobe.

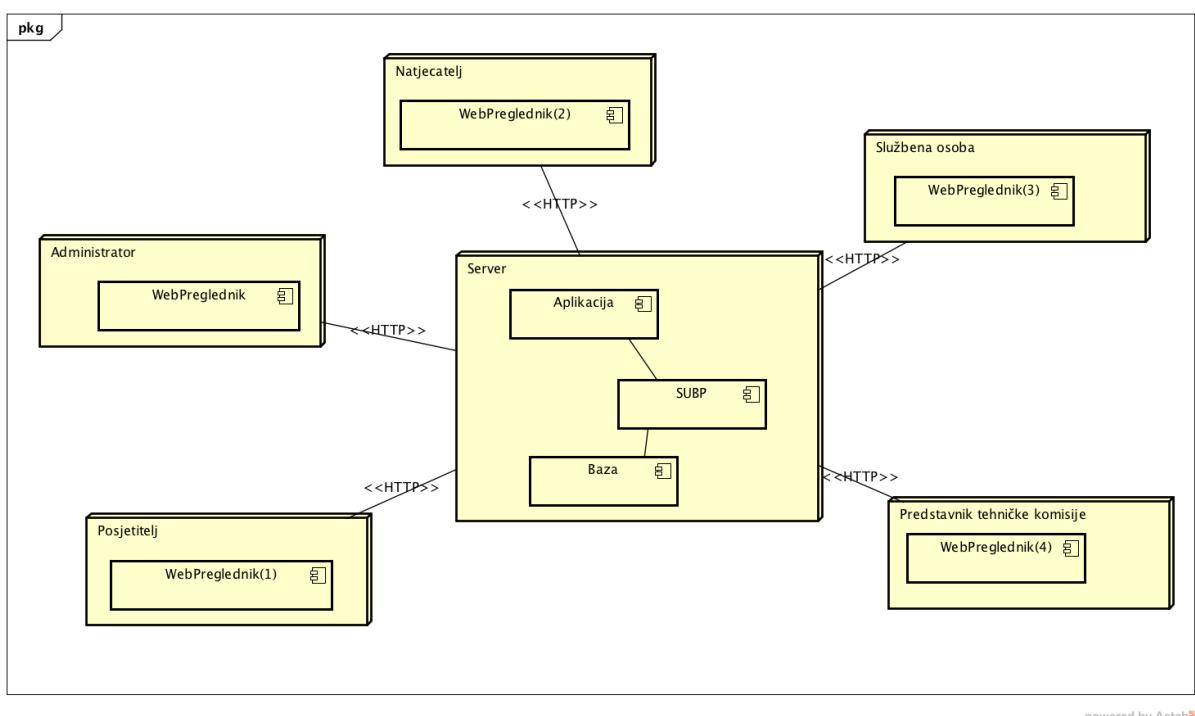
**Slika 6.12 Dijagram komponenti**

powered by Astah

7. Implementacija i korisničko sučelje

7.1. Dijagram razmještaja

Na slici 7.1 prikazana je generalna topologija sustava. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala klijenta (kod nas su to sve vrste korisnika) i poslužitelja odvija se preko HTTP veze. Unutar poslužitelja aplikacija, preko sustava za upravljanje bazom podataka, komunicira s bazom, koja može biti fizički i logički odvojena od same aplikacije.



Slika 7.1 Dijagram razmještaja

7.2. Korištene tehnologije i alati

Pri izradi web aplikacije korištena je razvojna okolina *Eclipse JEE Mars* i radni okvir *Servlet API*.

Za rad s bazom podataka korišten je *Apache Derby* sustav za upravljanje relacijskim bazama te radni okvir *Hibernate ORM* za mapiranje objektno orijentiranog modela na relacijsku bazu podataka.

Za izvedbu web poslužitelja korišten je *Apache Tomcat* web poslužitelj.

Prilikom razvoja korišten je *git* sustav za kontrolu verzija na repozitoriju putem web usluge *Bitbucket*.

Pri izradi UML dijagrama korišten je alat Astah Community.

Servlet API

Servlet API je temelj razvoja skoro svih Java Web tehnologija. Servleti pružaju osnovne mehanizme stvaranja Java razreda u web aplikacijama.

Servleti mogu odgovarati na HTTP zahtjeve te upravljati sjednicama.

Apache Derby

Apache Derby je jednostavan sustav za upravljanje relacijskim bazama podataka s vrlo malim instalacijskim otiskom (osnovna instalacija zauzima 2.6 MB na disku).

Njegova je osnovna zadaća pružanje jednostavnog ugrađenog pogona za rad s relacijskim bazama podataka koji podržava JDBC, SQL te razna aplikacijska sučelja.

Hibernate ORM

Hibernate ORM je radni okvir koji mapira objektni model s relacijskim modelom baze podataka u programskom jeziku Java.

Njegova je osnovna zadaća mogućnost jednostavnog povezivanja Java razreda s ekvivalentnim tablicama u bazi podataka. Također nudi sučelje velike apstrakcije za rad sa SQL upitima i automatski obrađuje konverziju rezultate SQL upita u java objekte.

Apache Tomcat

Apache tomcat je open-source web poslužitelj koji implementira razne Java EE specifikacije kao što su Java Servlet, JSP i WebSocket te nudi HTTP web poslužitelj u čistoj Javi.

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost

Baza podataka

Na slikama x vidimo prikaz sučelja *DAO.java* koje sadrži sve metode potrebne za rad s bazom podataka koje su ostvarene u *DAOImpl.java* razredu te *JPAEMProvider.java* razred koji koristi javax.persistence paket koji nudi mogućnost provođenja transakcija prema i iz baze podataka. Metode *DAOImpl.java* razreda ostvarene su uporabom *JPAEMProvider.java* razreda te su sve slične, a s obzirom da ih je mnogo, pružen je pregled sučelja uz jedan primjer programskog ostvarenja metode *DAO.java* sučelja, iz *DAOImpl.java* razreda.

```

3 public interface DAO {
4
5     public Kolo dohvatiKolo(long id);
6     public Kolo dodajKolo(Kolo kolo);
7
8     public Utakmica dohvatiUtakmicu(long id);
9     public Utakmica dodajUtakmicu(Utakmica utakmica);
10
11    public Klub dohvatiKlub(long id);
12    public Klub dohvatiKlub(String ime);
13    public void dodajKlub(Klub klub, Liga liga);
14    public List<Klub> dohvatiKlubove();
15
16    public Igrac dohvatiIgraca(long id);
17    public List<Igrac> dohvatiIgraca(String ime, String prezime);
18    public List<Igrac> dohvatiIgraca(Pozicija pozicija);
19    public void dodajIgracaUEkipu(Igrac igrac, VirtEkipa ekipa);
20    public void dodajIgraca(Igrac igrac);
21
22    public void dodajLigu(Liga liga);
23    public Liga dohvatiLigu(long id);
24
25    public void dodajVirtLigu(VirtLiga virtLiga);
26    public VirtLiga dohvatiVirtLigu(long id);
27
28    public VirtEkipa dohvatiEkipu(long id);
29    public void dodajEkipu(VirtEkipa ekipa, VirtLiga virtLiga);
30
31    public Korisnik dohvatiKorisnika(String username);
32    public void dodajKorisnika(Korisnik korisnik);
33
34    public Natjecatelj dohvatiNatjecatelja(String korime);
35    public VirtLiga dohvatiLigu(String ime);
36
37    public double dohvatiVrijednostDogadjaja(TipDogadjaja tip);
38    public DogadjajVrijednost postaviVrijednostDogadjaja(TipDogadjaja tip, double vrijednost);
39    public void dodajDogadjaj(Dogadjaj dogadjaj, Utakmica utakmica);
40    public Dogadjaj dohvatiDogadjaj(long id);
41
42    public List<Object> izvediUpit(String upit, Object... parametri);
43    public void izvediUpdateUpit(String upit, Object... parametri);
44    public void izvediNativeUpdateUpit(String upit, Object... parametri);
45 }
```

Slika 7.2 Implementacija DAO.java sučelja

```

3 public class JPAEMProvider {
4
5     private static ThreadLocal<LocalData> locals = new ThreadLocal<>();
6
7     public static EntityManager getEntityManager() {
8         LocalData ldata = locals.get();
9         if(ldata==null) {
10             ldata = new LocalData();
11             ldata.em = JPAEMFProvider.getEmf().createEntityManager();
12             ldata.em.getTransaction().begin();
13             locals.set(ldata);
14         }
15         return ldata.em;
16     }
17
18     public static void close() throws DAOException {
19         LocalData ldata = locals.get();
20         if(ldata==null) {
21             return;
22         }
23         DAOException dex = null;
24         try {
25             ldata.em.getTransaction().commit();
26         } catch(Exception ex) {
27             dex = new DAOException("Unable to commit transaction.", ex);
28         }
29         try {
30             ldata.em.close();
31         } catch(Exception ex) {
32             if(dex!=null) {
33                 dex = new DAOException("Unable to close entity manager.", ex);
34             }
35         }
36         locals.remove();
37         if(dex!=null) throw dex;
38     }
39
40     private static class LocalData {
41         EntityManager em;
42     }
43 }
```

Slika 7.3 Implementacija JPAEMProvider.java razreda

```

7 @Override
8 public Klub dohvatiKlub(String ime) {
9     try {
10         Klub klub = (Klub) JPAEMProvider
11                         .getEntityManager()
12                         .createQuery("select b from Klub as b where b.ime=:n")
13                         .setParameter("n", ime)
14                         .getSingleResult();
15         return klub;
16     } catch (NoResultException e) {
17         return null;
18     }
19 }
```

Slika 7.4 Isječak implementacije DAOImpl.java razreda

Simulator

Simulator.java i *DogadjajiVjerojatnosti.java* su razredi koji pružaju mogućnosti simuliranja utakmice. *DogadjajiVjerojatnosti.java* razred sadrži konstante koje predstavljaju vjerojatnost za određenu situaciju tijekom utakmice koje u svojoj implementaciji koristi razred *Simulator.java*. Na sljedećim je slikama prikazan *DogadjajiVjerojatnosti.java* razred te jezgreni dio *Simulator.java* razreda. Nakon dva razreda koji omogućuju simuliranje dana je implementacija java servleta koji koristi navedene java razrede da bi ostvario simulaciju u web aplikaciji, tj. *SimulatorServlet.java* servleta.

```

4 public void simuliraj(Utakmica utakmica) {
5     List<Igrac> iskljuceni = new ArrayList<>();
6     Map<Igrac, Integer[]> broj_vrijemeIskljucenja = new HashMap<>();
7     domaci = 0;
8     gost = 0;
9     double ukupnaVr = utakmica.getDomacin().getVrijednostKluba() + utakmica.getGost().getVrijednostKluba();
10    double domVr = utakmica.getDomacin().getVrijednostKluba() / ukupnaVr;
11    double gostVr = utakmica.getGost().getVrijednostKluba() / ukupnaVr;
12    int curr_time = 0;
13    double fact = domVr - gostVr;
14    int brojNapadaD = (int) Math.ceil(
15        (rand.nextGaussian() + fact * DogadjajiVjerojatnost.GAUSS_FACTOR) * DogadjajiVjerojatnost.GAUSS_FACTOR
16        + DogadjajiVjerojatnost.NUM_ATTACKS);
17    int brojNapadaG = (int) Math.ceil(
18        (rand.nextGaussian() - fact * DogadjajiVjerojatnost.GAUSS_FACTOR) * DogadjajiVjerojatnost.GAUSS_FACTOR
19        + DogadjajiVjerojatnost.NUM_ATTACKS);
20    int ukupanBrojNapada = brojNapadaD + brojNapadaG;
21    double dom_napad_prob = 1.0 * brojNapadaD / ukupanBrojNapada;
22    double gost_napad_prob = 1 - dom_napad_prob;
23    double gol_prob = 1.0 * DogadjajiVjerojatnost.GOALS_PER_MATCH / ukupanBrojNapada;
24    int totalEvents = ukupanBrojNapada + DogadjajiVjerojatnost.NUM_OF_STEALS
25        + DogadjajiVjerojatnost.SENT_OFF_PER_MATCH;
26    double steal_prob = 1.0 * DogadjajiVjerojatnost.NUM_OF_STEALS / (totalEvents + INC_DOG_STEAL);
27    double sent_off_prob = 1.0 * DogadjajiVjerojatnost.SENT_OFF_PER_MATCH / (totalEvents + INC_DOG_ISKLJ);
28    double shot_prob = 1 - steal_prob - sent_off_prob;
29    while (true) {
30        curr_time += rand.nextInt(MAX_SEK - MIN_SEK) + MIN_SEK;
31        Iterator<Igrac> it = iskljuceni.iterator();
32        while (it.hasNext()) {
33            Igrac sljedeci = it.next();
34            Integer[] vrijednosti = broj_vrijemeIskljucenja.get(sljedeci);
35            if (vrijednosti[0] >= BROJ_ISKLJ_ZA_TRAJNO) {
36                continue;
37            }
38            if (vrijednosti[1] + DVIJE_MIN <= curr_time) {
39                it.remove();
40            }
41        }
42        if (curr_time > KRAJ_UTAKMICE) break;
43        int event = whatEventHappens(shot_prob, steal_prob, sent_off_prob);
44        if (event == 0) {
45            sutPoGolu(utakmica, curr_time, dom_napad_prob, gost_napad_prob, gol_prob, iskljuceni);
46        } else if (event == 1) {
47            ukradena_lopta(utakmica, curr_time, iskljuceni);
48        } else {
49            iskljucenje(utakmica, curr_time, iskljuceni, broj_vrijemeIskljucenja);
50        }
51    }
52    if (domaci > gost) {
53        utakmica.getDomacin().setOstvarenibodovi(utakmica.getDomacin().getOstvarenibodovi() + BOD_POBJEDA);
54    } else if (gost > domaci) {
55        utakmica.getGost().setOstvarenibodovi(utakmica.getGost().getOstvarenibodovi() + BOD_POBJEDA);
56    } else {
57        utakmica.getDomacin().setOstvarenibodovi(utakmica.getDomacin().getOstvarenibodovi() + BOD_NERIJESEN);
58        utakmica.getGost().setOstvarenibodovi(utakmica.getGost().getOstvarenibodovi() + BOD_NERIJESEN);
59    }
60    utakmica.setZavrsena(true);
61}

```

Slika 7.5 Implementacija jezgre *Simuliraj.java* razreda

```

4 public class DogadjajiVjerojatnost {
5
6     // u sekundama
7     public static final int MATCH_DURATION = 3600;
8     public static final double IZRAVNO_ISKLJUCENJE = 0.02;
9     // Vjerojatnosti gol/šut
10    public static final double PENAL_PROB = 0.93;
11    public static final double U9_PROB = 0.85;
12    public static final double I9_PROB = 0.60;
13    // Vjerojatnost otkud šuta
14    public static final double SUT_I9_PROB = 0.25;
15    public static final double SUT_U9_PROB = 0.65;
16    public static final double SUT_7M_PROB = 0.1;
17    public static final double BLOK_PROB = 0.07;
18
19    public static final double NUM_ATTACKS = 52;
20
21    // Vjerojatnosti tko će dati gol ako je bio gol
22    public static final double KRILO_GOL_PROB = 0.1373/0.7344;
23    public static final double PIVOT_GOL_PROB = 0.1939/0.7344;
24    public static final double VANJSKI_GOL_PROB = 0.3005/0.7344;
25    public static final double PENAL_GOL_PROB = 0.1027/0.7344;
26
27    public static final double OBRANA_PROB = 0.3419;
28    public static final double OBRANA_U9_PROB = 4.0/9;
29    public static final double OBRANA_I9_PROB = 9.0/14;
30
31    public static final double GAUSS_FACTOR = 2;
32    public static final int GOALS_PER_MATCH = 55;
33    public static final int SENT_OFF_PER_MATCH = 10;
34    public static final int ATTACK_DURATION = 25;
35    public static final int NUM_OF_STEALS = 18;
36 }

```

Slika 7.6 Implementacija DogadjajiVjerojatnost.java razreda

```

4 public class SimulatorServlet extends HttpServlet {
5
6     @Override
7     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
8         if ((Long) DAOProvider.getDAO().izvediUpit("SELECT COUNT(*) FROM DogadjajVrijednost dv").get(0) == 0) {
9             req.setAttribute("poruka", "Nemoguće simulirati bez učitane početne konfiguracije!");
10        } else {
11            String[] utakmice = null;
12            String sel = req.getParameter(("selected"));
13            if (sel.isEmpty()) {
14                req.setAttribute("poruka", "Nije odabrana nijedna utakmica za simulaciju!");
15            } else {
16                utakmice = sel.split(",");
17            }
18            if (utakmice != null) {
19                Simulator sim = new Simulator();
20                for (String ut : utakmice) {
21                    Utakmica utakmica = DAOProvider.getDAO().dohvatiUtakmicu(Long.parseLong(ut));
22                    if (utakmica == null)
23                        continue;
24                    if (!utakmica.getDogadjaji().isEmpty()) {
25                        req.setAttribute("poruka", "Neke utakmica nije bilo moguće simulirati zbog toga što su već počele!");
26                        continue;
27                    }
28                    sim.simuliraj(utakmica);
29                    Klub domacin = DAOProvider.getDAO().dohvatiKlub(utakmica.getDomacin().getId());
30                    Klub gost = DAOProvider.getDAO().dohvatiKlub(utakmica.getGost().getId());
31                    domacin.setDaliGolova(domacin.getDaliGolova() + sim.getDomaci());
32                    domacin.setPrimiliGolova(domacin.getPrimiliGolova() + sim.getGost());
33                    gost.setDaliGolova(gost.getDaliGolova() + sim.getGost());
34                    gost.setPrimiliGolova(gost.getPrimiliGolova() + sim.getDomaci());
35                    utakmica.setRezultat(sim.getDomaci() + ":" + sim.getGost());
36                }
37            }
38            req.getRequestDispatcher("/servleti/admin/odabirzasim").forward(req, resp);
39        }
40    }

```

Slika 7.7 Implementacija SimulatorServlet.java servleta

Inicijalizacija rukometnih ekipa i lige

Main.java razred sadrži kod koji inicijalizira rukometnu ligu i njene pripadne ekipe. Raspored utakmica po kolima napravljen je prema Bergerovim tablicama. Isječak Main.java razreda prikazan je na sljedećoj slici koji sadrži implementaciju same inicijalizacije.

```

3   XLSread reader = new XLSread(args[1]);
4   List<Klub> klubovi = new ArrayList<>(reader.getKluboviMapa().values());
5   Liga liga = new Liga();
6   liga.setId(1);
7
8   EntityManager em = JPAEMProvider.getEntityManager();
9   DAOProvider.getDAO().dodajLigu(liga);
10  em.getTransaction().commit();
11
12  int j = 1;
13  for (Klub klub : klubovi) {
14      klub.setLiga(liga);
15      dodajKlub(emf, klub, j++);
16  }
17
18  dodajLigu(emf, new ArrayList<>(), new ArrayList<>());
19
20  String koloUtakmica =
21      "1    1-12    2-11    3-10    4-9    5-8    6-7    \n"
22      + "2    12-7    8-6    9-5    10-4    11-3    1-2    \n"
23      + "3    2-12    3-1    4-11    5-10    6-9    7-8    \n"
24      + "4    12-8    9-7    10-6    11-5    1-4    2-3    \n"
25      + "5    3-12    4-2    5-1    6-11    7-10    8-9    \n"
26      + "6    12-9    10-8    11-7    1-6    2-5    3-4    \n"
27      + "7    4-12    5-3    6-2    7-1    8-11    9-10    \n"
28      + "8    12-10    11-9    1-8    2-7    3-6    4-5    \n"
29      + "9    5-12    6-4    7-3    8-2    9-1    10-11    \n"
30      + "10   12-11    1-10    2-9    3-8    4-7    5-6    \n"
31      + "11   6-12    7-5    8-4    9-3    10-2    11-1";
32
33  Calendar datum = Calendar.getInstance();
34  datum.set(2016, 2, 1);
35
36  em.getTransaction().begin();
37  for (String koloRed : koloUtakmica.split("\n")) {
38      datum.add(Calendar.DAY_OF_MONTH, 1);
39      String raspored[] = koloRed.split("\\s+");
40
41      Kolo kolo = new Kolo();
42      kolo.setLiga(liga);
43      kolo.setBrojKola(Integer.parseInt(raspored[0]));
44
45      DAOProvider.getDAO().dodajKolo(kolo);
46
47      for (int i = 1; i < raspored.length; i++) {
48          Utakmica ut = new Utakmica();
49          String[] sud = raspored[i].split("-");
50          ut.setDatumOdigravanja(datum.getTime());
51          ut.setKolo(kolo);
52          ut.setDomacin(klubovi.get(Integer.parseInt(sud[0]) - 1));
53          ut.setGost(klubovi.get(Integer.parseInt(sud[1]) - 1));
54          DAOProvider.getDAO().dodajUtakmicu(ut);
55      }
56  }
57
58  em.getTransaction().commit();

```

Slika 7.8 Isječak Main.java razreda za inicijalizaciju ekipa i lige

7.4. Ispitivanje programskog rješenja

Prije nego krenemo na samo testiranje istaknimo zašto je testiranje programske podrške izričito važno kod izrade programske podrške:

- Da bi se istaknuli defekti i pogreške koje su učinjene tijekom razvojnih faza,
- Utvrđuje da će korisik biti zadovoljan pouzdanim radom aplikacije,
- Osigurava kvalitetu proizvoda; kvalitetan proizvod podiže povjerenje korisnika,
- Nužno je da bi se proizveo djelotvoran programski proizvod,
- Radi prevencije neočekivanog rada programske podrške te time sprječava moguće visoke troškove ponovnog prepravljanja iste,
- Da bi ostali kompetentni na tržištu.

Prije samih testova pogledajmo koje djelove ispitujemo te u koje se faze dijeli svaki od testova.

Ispitujemo sljedeće dijelove programskog rješenja:

1. Bazu podataka
2. Web aplikaciju preko svih korisnika
3. Nedozvoljen pristup dijelu web aplikacije od strane neovlaštene osobe

Faze ispitivanja programskog rješenja:

1. Opis ispitnog slučaja
2. Očekivani rezultat
3. Dobiveni rezultat

Test 1: Automatsko popunjavanje baze podataka tablicama koje su potrebne za izvođenje web aplikacije

Opis ispitnog slučaja:

Baza podataka je prazna, nema niti jednu tablicu koja se koristi u web aplikaciji.

Nakon pokretanja web aplikacije stvaraju se potrebne tablice. Provjeravamo direktno pomoću *ij* konzole (pogledati upute).

Očekivani rezultat:

Tablice će biti stvorene nakon što smo pokrenuli web aplikaciju.

Dobiveni rezultat:

Tablice su uspješno stvorene.

TABLE_SCHEM	TABLE_NAME	REMARKS
SYS	SYSALIASES	
SYS	SYSCHECKS	
SYS	SYSCOLPERMS	
SYS	SYSCOLUMNNS	
SYS	SYSCONGLOMERATES	
SYS	SYSCONSTRAINTS	
SYS	SYSDEPENDS	
SYS	SYSFILES	
SYS	SYSFOREIGNKEYS	
SYS	SYSKEYS	
SYS	SYSPERMS	
SYS	SYSROLES	
SYS	SYSROUTINEPERMS	
SYS	SYSSCHEMAS	
SYS	SYSSEQUENCES	
SYS	SYSSTATEMENTS	
SYS	SYSSTATISTICS	
SYS	SYSTABLEPERMS	
SYS	SYSTABLES	
SYS	SYSTRIGGERS	
SYS	SYSUSERS	
SYS	SYSVIEWS	
SYSIBM	SYSUMMY1	

Slika 7.9 Baza nema potrebne tablice za rad web aplikacije

TABLE_SCHEMA	TABLE_NAME	REMARKS
SYS	SYSALIASES	
SYS	SYSCHECKS	
SYS	SYSCOLPERMS	
SYS	SYSCOLUMNS	
SYS	SYSCONGLOMERATES	
SYS	SYSCONSTRAINTS	
SYS	SYSDEPENDS	
SYS	SYSFILES	
SYS	SYSFOREIGNKEYS	
SYS	SYSKEYS	
SYS	SYSPERMS	
SYS	SYSROLES	
SYS	SYSROUTINEPERMS	
SYS	SYSSCHEMAS	
SYS	SYSSEQUENCES	
SYS	SYSSTATEMENTS	
SYS	SYSSTATISTICS	
SYS	SYSTABLEPERMS	
SYS	SYSTABLES	
SYS	SYSTRIGGERS	
SYS	SYSUSERS	
SYS	SYSVIEWS	
SYSIBM	SYSDUMMY1	
ADMIN	DOGADJAJ	
ADMIN	DOGADJAJVRIJEDNOST	
ADMIN	IGRAC	
ADMIN	KLUB	
ADMIN	KOLO	
ADMIN	KORISNIK	
ADMIN	LIGA	
ADMIN	UTAKMICA	
ADMIN	VIRTEKIPA	
ADMIN	VIRTEKIPA_IMA_IGRACE	
ADMIN	VIRTEKIPA_POCETNIBODOVI	
ADMIN	VIRTLIGA	

35 rows selected
ij> █

Slika 7.10 Baza ima potrebne tablice za rad web aplikacije**Test 2: Izrada korisničkog računa za administratora****Opis ispitnog slučaja:**

Baza je tek napravljena, sve tablice su prazne. Pokretanjem Main.java razreda kao običnu java aplikaciju stvaramo administratora s podatcima koji su upisani u kodu.

Očekivani rezultat:

Nakon pokretanja web aplikacije možemo se prijaviti kao administrator.

Dobiveni rezultat:

Prijavljivanje kao administrator je uspješno.

```

1 ...
2 ...
3 ... String ime = "ime_administratora";
4 ... String prezime = "prezime_administratora";
5 ... String email = "email_administratora@xy.com";
6 ... String korime = "admin";
7 ... String sifra = "admin";
8 ... MessageDigest digest = null;
9 ... String hashsifra = null;
10 try {
11     digest = MessageDigest.getInstance("SHA-1");
12     digest.update(sifra.getBytes("utf8"));
13     byte[] digestBytes = digest.digest();
14     hashsifra = javax.xml.bind.DatatypeConverter.printHexBinary(digestBytes);
15 } catch (NoSuchAlgorithmException e) {
16     e.printStackTrace();
17 };
18 Administrator admin = new Administrator();
19 admin.setE_mail(email);
20 admin.setPassword(hashsifra);
21 admin.setPrezime(prezime);
22 admin.setUsername(korime);
23 admin.setIme(ime);
24 admin.setTip(TipKorisnika.ADMIN);
25
26 em.getTransaction().begin();
27 DAOProvider.getDAO().dodajKorisnika(admin);
28 em.getTransaction().commit();
29 ...

```

Slika 7.11 Isječak koda iz Main.java razreda koji stvara administratorski račun

The screenshot shows a web-based user login interface. At the top left, there are links for 'Registiraj se' and 'Početna stranica'. The main area is titled 'Prijava korisnika'. It features two input fields: one for 'Korisničko ime:' containing 'admin' and another for 'Šifra:' containing '****'. Below these fields are two buttons: a blue 'Prijavi se' button and a red 'Odustani' button.

Slika 7.12 Unos podataka administratora u formu za prijavu korisnika na web aplikaciju



Slike 7.13 Uspješna prijava na administratorski račun

Test 3: Stvaranje službene osobe

Opis ispitnog slučaja:

Nakon što smo stvorili administratora, administrator je zadužen za stvaranje računa službene osobe jer je to specijalan oblik korisničkog računa.

Očekivani rezultat:

Nakon unesenih podataka o službenoj osobi, uspješno se stvara novi račun.

Dobiveni rezultat:

Korisnički račun službene osobe uspješno je napravljen.

Dodavanje novog korisnika

Ime:
Službenik

Prezime:
Službenik

Email:
email_sluzbene_osobe@xy.com

Korisničko ime:
sluzbena_osoba

Sifra:

Vrsta korisnika:
Službena osoba ▾

Dodaj Korisnika

Slika 7.14 Unos podataka sa strane administratora radi stvaranja računa službene osobe

Dodavanje novog korisnika

Ime:
Unesite ime

Prezime:
Unesite prezime

Email:
Unesite email

Korisničko ime:
Unesite korisničko ime

Sifra:
Unesite šifru

Vrsta korisnika:
Službena osoba ▾

Dodaj Korisnika

Korisnik je uspješno dodan

Slika 7.15 Administrator je uspješno stvorio račun službene osobe

Test 4: Unos početne konfiguracije od strane službene osobe

Opis ispitnog slučaja:

Nakon što smo stvorili račun službene osobe, službena osoba unosi početnu konfiguraciju koja se koristi tijekom trajanja lige.

Očekivani rezultat:

Nakon ispravno unesenih podataka konfiguracija je učitana.

Dobiveni rezultat:

Kad su uneseni podatci ispravno uneseni, početna konfiguracija je učitana.

Službena osoba▼

➔ Odjava

Definicija početne konfiguracije

Proračun:
Unesite proračun

Zgoditak unutar 9 m:
Unesite vrijednost za zgoditak u 9 m

Zgoditak sa 7 m:
Unesite vrijednost za zgoditak sa 7 m

Zgoditak izvan 9 m:
Unesite vrijednost za zgoditak izvan 9 m

Obrana vratara:
Unesite vrijednost za obranu vratara

Blok šuta:
Unesite vrijednost za blok šuta

Slika 7.16 Ulaz u stranicu službene osobe u kojoj se učitava početna konfiguracija

Unesite vrijednost za promašaj

Promašaj šuta sa 7 m:
Unesite vrijednost za promašaj sa 7 m

Izgubljena lopta:
Unesite vrijednost za izgubljenu loptu

Primljen zgoditak:
Unesite vrijednost za primljen zgoditak

Primljen zgopoditak 7 m:
Unesite vrijednost za primljen zgopoditak sa 7 m

Isključenje 2 min:
Unesite vrijednost za isključenje na 2 minute

Trajno isključenje:
Unesite vrijednost za trajno isključenje

Izravno isključenje:
Unesite vrijednost za izravno isključenje

Definicija pocetne konfiguracije

Podatci nisu uneseni ispravno!

Slika 7.17 Ako podatke unesemo krivo definicija nije uspješna

Unesite vrijednost za promašaj

Promašaj šuta sa 7 m:
Unesite vrijednost za promašaj sa 7 m

Izgubljena lopta:
Unesite vrijednost za izgubljenu loptu

Primljen zgoditak:
Unesite vrijednost za primljen zgoditak

Primljen zgopoditak 7 m:
Unesite vrijednost za primljen zgopoditak sa 7 m

Isključenje 2 min:
Unesite vrijednost za isključenje na 2 minute

Trajno isključenje:
Unesite vrijednost za trajno isključenje

Izravno isključenje:
Unesite vrijednost za izravno isključenje

Definicija pocetne konfiguracije

Poèetna konfiguracija je uspješno definirana

Slika 7.18 Konfiguracija je uspješno učitana nakon ispravno unesenih podataka

Test 5: Unos početne konfiguracije od strane službene osobe nakon što je već unesena

Opis ispitnog slučaja:

Pomoću računa službene osobe ponovo pokušamo unijeti početnu konfiguraciju.

Očekivani rezultat:

Nismo u mogućnosti ponovo unijeti podatke.

Dobiveni rezultat:

Dobili smo ispravnu poruku.

The screenshot shows a dark-themed user interface. At the top, there is a navigation bar with the following items: "Rukometna liga" (highlighted in yellow), "Poredak rukometne lige", "Rezultati rukometne lige", "Virtualna liga ▾", "Sluzbena osoba ▾", and "Odjava". Below the navigation bar, the main content area has a light gray background. In the center, the text "Definicija početne konfiguracije" is displayed in bold black font. Below it, a red warning message "Početna konfiguracija već učitana!" is shown.

Slika 7.19 Ispis informacijske poruke nakon što službena osoba ponovo želi učitati početnu konfiguraciju

Test 6: Neovlašten pristup stranici

Opis ispitnog slučaja:

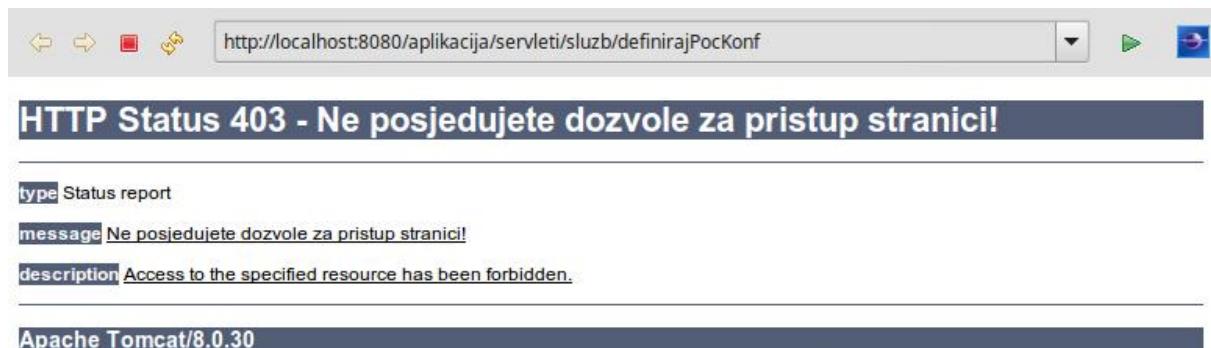
Recimo da netko zna putanju kojom se dolazi do unosa početne konfiguracije, ali nije ovlaštena osoba za tu radnju, web aplikacija će ga sprječavati, u toj operaciji.

Očekivani rezultat:

Pristup je blokiran jer je osoba neovlaštena za pristup.

Dobiveni rezultat:

Web aplikacija uspješno blokira osobu koja planira neovlašteno pristupiti sadržaju.



Slika 7.20 Nedozvoljen pristup stranici pravilno je obrađen

Test 7: Stvaranje korisničkog računa tehničke komisije

Opis ispitnog slučaja:

Administrator stvara korisnički račun tehničke komisije.

Očekivani rezultat:

Nakon unešenih podataka uspješno se stvara račun.

Dobiveni rezultat:

Račun se uspješno stvorio.

Dodavanje novog korisnika

Ime:
tehniciar

Prezime:
tehniciar

Email:
email_tehnicke_komisije@xy.com

Korisničko ime:
tehnicka_komisija

Sifra:

Vrsta korisnika:
Tehnička komisija

Dodaj Korisnika

Slika 7.21 Administrator unosi podatke za stvaranje računa tehničke komisije

Dodavanje novog korisnika

Ime:
Unesite ime

Prezime:
Unesite prezime

Email:
Unesite email

Korisničko ime:
Unesite korisničko ime

Sifra:
Unesite šifru

Vrsta korisnika:
Službena osoba

Dodaj Korisnika

Korisnik je uspješno dodan

Slika 7.22 Račun tehničke komisije uspješno je napravljen

Test 8: Simulacija i pregled rezultata

Opis ispitnog slučaja:

Administrator simulira određene utakmice. Nakon toga odlazi se na poredak lige da bi se vidjelo ima li kakvih promjena.

Očekivani rezultat:

Nakon označenih utakmica i pokretanja simulacije simulacija je uspješno izvršena. Na poretku lige uočavaju se promjene.

Dobiveni rezultat:

Simulacija je uspjela. Poredak lige je uspješno ažuriran.

1.Kolo			
Datum odigravanja	Domacin	Gost	Simuliraj
2016-03-02 13:36:39.605	RK Zagreb	KF Kolding	<input checked="" type="checkbox"/>
2016-03-02 13:36:39.605	RK Celje	Chekhovskiye Medvedi	<input checked="" type="checkbox"/>
2016-03-02 13:36:39.605	KS Vive Tauron Kielce	THW Kiel	<input checked="" type="checkbox"/>
2016-03-02 13:36:39.605	FC Barcelona	RK Vojvodina	<input checked="" type="checkbox"/>
2016-03-02 13:36:39.605	RK Metalurg Skopje	SG Flensburg-Handewitt	<input checked="" type="checkbox"/>
2016-03-02 13:36:39.605	Veszprem	Paris SG	<input type="checkbox"/>
			<input type="checkbox"/> Označi kolo

Slika 7.23 Administrator označava utakmice koje želi simulirati



Slika 7.24 Administrator pritiskom na gumb „Simulacija“ pokreće simulaciju odabranih utakmica

1.Kolo

Datum odigravanja	Domacin	Gost	Simuliraj
2016-03-02 13:36:39.605	RK Zagreb	KF Kolding	36:20
2016-03-02 13:36:39.605	RK Celje	Chekhovskiye Medvedi	25:31
2016-03-02 13:36:39.605	KS Vive Tauron Kielce	THW Kiel	29:32
2016-03-02 13:36:39.605	FC Barcelona	RK Vojvodina	29:25
2016-03-02 13:36:39.605	RK Metalurg Skopje	SG Flensburg-Handewitt	24:26
2016-03-02 13:36:39.605	Veszprem	Paris SG	<input type="checkbox"/>
			<input type="checkbox"/> Označi kolo

Slika 7.25 Nakon simulacije odmah vidimo rezultate simuliranih utakmica te znamo da je simulacija uspješno izvršena

Poredak

Ogranak	Ime kluba	Ostvareni bodovi	Golovi	Primljeni	Razlika
Ogranak1	RK Zagreb	3	36	20	16
Ogranak11	Chekhovskiye Medvedi	3	31	25	6
Ogranak4	FC Barcelona	3	29	25	4
Ogranak10	THW Kiel	3	32	29	3
Ogranak8	SG Flensburg-Handewitt	3	26	24	2
Ogranak6	Veszprem	0	0	0	0
Ogranak7	Paris SG	0	0	0	0
Ogranak5	RK Metalurg Skopje	0	24	26	-2
Ogranak3	KS Vive Tauron Kielce	0	29	32	-3
Ogranak9	RK Vojvodina	0	25	29	-4
Ogranak2	RK Celje	0	25	31	-6
Ogranak12	KF Kolding	0	20	36	-16

Slika 7.26 Vidimo da je simulacija uspješna jer se ažurirao poredak i samo ekipe koje nisu igrale utakmicu nemaju bodove

Test 9: Registracija novog korisnika te odabir virtualne ekipe

Opis ispitnog slučaja:

Simuliramo ponašanje običnog korisnika koji bi koristio ovu web aplikaciju. Pristupamo web aplikaciji preko web browsera te radimo novi korisnički račun i biramo sastav za virtualnu ligu. Nakon uspješne registracije i odabira virtualne ekipe provjeravamo na službenim tablicama postojimo li u sustavu.

Očekivani rezultat:

Stvaranje novog računa bit će uspješno, također i odabir sastava za virtualnu ligu. Nakon pregleda službenih tablica uočiti ćemo svoj račun.

Dobiveni rezultat:

Stvaranje računa je uspješno, također i odabir sastava momčadi. Vidimo u službenim tablicama svoj račun, tj. svoju ekipu.

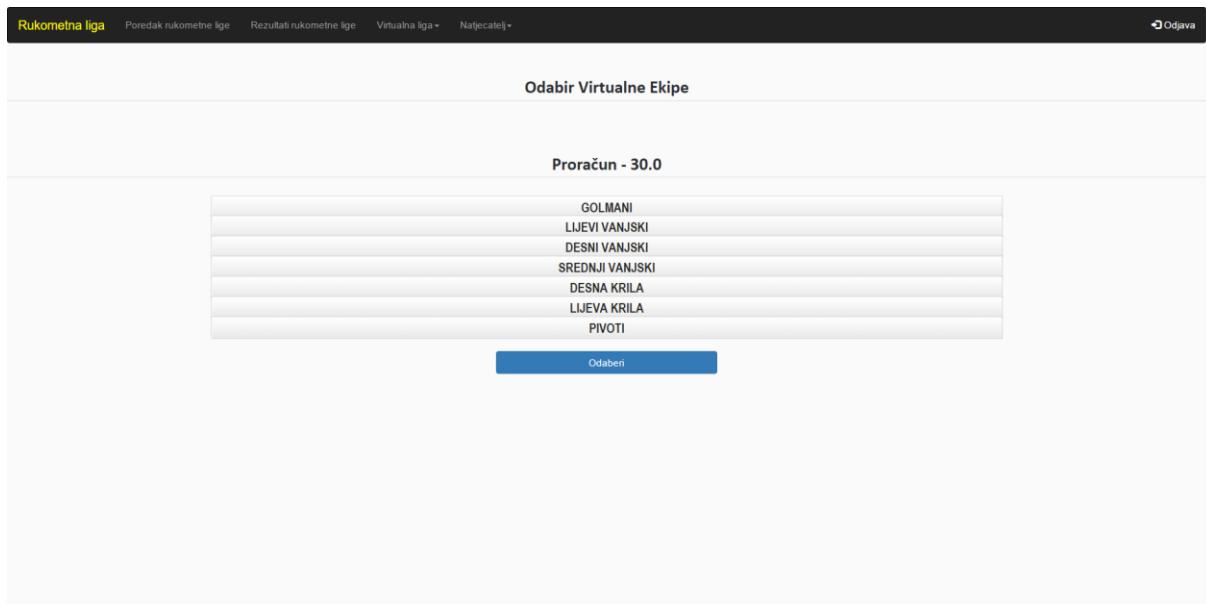
Početna stranica

Registracija korisnika

Ime:	Ivo
Prezime:	Ivić
Email:	ii21312@fer.hr
Korisničko ime:	ivo
Sifra:	...
Naziv virtualne ekipe:	Gladijatori
Država:	Hrvatska
Podržani klub:	RK Celje

Registiraj se Odustani

Slika 7.27 Korisnik se registrira unosom svojih osobnih podataka



Slika 7.28 Nakon registracije korisnik bira sastav svoje virtualne ekipe

Moja Ekipa				
Ime i prezime	Vrijednost	Broj bodova	Pozicija	Klub
Ivan Stevanović	1.0	0.0	Golman	RK Zagreb
Žiga Mlakar	4.0	0.0	Lijevi vanjski	RK Celje
Marko Buvinić	4.0	0.0	Desni vanjski	RK Metalurg Skopje
Josip Valčić	3.0	0.0	Srednji vanjski	RK Zagreb
Victor Tomas	3.0	0.0	Desno krilo	FC Barcelona
Antonije Kovačević	3.0	0.0	Ljevo krilo	RK Zagreb
Zoran Nikolić	2.0	0.0	Pivot	RK Vojvodina
UKUPNO		90.0		

Slika 7.29 Nakon uspješne registracije i odabira sastava sustav omogućuje pregled ekipe

Poredak virtualne lige				
Ime ekipe	Natjecatelj	Država natjecatelja	Bodovi	Podržani klub
Nosati	bobrock	SAD	104.0	FC Barcelona
KrademoSiromasnim DajemoBogatim	superhik	Not Sherwood Forest	100.0	Veszprem
Hipohondri	jeremija	Srbija	98.0	RK Vojvodina
Cvjecari	broj1	SAD	86.0	Paris SG
Kleptomani	siroliker	Ujedinjeno Kraljevstvo Velike Britanije i Sjeverne Irske	84.0	FC Barcelona
Zbunjivci	alanford	SAD	54.0	RK Zagreb
Inovatori	grunf	Njemačka	34.0	THW Kiel
Gladijatori	ivo	Hrvatska	0.0	RK Celje

Slika 7.30 Novostvoreni račun je sada dio ostalih službenih tablica

Test 10: Proglašavanje MVP-a pomoću računa tehničke komisije

Opis ispitnog slučaja:

Tehnička komisija se prijavi sa svojim korisničkim računom te odlazi na rezultate odigranih utakmica i proglašava najboljeg igrača utakmice.

Očekivani rezultat:

Bit će moguće odabrati najboljeg igrača te će odabir biti izvršen.

Dobiveni rezultat:

Najbolji igrač je uspješno odabran te je odabir izvršen.

RK Zagreb - KF Kolding					
REZULTAT					
36:20					
SAŽETAK			IGRAČI		
Vrijeme	Ime igrača	Vrsta događaja	Ime igrača	Pozicija	Ocjena
0:22	Filip Ivić	Obrana	Ivan Stevanović	Golman	10
0:45	Kasper Hvadt	Obrana	Filip Ivić	Golman	10
1:12	Sebastian Augstnussen	Gol izvan 9m	Josip Valčić	Srednji vanjski	10
1:36	Kasper Hvadt	Obrana	Luka Stepančić	Desni vanjski	10
2:28	Ivan Stevanović	Obrana	Luka Šibetić	Desni vanjski	10

Slika 7.31 Tehnička komisija ulazi u neku od odigranih utakmica te bira najboljeg igrača

RK Zagreb - KF Kolding					
REZULTAT			MVP		
36:20			Ivan Stevanović		
SAŽETAK			IGRAČI		
50:31	Ivan Stevanović	Obrana	Ime igrača	Pozicija	Ocjena
50:51	Kasper Hvadt	Obrana	Ivan Stevanović	Golman	10
51:13	Josip Valčić	Gol unutar 9m	Filip Ivić	Golman	10
51:33	Josip Valčić	Gol unutar 9m	Josip Valčić	Srednji vanjski	10
52:01	Luka Raković	Gol unutar 9m	Luka Stepančić	Desni vanjski	10
53:11	Cyril Viudes	Gol sa 7m	Luka Šibetić	Desni vanjski	10

Slika 7.32 Biranje najboljeg igrača uspješno je izvršeno te se on vidi pored rezultata utakmice

7.5. Upute za instalaciju

Za ispravni rad aplikacije potrebno je prethodno instalirati i podesiti Apache Derby (**Dodatak E**).

Za rad s bazama podataka potrebno je pokrenuti bazu podataka sa skriptom *server-start.bat*, te nakon toga stvoriti bazu podataka koju će koristiti aplikacija. Za to je potrebno pokrenuti skriptu *ij.bat*. Nakon toga se nalazite u *ij* konzoli. Sve naredbe upisati u jednom redu u *ij* konzoli.

Upisati sljedeću naredbu:

connect

'*jdbc:derby://localhost:1527/credentialsDB;user=sa;password=sapwd22;create=true*';

Naredbom se stvara korisnik vlasnik baze podataka, tako zvani superuser, te baza podataka *credentialsDB* koja služi kao inicijalna postavka za sve sljedeće kreirane baze podataka.

Upišite sljedeću naredbu da bi se odspojili od novo stvorene baze *credentialsDB*:

disconnect;

Sada krećemo na izradu baze podataka za našu web aplikaciju. Unesite sljedeću naredbu:

connect

'*jdbc:derby://localhost:1527/rukometBaza;user=sa;password=sapwd22;create=true*';

Stvorili smo novu bazu podataka *rukometBaza*.

Sljedeće što želimo je stvoriti novog korisnika koji će biti administrator novonastale baze te će on imati sve ovlasti nad tom bazom.

Unesite sljedeću naredbu:

CALL SYSCS_UTIL.SYSCS_CREATE_USER('admin', 'rukometAdmin');

Stvorili smo administratora „admin“ s lozinkom „rukometAdmin“. Dajmo mu sada i ovlasti nad bazom *rukometBaza*.

CALL

SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.database.fullAccessUsers', 'admin');

Gotovi smo, sada se odspojite od baze rukometBaza jer ste prijavljeni kao superuser „sa“. Unesite naredbu:

disconnect;

Spojite se s računom administratora za bazu rukometBaza. Unesite naredbu:

connect

'jdbc:derby://localhost:1527/rukometBaza;user=admin;password=rukometAdmin';

Sada možete unositi upite kao „SHOW TABLES;“, SELECT X FROM Y; i slično da biste provjerili stanje tablica u bazi podataka. Prije nego što pokrenete web aplikaciju, tablice aplikacije neće postojati u bazi podataka.

Nakon uspješne instalacije Apache Derby-ja potrebno je skinuti i instalirati Tomcat poslužitelj (**Dodatak F**) na kojem će se aplikacija izvršavati.

Aplikaciji se može pristupiti pomoću linka na listi aplikacija ili izravno na <http://localhost:8080/aplikacija/>. Drugi korisnici mogu pristupiti aplikaciji tako da zamijene localhost s IP adresom računala na kojem je upaljen server, ili nazivom domene.

Za učitavanje i čišćenje podataka u bazi služi datoteka *RadSBazom.jar*. Potrebno ju je pokrenuti sa 1 ili 2 argumenta, ovisno o tome što želite napraviti.

Ako treba samo očistiti bazu, tj. pobrisati sve podatke iz nje, preda se samo kao prvi argument riječ „brisi“, npr: *java –jar RadSBazom.jar brisi* ;

Ako treba učitati podatke iz .xls datoteke u bazu, kao prvi argument se preda riječ „ucitaj“, a kao drugi putanja do .xls datoteke u kojoj su definirani potrebni podaci. Primjer te datoteke je priložen uz *RadSBazom.jar*. Jedan primjer poziva za drugi slučaj može biti: *java –jar RadSBazom.jar ucitaj podatci.xls* ;

7.6. Korisničke upute

Za korištenje razvijene web aplikacije od strane krajnjeg korisnika potrebno je da korisnik ima pristup Internetu te neki od web preglednika po svom izboru, a sama aplikacija je napravljena tako da bude intuitivna za korištenje prosječnom korisniku Interneta.

Budući da razlikujemo nekoliko vrsta korisnika, odlučili smo korisničke upute podijeliti u nekoliko dijelova.

Napomena: korisnike koji imaju uloge tehničke komisije te službene osobe može stvoriti samo administrator, ne običan korisnik.

Neregistrirani korisnik

Na početnoj stranici neregistriranom korisniku biti će ponuđene sljedeće mogućnosti:

- Poredak rukometne lige
- Rezultati rukometne lige
- Registracija
- Prijava

Poredak rukometne lige je automatski prikazan, te nudi prikaz ostvarenih bodova, postignutih i primljenih golova, te njihovu razliku.

Poredak					
Ogranak	Ime kluba	Ostvareni bodovi	Golovi	Primjeni	Razlika
Ogranak7	Paris SG	16	196	152	44
Ogranak6	Veszprem	12	204	189	15
Ogranak9	RK Vojvodina	12	185	185	0
Ogranak12	KF Kolding	12	163	187	-24
Ogranak11	Chekhovskiye Medvedi	11	199	167	32
Ogranak3	KS Vive Tauron Kielce	10	192	189	3
Ogranak5	RK Metalurg Skopje	10	176	188	-12
Ogranak1	RK Zagreb	9	181	189	-8
Ogranak8	SG Flensburg-Handewitt	9	182	193	-11
Ogranak2	RK Celje	9	178	191	-13
Ogranak10	THW Kiel	7	180	188	-8
Ogranak4	FC Barcelona	6	176	194	-18

Slika 7.33 Poredak rukometne lige

Ako korisnik odabere pregled rezultata rukometne lige klikom na „Rezultati rukometne lige“, prikazuju mu se pregledi odigranih utakmica po kolima.

1.Kolo			
Datum odigravanja	Domacin	Gost	Rezultat
2016-03-02 10:31:31.769	RK Zagreb	KF Kolding	30:20
2016-03-02 10:31:31.769	RK Celje	Chekhovskiye Medvedi	25:26
2016-03-02 10:31:31.769	KS Vive Tauron Kielce	THW Kiel	25:23
2016-03-02 10:31:31.769	FC Barcelona	RK Vojvodina	32:26
2016-03-02 10:31:31.769	RK Metalurg Skopje	SG Flensburg-Handewitt	42:28
2016-03-02 10:31:31.769	Veszprem	Paris SG	30:27

2.Kolo			
Datum odigravanja	Domacin	Gost	Rezultat
2016-03-03 10:31:31.769	KF Kolding	Paris SG	
2016-03-03 10:31:31.769	SG Flensburg-Handewitt	Veszprem	
2016-03-03 10:31:31.769	RK Vojvodina	RK Metalurg Skopje	
2016-03-03 10:31:31.769	THW Kiel	FC Barcelona	
2016-03-03 10:31:31.769	Chekhovskiye Medvedi	KS Vive Tauron Kielce	
2016-03-03 10:31:31.769	RK Zagreb	RK Celje	

Slika 7.34 Pregled odigranih utakmica po kolima

Klikom na pojedinačnu utakmicu dobiva se uvid u sve događaje koji su se odvijali na utakmici, kao i u ocjene pojedinih igrača.

REZULTAT

30:20

SAŽETAK

Vrijeme	Ime igrača	Vrsta događaja
0:27	Bojan Spellerberg	Gol unutar 9m
1:37	Lars Jørgensen	Gol unutar 9m
2:05	Nikolaj Enderlein	Gol unutar 9m
3:38	Simon Jensen	Gol unutar 9m
4:04	Teo Čorić	Gol izvan 9m
4:24	Kasper Irmung Andersen	Gol unutar 9m
4:47	Kasper Hvidt	Obrama
5:13	Luka Šebetić	Gol unutar 9m
5:39	Kasper Irmung Andersen	Gol unutar 9m
6:05	Kasper Hvidt	Obrama
7:02	Marcus Cleverly	Obrama
7:27	Filip Ivčić	Obrama
7:50	Marcus Cleverly	Obrama
8:40	Kasper Hvidt	Obrama

IGRAČI

Ime igrača	Pozicija	Ocjena	Ime igrača	Pozicija	Ocjena
Ivan Stevanović	Golman	3.4	Kasper Hvidt	Golman	8.2
Filip Ivčić	Golman	3.4	Torsten Laen	Pivot	5.2
Josip Valičić	Srednji vanjski	8.2	Cyril Vludes	Pivot	2.8
Luka Stepančić	Desni vanjski	7	Lukas Karlsson	Desni vanjski	2.2
Luka Šebetić	Desni vanjski	8.2	Lars Jørgensen	Desni vanjski	8.2
Sandro Obranović	Srednji vanjski	10	Lasse Andersson	Srednji vanjski	5.2
Mateo Lukatčić	Srednji vanjski	2.2	Bojan Spellerberg	Srednji vanjski	8.2
Damagoj Pavlović	Ljevi vanjski	2.2	Kasper Irmung Andersen	Srednji vanjski	5.2
Tonči Valičić	Ljevi vanjski	7	Sebastian Augustinusen	Ljevi vanjski	5.2
Stipe Mandalinić	Ljevi vanjski	1	Nikolaј Enderlein	Ljevi vanjski	8.2
Zlatko Horvat	Desno krilo		Konstantin Igropulo	Desno krilo	2.2
Ilija Brozović	Pivot	8.2	Marcus Cleverly	Golman	3.4
Luka Raković	Desno krilo		Jakob Stork	Desno krilo	2.8
Lovo Sprem	Ljevo krilo	2.2	Magnus Landin-Jacobsen	Ljevo krilo	5.2

Slika 7.35 Statistika pojedinačne utakmice

Ukoliko to želi, neregistrirani korisnik može se registrirati u sustav i postati natjecatelj. Da bi to učinio, potrebno je kliknuti na "Registriraj se" te unijeti sve potrebne podatke u zadana polja. Sva polja moraju se popuniti, a željeno korisničko ime ne smije biti već postojeće u bazi podataka.

The screenshot shows a registration form titled "Registracija korisnika". The form consists of several input fields with placeholder text: "Unesite ime" (Enter first name), "Unesite prezime" (Enter last name), "Unesite email" (Enter email), "Unesite korisničko ime" (Enter username), "Unesite lozbu" (Enter password), "Unesite naziv virtualne ekipe" (Enter virtual team name), "Unesite državu" (Enter country), and "Podržani klub" (Supported club) with a dropdown menu showing "RK Zagreb". At the bottom are two buttons: a green "Registriraj se" (Register) button and a red "Odustani" (Cancel) button.

Slika 7.36 Registracija neregistriranog korisnika

Natjecatelj

Natjecatelju se proširuju mogućnosti koje ima na početnoj stranici, te on dobiva uvid u stanje virtualne lige, i to u sljedeće podatke:

- Poredak virtualne lige
- Poredak po podržanim klubovima
- Poredak po državama
- Ukupan popis sudionika virtualne lige

Ime ekipe	Natjecatelj	Država natjecatelja	Bodovi	Podržani klub
Hipohondri	jeremija	Srbija	1448.0	RK Vojvodina
Nosati	bobrock	SAD	1387.0	FC Barcelona
Kleptomani	siroliver	Ujedinstveno Kraljevstvo Velike Britanije i Sjeverne Irske	1203.0	FC Barcelona
KrademoSiromasnil DajemoBogatin	superhik	Not Sherwood Forest	1195.0	Veszprem
Orijearli	broj1	SAD	1069.0	Paris SG
Zbuniljci	alanford	SAD	981.0	RK Zagreb
Inovatori	grunf	Njemačka	844.0	THW Kiel

Slika 7.37 Poredak virtualne lige

Ako se odabere poredak virtualne lige klikom na padajući izbornik „Virtualna liga“, te zatim na ponuđenu opciju „Poredak virtualne lige“, natjecatelj dobiva uvid u bodovni poredak svih registriranih natjecatelja za trenutno aktivnu virtualnu ligu. Ako bi radije htio vidjeti poredak po državama ili po podržanim klubovima, takav pogled se dobiva odabirom pripadajuće opcije u padajućem izborniku virtualne lige.

Ime ekipe	Natjecatelj	Bodovi
Nosati	bobrock	1387.0
Kleptomani	siroliver	1203.0

Slika 7.38 Poredak virtualne lige po podržanim klubovima

Prije sudjelovanja u virtualnoj ligi, svaki korisnik mora odabrati svojih 7 igrača, pazeći da ne prekorači zadani budžet. Za svaku poziciju može se odabrati maksimalno jedan igrač, neovisno o tome je li taj igrač već izabran od strane nekog drugog natjecatelja. Nije dopušten odabir igrača na pogrešnu poziciju (npr. nije moguć odabir igrača koji igra na poziciji lijevog vanjskog na poziciju golmana).

Natjecatelj vlastitu virtualnu ekipu može složiti samo jednom za svaku pojedinu virtualnu ligu, i to klikom na padajući izbornik „Natjecatelj“ te zatim odabirom opcije „Odabir virtualne ekipe“.

Odabir Virtualne Erike				
Proračun - 30.0				
	GOLMANI	LJEVI VANJSKI	DESNI VANJSKI	
Klub	Ime	Vrijednost	Bodovi	Odaberi
RK Zagreb	Luka Stepančić	4.0	0.0	<input type="radio"/>
RK Zagreb	Luka Šibetić	3.0	0.0	<input type="radio"/>
RK Celje	Povilas Babarskas	4.0	0.0	<input type="radio"/>
RK Celje	David Razgor	3.0	0.0	<input type="radio"/>
KS Vive Tauron Kielce	Michał Jurecki	4.0	0.0	<input type="radio"/>
KS Vive Tauron Kielce	Grzegorz Tkaczyk	3.0	0.0	<input type="radio"/>
FC Barcelona	Marko Kopjar	4.0	0.0	<input type="radio"/>
FC Barcelona	Eduardo Gurbido	3.0	0.0	<input type="radio"/>
RK Metalurg Skopje	Marko Buvinić	4.0	0.0	<input type="radio"/>
RK Metalurg Skopje	Janko Božović	3.0	0.0	<input type="radio"/>
Veszprem	Laszlo Nagy	4.0	0.0	<input type="radio"/>
Paris SG	Xavier Barachet	4.0	0.0	<input type="radio"/>
Paris SG	Sergiy Onufrienko	3.0	0.0	<input type="radio"/>
SG Flensburg-Handewitt	Holger Glandorf	4.0	0.0	<input type="radio"/>
SG Flensburg-Handewitt	Thomas Mogensen	3.0	0.0	<input type="radio"/>
RK Vojvodina	Nikola Arsenić	4.0	0.0	<input type="radio"/>

Slika 7.39 Odabir virtualne ekipe

Kako se utakmice budu odigravale, tako će svaki natjecatelj odabirom opcije „Moja ekipa“ u padajućem izborniku „Natjecatelj“ imati uvid u bodove koje je ostvario pojedini igrač izabran u ekipu.

Ime igrača	Broj godina	Vrijednost igrača	Pozicija igrača	Ukupni bodovi
Ivan Gajic	36	2.0	Golman	153.0
Jim Gottfridsson	23	3.0	Ljevi vanjski	304.0
Marko Buvnić	19	4.0	Desni vanjski	226.0
Sergei Shelemenko	32	2.0	Srednji vanjski	216.0
Blaž Blagotinšek	21	2.0	Pivot	147.0
Eduardo Koksharov	40	2.0	Ljevo krilo	74.0
Torsten Jansen	39	3.0	Desno krilo	75.0
UKUPNO				1195.0

Slika 7.40 Bodovi koje je ostvario pojedini igrač

Službena osoba

Službena osoba uz uvid u rezultate i poretke stvarne i virtualne lige ima mogućnosti definicije početne konfiguracije te postavljanja cijene pojedinih igrača. Klikom na jednu od tih ponuđenih opcija u padajućem izborniku „Sluzbena osoba“ otvara se nova stranica u kojoj je moguće namjestiti sve potrebne parametre.

Bitno je napomenuti da se cijene pojedinih igrača mogu namještati samo prije početka lige.

Definicija početne konfiguracije

Proračun:	<input type="text" value="Unesite proračun"/>
Zgoditak unutar 9 m:	<input type="text" value="Unesite vrijednost za zgoditak u 9 m"/>
Zgoditak sa 7 m:	<input type="text" value="Unesite vrijednost za zgoditak sa 7 m"/>
Zgoditak izvan 9 m:	<input type="text" value="Unesite vrijednost za zgoditak izvan 9 m"/>
Obrana vrataraz:	<input type="text" value="Unesite vrijednost za obranu vratara"/>
Blok šuta:	<input type="text" value="Unesite vrijednost za blok šuta"/>
Ukradena lopta:	<input type="text" value="Unesite vrijednost za ukradenu loplu"/>
Obrana vratara 7 m:	<input type="text" value="Unesite vrijednost za obranu sa 7 m"/>
Najbolji igrač utakmice:	<input type="text" value="Unesite vrijednost za najboljeg igrača"/>
Promašaj šuta:	<input type="text" value="Unesite vrijednost za promašaj"/>
Promašaj šuta sa 7 m:	<input type="text" value="Unesite vrijednost za promašaj sa 7 m"/>
Izgubljena lopta:	<input type="text" value="Unesite vrijednost za izgubljenu loptu"/>
Primljen zgoditak:	<input type="text" value="Unesite vrijednost za primljen zgoditak"/>
Primljen zgoditak 7 m:	<input type="text" value="Unesite vrijednost za primljen zgoditak sa 7 m"/>
Isključenje 2 min:	<input type="text" value="Unesite vrijednost za isključenje na 2 minute"/>
Trajno isključenje:	<input type="text" value="Unesite vrijednost za trajno isključenje"/>
Izravno isključenje:	<input type="text" value="Unesite vrijednost za izravno isključenje"/>

Definicija početne konfiguracije

Slika 7.41 Definicija početne konfiguracije

Tehnička komisija

Korisnik koji je član tehničke komisije uz uvid u rezultate i poretke stvarne i virtualne lige ima mogućnost odabira najboljeg igrača (MVP – *Most Valuable Player*) za odigrane utakmice odabirom odgovarajuće opcije „Proglašavanje MVP-a“, te zatim odabirom konkretne utakmice te naponslijetku odabirom igrača.

Za svaku utakmicu može se odabrati samo 1 najbolji igrač.

REZULTAT

30:23

SAŽETAK

Vrijeme	Ime igrača	Vrsta dogadaja
1:29	Marcus Cleverly	Obrana
1:53	Konstantin Igropulo	Gol sa 7m
2:13	Torsten Laen	Isključenje 2min
3:06	Kasper Irmng Andersen	Gol unutar 9m
3:35	Luka Stepančić	Gol izvan 9m
5:12	Tonči Valičić	Isključenje 2min
5:33	Ilija Brozović	Gol unutar 9m
5:59	Mateo Lukačec	Gol unutar 9m
6:24	Ivan Stevanović	Obrana
6:44	Kasper Irmng Andersen	Gol izvan 9m
7:12	Lars Jorgensen	Gol unutar 9m
8:45	Filip Ivic	Obrana

IGRAČI

Ime igrača	Pozicija	Ocjena	Ime igrača	Pozicija	Ocjena		
Ivan Stevanović	Golman	3.08	Proglaši MVP	Kasper Hvidt	Golman	2.38	Proglaši MVP
Filip Ivic	Golman	2.38	Proglaši MVP	Torsten Laen	Pivot	4.46	Proglaši MVP
Josip Valičić	Srednji vanjski	10	Proglaši MVP	Cyril Vuades	Pivot	4.46	Proglaši MVP
Luka Stepančić	Desni vanjski	1	Proglaši MVP	Lukas Karlsson	Desni vanjski	1	Proglaši MVP
Luka Šibetić	Desni vanjski	1	Proglaši MVP	Lars Jorgensen	Desni vanjski	7.92	Proglaši MVP
Sandro Obranović	Srednji vanjski	6.54	Proglaši MVP	Lasse Andersson	Srednji vanjski	1.69	Proglaši MVP
Mateo Lukačec	Srednji vanjski	7.92	Proglaši MVP	Bojan Spellerberg	Srednji vanjski	1.69	Proglaši MVP
Domagoj Pavlović	Ljevi vanjski	6.54	Proglaši MVP	Kasper Irmng Andersen	Srednji vanjski	7.92	Proglaši MVP
Tonči Valičić	Ljevi vanjski	1	Proglaši MVP	Sebastian Augustinussen	Ljevi vanjski	10	Proglaši MVP
Stipe Mandalinić	Ljevi vanjski	7.92	Proglaši MVP	Nikolaj Enderleit	Ljevi vanjski	1	Proglaši MVP
Zlatko Horvat	Desno krilo	1	Proglaši MVP	Konstantin Igropulo	Desno krilo	6.54	Proglaši MVP
Ilija Brozović	Pivot	7.92	Proglaši MVP	Marcus Cleverly	Golman	7.92	Proglaši MVP

Slika 7.42 Proglašavanje MVP-a

Administrator

Administrator, kao i službena osoba te predstavnik tehničke komisije, ima uvid u stanje stvarne te virtualne lige, a osim toga ima još određene dodatne mogućnosti:

- Učitavanje početne konfiguracije prethodno definirane od strane službene osobe,
- Simulacija pojedinačne utakmice,
- Simulacija cijelog kola/lige,
- Dodavanje korisnika (tehničke komisije i službene osobe),
- Restart lige (stvarne i virtualne).

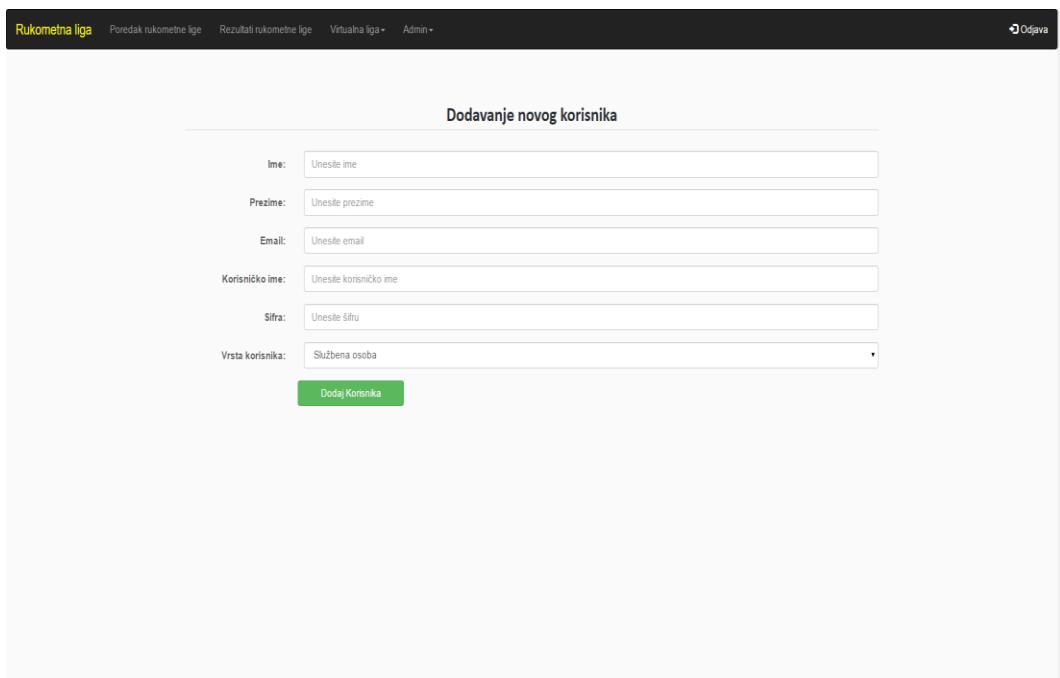
Odabirom opcije „Simulacija“ u padajućem izborniku „Admin“ dolazi se na ovakvu stranicu gdje administrator može odabrati točno koje utakmice želi simulirati. Također, administrator ima mogućnost označavanja i simuliranja cijelog kola.

3.Kolo			
Datum odigravanja	Domacin	Gost	Simuliraj
2016-03-04 10:31:31.769	RK Celje	KF Kolding	<input type="checkbox"/>
2016-03-04 10:31:31.769	KS Vive Tauron Kielce	RK Zagreb	<input type="checkbox"/>
2016-03-04 10:31:31.769	FC Barcelona	Chekhovskije Medvedi	<input type="checkbox"/>
2016-03-04 10:31:31.769	RK Metalurg Skopje	THW Kiel	<input type="checkbox"/>
2016-03-04 10:31:31.769	Veszprem	RK Vojvodina	<input type="checkbox"/>
2016-03-04 10:31:31.769	Paris SG	SG Flensburg-Handewitt	<input type="checkbox"/>
			<input type="checkbox"/> Označi kolo

4.Kolo			
Datum odigravanja	Domacin	Gost	Simuliraj
2016-03-05 10:31:31.769	KF Kolding	SG Flensburg-Handewitt	<input type="checkbox"/>
2016-03-05 10:31:31.769	RK Vojvodina	Paris SG	<input type="checkbox"/>
2016-03-05 10:31:31.769	THW Kiel	Veszprem	<input type="checkbox"/>
2016-03-05 10:31:31.769	Chekhovskije Medvedi	RK Metalurg Skopje	<input type="checkbox"/>
2016-03-05 10:31:31.769	RK Zagreb	FC Barcelona	<input type="checkbox"/>
2016-03-05 10:31:31.769	RK Celje	KS Vive Tauron Kielce	<input type="checkbox"/>
			<input type="checkbox"/> Označi kolo

Slika 7.43 Raspored utakmica

Opcija „Dodavanje korisnika“ u padajućem izboniku „Admin“ koristi se za dodavanje službene osobe i/ili predstavnika tehničke komisije u sustav, budući da oni predstavljaju poseban tip korisnika te ih treba odvojiti od natjecatelja, koji se mogu sami registrirati.



Dodavanje novog korisnika

Ime:

Prezime:

Email:

Korisničko ime:

Sifra:

Vrsta korisnika:

Dodatak Korisnika

Slika 7.44 Dodavanje korisnika

8. Zaključak i budući rad

Zadatak projekta bio je razviti web aplikaciju koja će služiti za prikaz rezultata i stanja stvarne rukometne lige, uz pružanje podrške za odigravanje virtualne lige između registriranih korisnika. Nakon tri mjeseca timskog rada, to je i ostvareno. Realizacija projekta bila je podijeljena u dvije faze.

U prvoj fazi obavljene su sve formalnosti poput formiranja i upoznavanja tima, iznošenja ideja te podjele uloga pojedinim članovima tima. Općenito, prva faza je služila kao priprema za konačno ostvarenje računalnog sustava. Mogli bismo reći da je u njoj napravljen kostur cijelog sustava. Taj kostur je ponajprije činilo dobro postavljanje funkcionalnih i nefunkcionalnih zahtjeva te ispravan odabir arhitekture i dizajna. Veliku pomoć pri ostvarenju druge faze predstavljali su razni obrasci, dijagrami i modeli poput obrazaca uporabe, sekvensijskih dijagrama, modela baze podataka, dijagrama razreda, dijagrama objekata i ostalih (koji su bili predviđeni za izradu u drugoj fazi projekta). Postojanje grafičkog prikaza apstraktnih modela uvelike je pomoglo prilikom njihovog programskog ostvarenja, ali i prije toga, prilikom komunikacije među članovima tima jer su stvari bile jasno opisane.

Druga je faza bila nešto kraća i u njoj je napravljena realizacija i finalizacija projekta. Prije toga, završen je kostur projekta koji je (uz sve prethodno napravljeno u prvoj fazi) uključivao izradu komunikacijskih dijagrama te dijagrama stanja, aktivnosti, razmještaja i komponenti. Iako je druga faza bila kraća, ona je isto tako bila i radno zahtjevnija. U skraćivanju potrebnog vremena je uvelike pomogla kvalitetna priprema projekta napravljena u prvoj fazi te na početku druge faze jer su sve stavke bile jasno i kvalitetno definirane. Sve u svemu, mogli bismo reći da je dodatno vrijeme utrošeno u prvoj fazi rezultiralo uštedom vremena u drugoj, a pritom nas sačuvalo od određenih nevolja tijekom realizacije projekta. Također, primjena znanja stečenih na predmetu pridonijela je uspješnom ostvarenju projekta.

Komunikacija između članova tima bila je konstantna, a ponajviše se odvijala preko društvenih mreža (Facebook), što je uvelike pridonijelo upoznatosti svih članova tima sa svim aspektima projekta. Prilikom rada na projektu, koristili smo sustav BitBucket za upravljanje raznim verzijama koda.

Kao neka od mogućih proširenja koja sad nismo stigli implementirati vidimo bolju verziju aplikacije za mobilne uređaje, što bi zasigurno imalo veliku primjenu u današnjem pokretnom svijetu. Također bi prikaz utakmica uživo, kao i proširenje na više liga (i virtualnih i stvarnih) bilo nešto što bi poprilično doprinjelo funkcionalnosti, ali i rasprostranjenosti naše aplikacije.

Sve u svemu, ovaj projekt bio je jedno zanimljivo i poučno iskustvo. Kao najzahtjevniji element pokazala se zapravo podjela zadataka i koordinacija posla između svih članova. Jednom kad smo to riješili, sve je išlo relativno glatko. Upravo zato, ovaj projekt je poslužio kao dobra priprema za rad „u stvarnom životu“ s kakvim se još nismo susreli na ovakvoj razini tijekom studija. Za kraj, zadovoljni smo što smo projekt uspješno priveli kraju i stekli nove vještine koje će zasigurno biti od koristi u našem dalnjem akademskom životu, ali i nakon studija.

9. Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. Oblikovanje programske potpore, FER ZEMRIS,
<http://www.zemris.fer.hr/predmeti/opp>
3. Apache Tomcat 7, <http://tomcat.apache.org/tomcat-7.0-doc>
4. Apache Derby 10.8, <http://db.apache.org/derby/manuals/> I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. Jboss Hibernate, <http://www.hibernate.org/docsUML 2 Class Diagram Guidelines>, <http://www.agilemodeling.com/style/classDiagram.htm>
6. Astah Community, <http://astah.net/editions/community>

Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

Slika 4.1 Sekvencijski dijagram za UC10 (UlogirajSe)	15
Slika 4.2 Sekvencijski dijagram za UC1 (PogledajStvarnoStanje)	17
Slika 4.3 Sekvencijski dijagram za UC2 (RegistrirajNatjecatelja)	18
Slika 4.4 Sekvencijski dijagram za UC3 (SastaviEkipu).....	19
Slika 4.5 Sekvencijski dijagram za UC4 (PogledajStanjeVirtualneLige)	20
Slika 4.6 Sekvencijski dijagram za UC5 (DefinirajPočetnuKonfiguraciju)	21
Slika 4.7 Sekvencijski dijagram za UC6 (UpišiDogađaje)	22
Slika 4.8 Sekvencijski dijagram za UC7 (ProglasiMVP).....	23
Slika 4.9 Sekvencijski dijagram za UC8 (AdministrirajBazuPodataka)	24
Slika 4.10 Sekvencijski dijagram za UC9 (Simuliraj)	25
Slika 4.11 Sekvencijski dijagram za UC10 (UlogirajSe)	26
Slika 6.1 Podsustavi arhitekture sustava	28
Slika 6.2 ER dijagram baze podataka.....	33
Slika 6.3 Dijagram razreda podpaketa modeli.liga	35
Slika 6.4 Dijagram razreda podpaketa modeli.korisnik i simulator	36
Slika 6.5 Dijagram razreda podpaketa DAO	37
Slika 6.6 Dijagram objekata Stvarna liga	39
Slika 6.7 Dijagram objekata Virtualna Liga	40
Slika 6.8 Dijagram aktivnosti registriranog natjecatelja	42
Slika 6.9 Dijagram stanja registriranog natjecatelja	44
Slika 6.10 Komunikacijski dijagram registriranje korisnika.....	45
Slika 6.11 Komunikacijski dijagram unošenje početne konfiguracije.....	45
Slika 6.12 Dijagram komponenti	47
Slika 7.1 Dijagram razmještaja.....	48
Slika 7.2 Implementacija DAO.java sučelja	50
Slika 7.3 Implementacija JPAEMProvider.java razreda	51
Slika 7.4 Isječak implementacije DAOImpl.java razreda	51
Slika 7.5 Implementacija jezgre Simuliraj.java razreda	52
Slika 7.6 Implementacija DogadjajiVjerljost.java razreda	53
Slika 7.7 Implementacija SimulatorServlet.java servleta.....	53
Slika 7.8 Isječak Main.java razreda za inicijalizaciju ekipa i lige	54

Slika 7.9 Baza nema potrebne tablice za rad web aplikacije	56
Slika 7.10 Baza ima potrebne tablice za rad web aplikacije	57
Slika 7.11 Isječak koda iz Main.java razreda koji stvara administratorski račun	58
Slika 7.12 Unos podataka administratora u formu za prijavu korisnika na web aplikaciju	58
Slike 7.13 Uspješna prijava na administratorski račun.....	59
Slika 7.14 Unos podataka sa strane administratora radi stvaranja računa službene osobe.....	60
Slika 7.15 Administrator je uspješno stvorio račun službene osobe.....	60
Slika 7.16 Ulaz u stranicu službene osobe u kojoj se učitava početna konfiguracija	61
Slika 7.17 Ako podatke unesemo krivo definicija nije uspješna.....	62
Slika 7.18 Konfiguracija je uspješno učitana nakon ispravno unesenih podataka	62
Slika 7.19 Ispis informacijske poruke nakon što službena osoba ponovo želi učitati početnu konfiguraciju.....	63
Slika 7.20 Nedozvoljen pristup stranici pravilno je obrađen.....	64
Slika 7.21 Administrator unosi podatke za stvaranje računa tehničke komisije	65
Slika 7.22 Račun tehničke komisije uspješno je napravljen.....	65
Slika 7.23 Administrator označava utakmice koje želi simulirati	66
Slika 7.24 Administartor pritiskom na gumb „Simulacija“ pokreće simulaciju odabralih utakmica.....	66
Slika 7.25 Nakon simulacije odmah vidimo rezultate simuliranih utakmica te znamo da je simulacija uspješno izvršena.....	67
Slika 7.26 Vidimo da je simulacija uspješna jer se ažurirao poredak i samo ekipe koje nisu igrale utakmicu nemaju bodove.....	67
Slika 7.27 Korisnik se registrira unosom svojih osobnih podataka	68
Slika 7.28 Nakon registracije korisnik bira sastav svoje virtualne ekipe	69
Slika 7.29 Nakon uspješne registracije i odabira sastava sustav omogućuje pregled ekipe	69
Slika 7.30 Novostvoreni račun je sada dio ostalih službenih tablica	69
Slika 7.31 Tehnička komisija ulazi u neku od odigranih utakmica te bira najboljeg igrača.....	70
Slika 7.32 Biranje najboljeg igrača uspješno je izvršeno te se on vidi pored rezultata utakmice	70
Slika 7.33 Poredak rukometne lige	74

Slika 7.34 Pregled odigranih utakmica po kolima.....	74
Slika 7.35 Statistika pojedinačne utakmice.....	75
Slika 7.36 Registracija neregistriranog korisnika.....	76
Slika 7.37 Poredak virtualne lige.....	77
Slika 7.38 Poredak virtualne lige po podržanim klubovima.....	77
Slika 7.39 Odabir virtualne ekipe	78
Slika 7.40 Bodovi koje je ostvario pojedini igrač	79
Slika 7.41 Definicija početne konfiguracije	80
Slika 7.42 Proglašavanje MVP-a.....	81
Slika 7.43 Raspored utakmica	82
Slika 7.44 Dodavanje korisnika	82

Dodatak B: Dnevnik sastajanja

Sastanak1: 19.10.2015. - FER

Prisutni: Goran Brlas, Filip Džidić, Antonio Ivčec, Vinko Kolobara, Vladimir Dragutin Livaja, Lovre Torbarina, Lovro Vidović

Sažetak: Upoznavanje članova tima i inicijalna diskusija o zadatku. Analiza ključnih točaka zadatka i formiranje potrebnih objašnjena funkcionalnosti.

Zaključci: Najavljen početak na formalnom opisivanju problema, funkcijskih i ostalih zahtjeva. Rad na UML dijagramima odgođen dok svi članovi grupe nisu upoznati s teorijskom pozadinom i potrebnim alatima.

Sastanak2: 21.10.2015. - FER

Prisutni: Goran Brlas, Filip Džidić, Antonio Ivčec, Vinko Kolobara, Vladimir Dragutin Livaja, Lovre Torbarina, Lovro Vidović

Sažetak: Sastanak s asistentom. Dodatna analiza projekta te rješavanje nedoumica oko projekta.

Zaključci: Početni dogovor oko podjele posla, dogovoren da će svaki član tima raditi na određenom dijelu dokumentacije u 1. ciklusu.

Sastanak3: 12.11.2015. - FER

Prisutni: Goran Brlas, Filip Džidić, Antonio Ivčec, Vinko Kolobara, Vladimir Dragutin Livaja, Lovre Torbarina, Lovro Vidović

Sažetak: Dogovor oko arhitekture sustava. Napravljen model razreda te model baze podataka.

Zaključci: Podjela posla oko izrade dijagrama na temelju napravljenog modela.

Sastanak4: 15.11.2015. - FER

Prisutni: Filip Džidić, Lovre Torbarina

Sažetak: Izrada dokumentacije arhitekture sustava te modela baze podataka.

Sastanak5: 17.11.2015. - FER

Prisutni: Antonio Ivčec, Vladimir Dragutin Livaja

Sažetak: Izrada dokumentacije dijagrama objekata.

Sastanak6: 22.12.2015. – FER

Prisutni: Goran Brlas, Filip Džidić, Antonio Ivčec, Vinko Kolobara, Vladimir Dragutin Livaja, Lovre Torbarina, Lovro Vidović

Sažetak: Podjela posla oko implementacije, dogovoren općeniti izgled aplikacije.

Sastanak7: 9.1.2016. – FER

Prisutni: Filip Džidić, Vinko Kolobara, Lovre Torbarina, Lovro Vidović

Sažetak: Implementacija aplikacije i testiranje za vrijeme implementacije.

Sastanak8: 12.1.2016. – FER

Prisutni: Filip Džidić, Vinko Kolobara, Lovro Vidović

Sažetak: Sastanak s asistentom. Prezentacija alfa inačice projekta.

Sastanak9: 15.1.2016 – FER

Prisutni: Goran Brlas, Antonio Ivčec, Vladimir Dragutin Livaja

Sažetak: Rad na dokumentaciji – dijagram aktivnosti i korisničke upute.

Sastanak10: 19.1.2016. – FER

Prisutni: Antonio Ivčec, Vladimir Dragutin Livaja, Lovre Torbarina

Sažetak: Rad na dokumentaciji – dijagrami stanja, komunikacijski, komponentni i dijagram razmještaja, dodatno testiranje gotove aplikacije.

Dodatak C: Prikaz aktivnosti grupe

Članovi grupe (abecednim redom)

Popis aktivnosti	Goran Brlas	Filip Džidić	Antonio Ivčec	Vinko Kolobara	Vladimir Dragutin Livaja	Lovre Torbarina	Lovro Vidović
Upravljanje projektom	100%						
Opis projektnog zadatka	100%						
Rječnik pojmova	40%	10%	10%	10%	10%	10%	10%
Opis funkcionalnih zahtjeva	50%						50%
Opis ostalih zahtjeva	100%						
Arhitektura i dizajn sustava							
Svrha, opći prioriteti i skica sustava		50%				50%	
Dijagram razreda s opisom				100%			
Dijagram objekata			50%		50%		
Ostali UML dijagrami	20%		40%		40%		
Implementacija i korisničko sučelje							

Dijagram razmještaja			50%		50%		
Korištene tehnologije i alati		100%					
Isječak programskog kôda						100%	
Ispitivanje programskog rješenja				33%		33%	33%
Upute za instalaciju	100%						
Korisničke upute	100%						
Plan rada	100%						
Pregled rada i stanje ostvarenja	100%						
Zaključak i budući rad	100%						
Popis literature	100%						
Dodaci							
Indeks	100%						
Dnevnik sastajanja	100%						

Dodatak D: Plan rada

Ostvareno u rev. 1:

- Opis projektnog zadatka,
- Funkcionalni zahtjevi,
- Ostali zahtjevi,
- Arhitektura i dizajn sustava.

Ostvareno u rev. 2:

- Implementiran sustav stvarne i virtualne rukometne lige,
- UML dijagrami: aktivnosti, stanja, komunikacijski, komponentni, razmještaja,
- Upute za instalaciju (dodatci E i F),
- Bitni isječci koda, dokumentiranje ispitivanja programske potpore,
- Upute za intalaciju,
- Korisničke upute.

Dodatak E: Apache Derby

Kako bi Apache Derby pravilno radio potrebno je imati instaliran Java Development Kit koji se može preuzeti na slijedećoj poveznici:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Program se jednostavno instalira pokretanjem preuzete datoteke te odabirom željene odredišne mape.

Nakon instalacije potrebno je podesiti neke parametre kako bi program pravilnom radio. Stoga otvorite Command Prompt i unosite redom slijedeće naredbe

(NAPOMENA! u ovom primjeru korišten je JDK 1.8, ukoliko preuzimate i instalirate drugačiju verziju, naziv mape JDK je drugačiji!):

WINDOWS:

```
set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_65
```

```
set PATH=%PATH%;%JAVA_HOME%\bin
```

UNIX:

```
export JAVA_HOME=/opt/jdk1.8.0_65
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

Po završetku instalacije JDK slijedi instalacija Apache Derby-ja. Program možete preuzeti na poveznici: http://db.apache.org/derby/derby_downloads.html

Potrebno je preuzeti jednu od slijedećih datoteka:

Windows: db-derby-10.9.1.0-bin.zip

UNIX, Linux, and Mac: db-derby-10.9.1.0-bin.tar.gz

Nakon preuzimanja datoteke, datoteku raspakirajte (koristeći WinZip, Winrar, 7-zip, i sl.) u željenu mapu (u našem primjeru C:\Apache). Sada je potrebno izvršiti

konfiguraciju parametara kako bi Derby radio kako spada. Započinjemo slijedećim naredbama kojima dovršavamo instalaciju:

WINDOWS:

```
set DERBY_INSTALL=C:\Apache\db-derby-10.9.1.0-bin
```

UNIX:

```
$ export DERBY_INSTALL=/opt/Apache/db-derby-10.9.1.0-bin
```

Slijedi konfiguracija putanje:

WINDOWS:

```
set
```

```
CLASSPATH=%DERBY_INSTALL%\lib\derby.jar;%DERBY_INSTALL%\lib\derbytools.jar;
```

UNIX:

```
export
```

```
CLASSPATH=$DERBY_INSTALL/lib/derby.jar:$DERBY_INSTALL/lib/derbytools.jar:
```

```
ar:
```

Koju nastavljamo naredbama:

WINDOWS:

```
C:\> cd %DERBY_INSTALL%\bin
```

```
C:\Apache\db-derby-10.9.1.0-bin\bin> setEmbeddedCP.bat
```

UNIX:

```
$ cd $DERBY_INSTALL/bin
```

```
$ . setEmbeddedCP.ksh
```

Da bi provjerili je li instalacija uspješna unosimo slijedeću naredbu:

```
java org.apache.derby.tools.sysinfo
```

Ako je ispis sadržan od Java Information i Derby Information, instalacija je uspješna i možemo nastaviti dalje s konfiguracijom. No, ukoliko je ispis slijedeći, došlo je do greške i potrebno je poduzeti korake koji su navedeni ispod:

GREŠKA:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
```

```
org/apache/derby/tools/sysinfo
```

WINDOWS:

```
C:\> echo %CLASSPATH%
```

```
C:\Apache\DB-DER~1.1-B\lib\derby.jar;C:\Apache\DB-DER~1.1-
B\lib\derbytools.jar;
```

UNIX:

```
$ echo $CLASSPATH
```

```
/opt/Apache/db-derby-10.9.1.0-bin/lib/derby.jar:/opt/Apache/db-derby-10.9.1.0-
bin/lib/derbytools.jar:
```

Nakon postavljanja putanje otvaraju nam se sljedeće mogućnosti:

1. Pokretanje ij alata:

```
java org.apache.derby.tools.ij
```

Ispis bi trebao biti slijedeći:

```
ij version 10.4
```

```
ij>
```

Ukoliko je došlo do greške, potrebno je ponovno postaviti putanju.

2. Kreiranje baze podataka:

```
ij> connect 'jdbc:derby:MyDbTest;create=true';
```

```
ij> exit;
```

3. Spajanje na bazu podataka:

```
java org.apache.derby.tools.ij
```

```
ij> connect 'jdbc:derby:MyDbTest';
```

4. Sada možete izvoditi naredbe:

```
ij> create table derbyDB(num int, addr varchar(40));
```

```
ij> insert into derbyDB values (1956,'Webster St.');
```

```
ij> insert into derbyDB values (1910,'Union St.');
```

```
ij> update derbyDB set num=180, addr='Grand Ave.' where num=1956;
```

```
ij> select * from derbyDb;
```

ili pokretati skripte:

```
ij> run 'my_file.sql';
```

Na kraju, potrebno je još pokrenuti Network Server, a to radimo slijedećim nizom naredbi:

WINDOWS:

C:\> set

```
CLASSPATH=%DERBY_INSTALL%\lib\derbytools.jar;%DERBY_INSTALL%\lib\derby.jar;
```

ynet.jar;

C:\Apache\db-derby-10.4.1.3-bin\bin> startNetworkServer.bat

Security manager installed using the Basic server security policy.

Apache Derby Network Server - 10.4.1.3 - (648739) started and ready to accept connections on port 1527 at 2008-04-28 17:13:13.921 GMT

UNIX:

```
$ export CLASSPATH=$DERBY_INSTALL/lib/derbytools.jar:
```

```
$DERBY_INSTALL/lib/derbynet.jar:
```

\$ startNetworkServer.ksh

Security manager installed using the Basic server security policy.

Apache Derby Network Server - 10.4.1.3 - (648739) started and ready to accept connections on port 1527 at 2008-04-30 09:35:55.871 GMT

Nakon izvođenja ovih naredbi spremni ste za korištenje Apache Derby-ja.

Dodatak F: Apache Tomcat

Instalaciju Apache Tomcat-a na operativni sustav započinjemo preuzimanjem najnovije verzije proizvoda sa službene stranice, i to klikom na slijedeći link : <http://tomcat.apache.org/download-70.cgi> te odabirom zip datoteke (prva datoteka) ispod podnaslova Binary Distributions -> Core.

Jednom preuzetu datoteku potrebno je raspakirati na željeno mjesto na sustavu. Nakon uspješnog preuzimanja i raspakiravanja Tomcat-a najprije je potrebno postaviti path za izvršnu datoteku. Za to je potrebno imati instalirane najnovije verzije Java jdk i jre datoteka. Dotične se datoteke mogu preuzeti na linku: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Datoteke je dovoljno preuzeti i pokrenuti, odabratи željenu mapu za spremanje datoteka te će se instalacija izvršiti sama. Po završetku instalacije radimo path. Potrebno je na računalu odabratи Start, Computer (My Computer) a zatim pritiskom na desnu tipku miša odabratи Properties. Kada se otvorи novi prozor na lijevoj strani ponudi se izbornik u kojem je potrebno odabratи Advanced system settings nakon čega iskače prozor sa novim opcijama. Potrebno je pozicionirati se na tab Advanced te odabratи Environment Options. Otvara se još jedan prozor u kojem je ispod System variables potrebno odabratи New. Pod Variable name potrebno je upisati JAVA_HOME, a pod Variable value potpunu putanju do jdk datoteke (primjerice C:\User\Program files\jdk1.7.0) te zatim kliknuti Ok i to ponoviti u svim prozorima.

Nakon postavljene putanje potrebno se pozicionirati u mapu u koju smo spremili Apache Tomcat prilikom instalacije (npr. C:\User\apache-tomcat-7.0.34), a zatim i otvoriti mapu bin. Unutar mape bin potrebno je dvostrukim klikom pokrenuti datoteku startup.bat, čime se pokreće rad Tomcat poslužitelja.

Pristupni podaci za razne uloge u radu poslužitelja mogu se vidjeti i podešavati u datoteci tomcat-users.xml, unutar mape conf u mapi u kojoj je instaliran Tomcat

poslužitelj. Standardni port koji Tomcat koristi je 8080, no to se može podesiti u datoteci conf/server.xml .