

Razvoj web aplikacije za vođenje nogometnog kluba

Bašić, Sebastian

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **The Polytechnic of Rijeka / Veleučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:125:847638>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-18**



Repository / Repozitorij:

[Polytechnic of Rijeka Digital Repository - DR PolyRi](#)



VELEUČILIŠTE U RIJECI

Sebastian Bašić

RAZVOJ WEB APLIKACIJE ZA VOĐENJE NOGOMETNOG KLUBA

(završni rad)

Rijeka, 2020.

VELEUČILIŠTE U RIJECI

Poslovni odjel

Preddiplomski stručni studij Informatika

RAZVOJ WEB APLIKACIJE ZA VOĐENJE NOGOMETNOG KLUBA

(završni rad)

MENTOR:

izv. prof. dr. sc. Alen Jakupović, prof. v. š.

STUDENT:

Sebastian Bašić

MBS: 2422000021/15

Rijeka, rujan, 2020.

VEUČILIŠTE U RIJECI

Poslovni odjel

Rijeka, 9.5.2020.

**ZADATAK
za završni rad**

Pristupnik Sebastian Bašić, MBS: 2422000021/15.

**Studentu preddiplomskog stručnog studija Informatika izdaje se zadatak za završni rad –
tema završnog rada pod nazivom:**

**RAZVOJ WEB APLIKACIJE ZA
VOĐENJE NOGOMETNOG KLUBA**

Sadržaj zadatka:

Opisati tehnologije i alate koji su korištenu u razvoju web aplikacije: HTML, CSS, PHP, JavaScript, Visual Studio Code, Laravel, Vue, Bootstrap. Provesti analizu i izraditi dijagram dekompozicije i model procesa (primjenom DTP-a). Provesti dizajn i izraditi model podataka (primjenom EVA) i arhitekturu programske podrške. Opisati važnije implementirane algoritme. Detaljno opisati uporabu gotove aplikacije.

Preporuka:

Rad obraditi sukladno odredbama Pravilnika o završnom radu Veleučilišta u Rijeci.

Zadano: 9.5.2020.

Predati do: 15.9.2020.

Mentor:



(izv.prof.dr.sc. Alen Jakupović, prof.v.š.)

Pročelnik odjela:



(mr.sc. Anita Stilin, v. pred.)

Zadatak primio dana: 9.5.2020.

Pristupnik:



(Sebastian Bašić)

Dostavlja se:

- mentoru
- pristupniku

IZJAVA

Izjavljujem da sam završni rad pod naslovom RAZVOJ WEB APLIKACIJE ZA VOĐENJE NOGOMETNOG KLUBA izradio samostalno pod

nadzorom i uz stručnu pomoć mentora: izv.prof.dr.sc. Alen Jakupović prof. v.š.

Sebastian Bašić



(potpis studenta)

SAŽETAK

U ovom radu je dokumentirana izrada informacijskog sustava za nogometne klubove od baze podataka do korisničkog sučelja. Aplikacija je izrađena pomoću tri frameworka. Prvi je Laravel, korišten za izradu back end, api dijela aplikacije na php-u. Drugi framework je Vue, korišten za izradu front end dijela aplikacije. Posljednji framework je Bootstrap, koji je korišten za stiliziranje grafičkog korisničkog sučelja aplikacije i stvaranje responzivnog dizajna. Podaci informacijskog sustava se čuvaju u relacijskoj bazi podataka.

Ključne riječi: informacijski sustav, Laravel, Bootstrap, Vue, baza podataka

SADRŽAJ

1. UVOD	1
2. KORIŠTENE TEHNOLOGIJE I ALATI.....	2
2.1. HTML.....	2
2.2. CSS STILOVI.....	3
2.3. PHP PROGRAMSKI JEZIK	7
2.4. JAVASCRIPT.....	8
2.5. VISUAL STUDIO CODE	9
2.6. LARAVEL FRAMEWORK.....	10
2.7. VUE FRAMEWORK.....	12
2.8. BOOTSTRAP FRAMEWORK	15
3. OPIS SUSTAVA	18
3.1. OPĆI OPIS.....	18
3.2. SPECIFIČNI PRIMJERI.....	20
4. DIJAGRAM DEKOMPOZICIJE FUNKCIJA	22
5. DIJAGRAM TOKA PODATAKA	24
5.1. DIJAGRAM TOKA PODATAKA NULTE RAZINE.....	24
5.2. DIJAGRAM TOKA PODATAKA PRVE RAZINE.....	26
5.3. DIJAGRAM TOKA PODATAKA DRUGE RAZINE	29
6. OPIS TOKOVA I SKLADIŠTA PODATAKA.....	32
6.1. PROCES - EVIDENCIJA TRENERA.....	32
6.2. PROCES - EVIDENCIJA ČLANOVA	33
6.3 PROCES - EVIDENCIJA KATEGORIJA.....	34
6.4. PROCES - EVIDENCIJA OPREME	35
6.5. PROCES - EVIDENCIJA TRENINGA	36
6.6. PROCES - EVIDENCIJA ČLANARINA	37
6.7 PROCES - EVIDENCIJA UTAKMICA	38
6.8. PROCES - EVIDENCIJA POGODAKA	39
6.9 PROCES - EVIDENCIJA KARTONA	40
7. MODEL PODATAKA - EVA MODEL	41

7.1. EVA MODEL	41
7.2. OPIS TIPOVA ENTITETA	42
7.3. OPIS ATRIBUTA TIPOVA ENTIETA	43
7.4. OPIS VEZA EVA MODELA	46
8. RELACIJSKI MODEL PODATAKA	47
8.1. POPIS RELACIJSKIH SHEMA	47
8.2. RELACIJSKI MODEL	49
9. ARHITEKTURA PROGRAMSKE PODRŠKE	50
9.1. DIJAGRAM PROGRAMSKE PODRŠKE (DPP)	50
9.2. TEKSTUALNI OPIS DPP DIJAGRAMA	51
10. IMPLEMENTACIJA POSLOVNE LOGIKE	53
11. OPIS I IMPLEMENTACIJA IZVJEŠTAJA	54
11.1. IZVJEŠTAJ TRENINGA	54
11.2. IZVJEŠTAJ ČLANARINA	54
11.3. IZVJEŠTAJ ČLANOVA	54
11.3. IZVJEŠTAJ TRENERA	55
11.4. IZVJEŠTAJ OPREME	55
11.5. IZVJEŠTAJ KATEGORIJA	55
11.6. IZVJEŠTAJ UTAKMICA	55
11.7. IZVJEŠTAJ POGODAKA	56
11.8. IZVJEŠTAJ KARTONA	56
12. PRIKAZ KORIŠTENJA PROGRAMSKOG RJEŠENJA	57
13. ZAKLJUČAK	61

1. UVOD

Programsko rješenje Informacijski sustav nogometnog kluba služi za evidenciju osnovnih podataka nogometnog kluba i izvješćivanje o upisanim podacima korisnicima rješenja. Korisnici ovog sustava su treneri i administratori kluba u kojemu se sustav koristi. Sustav je baziran na webu i izrađen je uz pomoć Laravel, Vue i Bootstrap frameworka. Sastoji se od baze podataka, programske logike i korisničkog sučelja koje se prikazuje u internet pregledniku.

Razlog odabira ove teme je moj interes prema istraživanju web aplikacija. Smatram da će web aplikacije biti dugo vremena aktualne i zainteresiran sam za traženje posla u tom području.

2. KORIŠTENE TEHNOLOGIJE I ALATI

2.1. HTML

HTML, (*Hypertext Markup Language*) je jezik za definiranje strukture web stranica. Pomoću HTML-a možemo međusobno povezivati više dokumentata neovisno o platformi. Jedna od prednosti HTML-a je ta što, za razliku od programskih jezika, može biti interpretiran koristeći bilo koji web pretraživač ili tekst procesor.

Prema knjizi Sams Tech HTML and XHTML in 10 Minutes (*Hayes, 2002: 7*) Web pretraživači interpretiraju HTML dokument identificirajući i odvajajući određene elemente unutar dokumenta. HTML elementi su označeni HTML oznakama (*eng. Tags*). Većina HTML elemenata je označena sa dvije oznake (*eng. double tag*), ali postoje i elementi koji su označeni samo jednom oznakom (*eng. Single tag*).

HTML elementima se mogu pridruživati određeni atributi koji nam donose dodatne informacije o pojedinom elementu. Attribute upisujemo unutar otvarajuće oznake elementa i sastoje se od dva dijela, a to su naziv i vrijednost između kojih se nalazi znak jednakosti.

Duckett (*HTML & CSS: Design and Build Web Sites, 2011: 27*) govori kako se prosječni HTML dokument u osnovi sastoji od `<html></html>` oznake koja određuje granice dokumenta, `<head></head>` oznake i `<body></body>` oznake. `<head></head>` oznaka sadrži informacije koje se ne prikazuju u samoj strukturi web stranice, ali su bitne web pregledniku. Unutar `<head></head>` oznake se često nalazi `<title></title>` oznaka koja u pretraživaču definira naslov stranice koji će se prikazati pri vrhu pretraživača.

2.2. CSS STILOVI

CSS(*Cascading Style Sheets*) je jezik za stiliziranje HTML elemenata. Nastao je radi potrebe za boljim stiliziranjem web stranica. U početku weba, koristili su se HTML elementi i njihovi atributi kako bi se uredio sadržaj web stranice. Kako je potreba za boljim i dinamičnijim izgledom web stranica rasla, tako je HTML kod postajao sve složeniji i teži za razumijevanje. Iz tog razloga je razvijen CSS.

Pojavom CSS-a, postalo je moguće odvojiti strukturu sadržaja na web stranici od njezina dizajna. CSS određuje način na koji će se određeni HTML element prikazati u pregledniku. Moguće je odraditi boju teksta elementa, boju pozadine, veličinu fonta i brojne druge mogućnosti.

Prema knjizi *CSS Mastery (Budd, Moll, Collison, 2006: 11-12)* CSS pruža jednostavan način odabiranja i uređivanja pojedinih HTML elemenata. CSS kod sastoji se od pravila stila (*eng. Style rule*). Svako pravilo sastoji se od selektora i deklaracije. Uobičajeni selektori su selektor elementa (selektor tipa), selektor nasljednika, klasni selektor i selektor identifikatora.

Slika 1: Primjer uobičajenih selektora

```
5 <style type="text/css">
6   /*selektor tipa*/
7   p{
8     height: 100px;
9     width: 100px;
10  }
11
12  /*Selektor nasljednika*/
13  p a{
14    height: 100px;
15    width: 100px;
16  }
17
18  /*Klasni selektor*/
19  .klasaA{
20    height: 100px;
21    width: 100px;
22  }
23
24  /*Selektor identifikatora*/
25  #id{
26    height: 100px;
27    width: 100px;
28  }
29 </style>
```

Izvor: autor

Selektor elementa označava, odnosno utječe na elemente određenog tipa (npr. odlomak `<p></p>` ili naslov `<h1></h1>`). Selektori nasljednika utječu na elemente koji se nalaze unutar određenog elementa. Na primjer, ako imamo element `<p></p>` unutar kojeg se nalazi element `<a>` možemo postaviti selektor tako da utječe samo na elementa `<a>` koji se nalaze unutar elementa `<p></p>`. Ako u dokumentu postoji koji element `<a>`, a da se ne nalazi unutar elementa `<p></p>`, neće biti uređen. Klasni selektori uređuju samo elemente kojima je dodijeljena određena klasa. Selektori identifikatora će uređivati samo elemente kojima je dodijeljen određeni identifikator (*eng. id*).

Prema Duckett (*HTML & CSS: Design and build websites, 2011: 231-232*) Deklaracije definiraju na koji način bi se element odabran selektorom trebao uređivati. One se sastoje od dva dijela, a to su parametar (*eng. property*) i vrijednost (*eng. value*). Parametar određuje na koji način želimo oblikovati element, a vrijednost u kojem obimu, koliko želimo oblikovati element, koju boju želimo odabrati i sl...

Stilove koje smo definirali unutar CSS koda moramo povezati sa HTML kodom kako bi oni funkcionirali. To možemo učiniti na tri načina. Prvi način je da definiramo CSS stil unutar samog elementa. HTML elementu se pridružuje atribut `style` unutar kojeg možemo definirati stil koji želimo promijeniti. U ovom slučaju, stil će se odnositi samo na element unutar kojeg smo ga definirali. Drugi način da definiramo CSS stilove unutar HTML dokumenta, unutar `<style></style>` oznaka. Tada se stil odnosi na cijeli HTML dokument unutar kojeg smo ga definirali. Treći način je da uključimo vanjski CSS dokument u HTML dokument koji želimo stilizirati. To možemo učiniti putem `<link></link>` oznaka (*eng. tagova*). Prednost ove metode je to što CSS dokument možemo povezati sa više HTML dokumenata i on će ih sve oblikovati na isti način.

Metoda sa vanjskim CSS dokumentom je iznimno korisna kod izrade web sjedišta sa puno stranica koje sve imaju ujednačen stil. Metoda uključivanja putem `<style></style>` oznaka koristit će se u slučaju kada na određenoj stranici trebamo urediti stranice koje će se po nečemu razlikovati od ostalih stranica na web sjedištu. A metoda uključivanja CSS stilova unutar elementa, koristit će se onda kada je potrebno posebno oblikovati samo određen element na jednoj stranici.

Slika 2: Primjer - metoda povezivanja CSS-a i HTML-a unutar dokumenta

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Stranica</title>
5 <style type="text/css">
6 |   h1{
7 |     background-color: blue;
8 |   }
9 </style>
10 </head>
11 <body>
12
13 <h1>Naslov</h1>
14 <p>Odlomak</p>
15
16 </body>
17 </html>
```

Izvor: autor

Slika 3: Primjer - metoda povezivanja CSS-a i HTML-a unutar elementa

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Stranica</title>
5 </head>
6 <body>
7
8 <h1 style="background-color: blue;
9 |   font-size: 10px;">Naslov</h1>
10 <p>Odlomak</p>
11
12 </body>
13 </html>
```

Izvor: autor

Slika 4: Primjer - metoda povezivanja CSS-a i HTML-a iz vanjskog dokumenta

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Stranica</title>
5 <link rel="stylesheet" type="text/css"
6   href="stil.css">
7 </head>
8 <body>
9
10 <h1>Naslov</h1>
11 <p>Odlomak</p>
12
13 </body>
14 </html> |
```

Izvor: autor

2.3. PHP PROGRAMSKI JEZIK

PHP (*Hypertext procesor*) je skriptni jezik za izradu web aplikacija i izvršava se na serverskoj strani. Koncipiran je 1994.godine od strane Rasmusa Lerdorfa.

Kao što Lerdorf, Tatroe i MacLntyre tvrde (*Programing PHP, 2006: 1-2*), PHP je moguće koristiti na svim većim operacijskim sustavima i vodećim web preglednicima. Vrlo je fleksibilan i može podržati širok spektar baza podataka (npr. MySQL, Postgre SQL, Oracle, Sybase i baze podataka kompatibilnosti s ODBC-om). PHP sadrži biblioteku PHP koda PEAR(*PHP Extension and Application Repository*) koja sadrži potrebne funkcionalnosti za obavljanje uobičajenih poslova na PHP-u i brojne ponovno uporabljive PHP komponente.

Kao što tvrdi Baniček (*PHP/MySQL, 2004: 69*) činjenica da se PHP izvršava na serveru daje mu prednost, jer to znači da nije ovisan o kompatibilnosti uređaja korisnika. Pisanje koda također je jednostavno s obzirom da nam za to nije potreban nikakav poseban alat do uređivača teksta.

PHP skripte se uvijek, radilo se to o vanjskoj datoteci ili unutar HTML-a, pišu unutar posebnih PHP oznaka.

U ovom projektu, PHP je korišten u sklopu Laravel frameworka koji će biti kasnije objašnjen. PHP je korišten za dohvaćanje podataka sa serverske strane i isporučavanja podataka kroz poslovnu logiku front end dijelu aplikacije koje je izrađena u Vue frameworku. Također su se preko PHP-a slali zahtjevi na server.

2.4. JAVASCRIPT

Javascript je skriptni jezik koji se izvršava na klijentskoj strani. Ugrađuje se u HTML kod korištenjem `<script></script>` oznaka unutar zaglavlja ili tijela dokumenta. Javascript pomaže kako bi dinamizirali web stranicu korisniku i omogućiti izmjenu sadržaja web stranice bez dohvaćanja skripti sa servera. JavaScript može pristupati pojedinim HTML elementima i dinamički ih mijenjati. Sve se odvija na strani klijenta odnosno korisnika.

Kako tvrdi Powers (*Learning JavaScript, 2009: 1*) jedan od razlog popularnosti JavaScripta je vrlo lako dodavanje skripti web stranicama. Instalacija nije potrebna, niti je potrebno konfigurirati putanje do raznih biblioteka. Javascript također funkcionira u većini web preglednika.

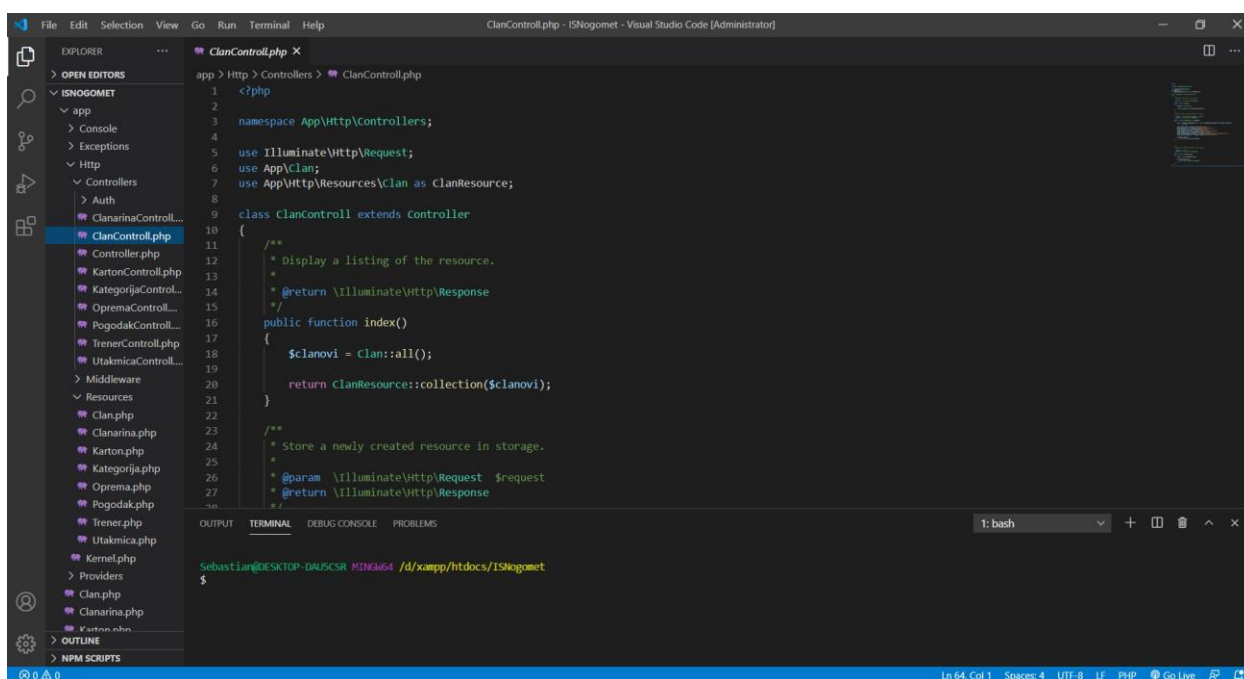
Iako se skripta može smjestiti unutar zaglavlja ili tijela dokumenta u knjizi Learning JavaScript (*Powers, 2009: 1-5*) savjetuje se da se da skriptu ne smještamo unutar tijela dokumenta, s obzirom da će ga tamo biti teže kasnije pronaći, ali i kako je ipak dobro uključiti skriptu unutar tijela iz drugog ih razloga. Ako smo smjestili skriptu u zaglavlje, preuzimanje ostatka HTML dokumenta može se završiti prije učitavanja skripte. Osim toga, preglednik može zaustaviti iscrtavanje stranice jer se unutar skripte možda nalazi `document.write` metoda. Ako je JavaScript skripta prevelika, slike na web stranici i druge informacije mogu kasniti. Postoje još mnoge prednosti kada su u pitanju performanse.

„Dinamičke Web stranice izrađene putem JavaScripta, ranije zvane DHTML ili dinamički HTML, a sada Ajax, prisutne su već oko 12 godina. Ključni elementi za ovaj koncept bilo je uvođenje kaskadnih opisa stilova (*eng. Cascading Style Sheets, CSS*) i DOM-a“ (*Powers, 2009: 261*).

2.5. VISUAL STUDIO CODE

Visual studio code je besplatni source code editor podržan na Windows, macOS i Linux operacijskim sustavima. Podržava JavaScript, TypeScript, Node.js i sadrži brojne dodatke za druge programske jezike kao što su C++, C#, Java, Python, PHP, Go...

Slika 5: Grafičko korisničko sučelje Visual Studio Code-a



Izvor: autor

2.6. LARAVEL FRAMEWORK

Laravel je open source PHP framework za web aplikacije. Developeri laravela nastoje učiniti programiranje na Laravelu ugodnim i jednostavnim, a da ne umanjuju kvalitetu samog rada. Laravel je vrlo popularan framework, čak i među programerima koji ne vole PHP. Baziran je na model-view-controll modelu aplikacije što ga čini vrlo jasnim i jednostavnim.

Prema knjizi *Laravel: up and Running (Stauffer, 2017: 78)* laravel nam nudi dva formata view-a koja možemo koristiti: plain PHP ili Blade template. U ovom projektu je korišten Blade template koji se pokreće preko blade engine-a, iako vrlo malo, s obzirom da se cijeli front end bazira na Vue frameworku.

Controller-i su klase koje organiziraju logiku svih rout-a određenog modela na jednom mjestu. Unutar controller-a, slične route-e se grupiraju zajedno. U slučaju ove ovog projekta, razvijena je obična CRUD aplikacija, tako da controller može rukovati svim akcijama nad određenim modelom, odnosno resursom. Resursi su šablone po kojima možemo oblikovati podatke koji se dobivaju iz modela i prosljeđivati ih serveru. Podatke možemo formatirati kako god želimo i takvim podacima dalje rukovati.

Jedna od značajka Laravel Frameworka koja je korištena u izradi ove aplikacije je ORM (*Obect-relational mapping*) Eloquent model. Putem eloquent-a možemo kreirati, uređivati i brisati tablice i raditi sve operacije s tablicama koje bi inače radili preko MySQL-a. Svaka tablica u bazi podataka ima svoj pripadajući eloquent model objekt. Preko tog objekta možemo unutar php koda upravljati tablicom koja se nalazi u bazi podataka.

Laravel rukuje routingom na vrlo jednostavan način. Dovoljno je specificirati url i tip zahtjeva koji želimo poslati na server.

Laravel također dodaje sloj sigurnosti u aplikaciju. Stoji između zahtjeva koji se šalju na server i samog servera i brine se da ne dođe do neovlaštenog pristupa.

Vrlo je jednostavno raditi s bazama podataka. Laravel dolazi u paketu sa Artisanom, jednostavnim command line tool-om. Uz pomoć Artisana je moguće na jednostavan način obavljati razne zadatke vezane za bazu podataka i projekt općenito kao što su migracije ili upisivanje testnih podataka. Iako migracije nisu korištene u izradi ove aplikacije, Artisan je korišten za brzo i jednostavno stvaranje već pripremljenih model, view i controller datoteka.

Laravel čini izradu autentifikacije korisnika vrlo jednostavnom i svaki Laravel projekt dolazi sa već spremnim migracijama i datotekama potrebnim za izradu autentifikacije. Laravel također pruža pomoć pri upravljanju sesijama, potvrđivanju email adresa, upravljanju, cache memorijom i mnogim drugim operacijama.

Kako tvrdi Stauffer (*Laravel: up and Running, 2017: 28*), dvije glavne vrijednosti Laravel Frameworka su brzina razvoja aplikacija i zadovoljstvo programera. Kreatori laravela išli toliko daleko da su se pobrinuli sama terminologija unutar Laravel Frameworka zvuči opuštajući, stoga smo dobili nazive tipa Artisan ili Eloquent.

Jedan od glavnih atributa Laravela je čistoća i ljepota koda. Glavni kreator Laravela Taylor Otwell vjeruje da je ljepota koda ključ uspješnog programiranja. U knjizi *Laravel: up and Running (Stauffer, 2017: 28)*, Stauffer citira Taylora Otwella: „Ponekad trošim smiješno velike količine vremena, mučeći se da učinim kod ljepšim.“

Sušтина Laravela je omogućiti programerima čist, jednostavan i lijep kod, uz funkcionalnosti koje pomažu programerima da brzo nauče, pisati jednostavan i čist kod.

2.7. VUE FRAMEWORK

Vue je progresivni framework specijaliziran za izradu grafičkog korisničkog sučelja. Vrlo je prilagodljiv i njegova glavna biblioteka (*eng. library*) je fokusirana samo na prezentacijski sloj aplikacije. Vue je lako uklopiti u druge projekte.

Vue koristi JavaScript, tako da se sav front end očitava u web pregledniku. Grafičko korisničko sučelje je dinamično i nema potrebe za dohvaćanjem stranica sa servera. Vue omogućuje korištenje više reusable komponenti od kojih svaka ima svoj JavaScript kod, HTML i CSS. Te komponente onda možemo pozivati po želji.

Najčešće korištena Vue značajka u ovom projektu je korištenje Vue direktiva. Pomoću Vue direktiva možemo umetati JavaScript vrijednosti u HTML. Moguće je koristiti for petlju i if, else naredbe unutar HTML koda, prikazivati ili sakriti određeni dio korisničkog sučelja ovisno o određenom uvjetu i dodijeljivati dinamičke klase.

Vue sve komponente sažima u jednu datoteku. Rad u Vue Frameworku započinjemo stvaranjem Vue instance. Svaka instanca uređuje određeni div HTML element, tako da se sav kod sažima jedan HTML element.

Jedan od bitnih koncepata Vue frameworka je obostarno povezivanje podataka (*eng. Two-way data binding*). Prema knjizi *Vue.js: Up and Running (Macrae, 2018: 17-18)*, u Vue-u je moguće umetati vrijednosti iz JavaScript koda u html kod putem direktive `v-bind`, ali tu se radi samo o jednostranom povezivanju podataka (*eng. One way data binding*). Kako bi ostvarili obostrano povezivanje podataka, potrebno je koristiti direktivu `v-model`.

Pomoću v-model direktive moguće je npr. povezati nekakav tekstualni okvir sa određenom varijablom unutar JavaScript skripte. Ako se unutar neke metode varijabla promijeni, promjenit će se i ispis na tekstualnom polju, a ako korisnik izmjeni tekstualno polje, izmijeniti će se i vrijednost varijable.

Jedan od često korištenih Vue direktiva, kako u svijetu, tako i u ovom projektu je v-bind direktiva. Kako je već spomenuto, pomoću nje je moguće povezivati javascript varijable i HTML. Direktivu v-bind možemo dodijeliti atributima HTML elemenata i tako ih povezati sa željenim varijablama.

Kako Macrae tvrdi (*Vue.js: Up an Running, 2018: 69*) vrlo česta uporaba v-bind (koju skraćeno možemo unutar koda označavati dvotočkom) direktive je upravo na klasama HTML elemenata. Tim putem moguće je dinamično izmijenjivati vrijednost klasa na temelju vrijednosti određenih varijabli unutar JavaScript-a. Tako možemo vrlo lako i dinamično izmijenjivati izgled stranice bez pozivanja na server. Moguće je i proslijediti čitavo polje (*eng. Array*) klasa i sve će se klase udružiti na tom elementu.

Također je moguće prosljeđivanje json objekata pomoću v-bind direktive. Tako će u objektu svaka klasa imati svoj naziv i vrijednost. Vrijednost može biti *true* ili *false*. Ako je vrijednost *false*, klasa će biti isključena, ako je *true* bit će prikazana.

Slika 6: Prikaz dodjeljivanja klasa putem v-bind direktive

```
<div class="jumbotron jumbotron-fluid">
  <div id="naslov" class="">
    <h1 class="display-4">Informacijski sustav nogometnog kluba</h1>
    <p class="lead">Evidencija treninga</p>
    <!-- Unutrašnji navbar -->
    <nav class="nav nav-pills nav-justified">
      <a @click="promjeniMeni('meni-evidencije', 'evidencija')" :class="{ 'nav-link': navLink1, active: active1}" href="#"><router-link class="link" to="/evidencije">Evidencije</router-link></a>
      <a @click="promjeniMeni('meni-izvjestaji', 'izvjestaji')" :class="{ 'nav-link': navLink2, active: active2}" href="#"><router-link class="link" to="/izvjestaji">Izvjestaji</router-link></a>
    </nav>
  </div>
</div>
<component :is="meni"></component>
</div>
</template>
</script>
import MeniEvidencije from "./MeniEvidencije"
import MeniIzvjestaji from "./MeniIzvjestaji"
```

Izvor: autor

2.8. BOOTSTRAP FRAMEWORK

Bootstrap je besplatan i open source CSS framework za izradu responzivnog dizajna grafičkog korisničkog sučelja u web aplikacijama. Sadrži CSS i JavaScript podržane elemente kao na primjer HTML prijedlošci za forme, gumbe, navigaciju i sl.

Bootstrap je u ovom projektu korišten u svojoj CLI verziji. Svi korišteni Bootstrap elementi mogu se pronaći na stranicama Bootstrap frameworka.

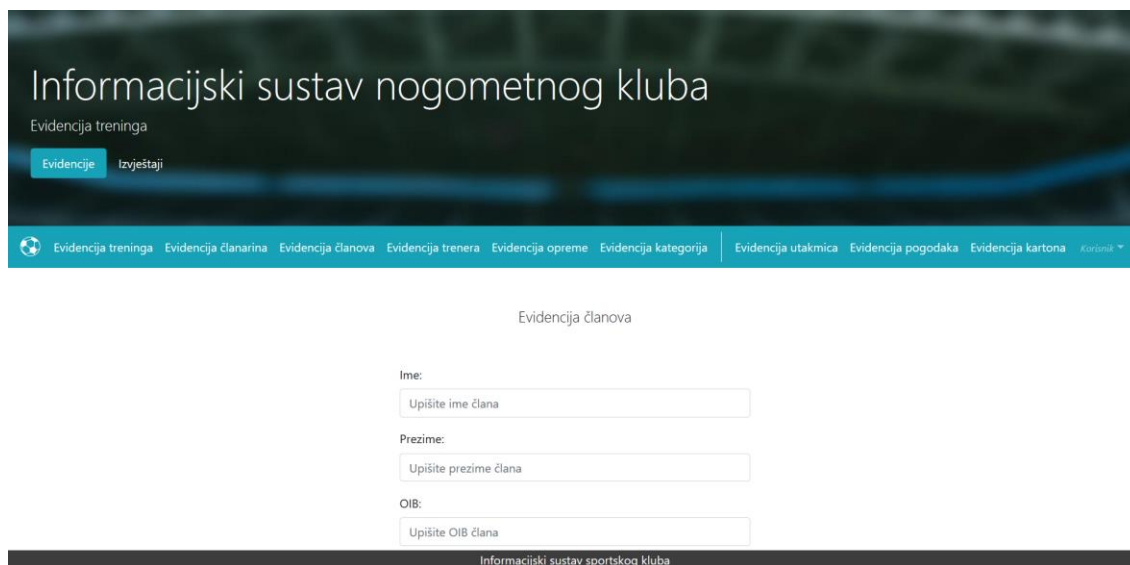
Glavna odlika Bootstrap frameworka je responzivni dizajn. Kako tvrde Shenoy i Sossou (*Learning Bootstrap, 2014: 7*), sa razvojem mobilnih uređaja i tableta, responzivni web dizajn je postao nužnost. Prije bi programeri, kada bi trebali prilagoditi nekakvu desktop aplikaciju mobilnom uređaju, uklanjali određene značajke koje mali ekrani ne bi bili u stanju podržati i tako prilagođavali aplikaciju situaciji.

Gasston(*The Mordern Web With HTML5, CSS3, and JavaScript, 2013: 41*) tvrdi: „Moje je mišljenje da je, kada se u obzir uzme brojnost različitih uređaja i zaslona, izrada niza stranica fiksne veličine kako bi se zadovoljile čak i samo najzastupljenije dimenzije zaslona, uzaludan posao. Bolji pristup, i ključ za reponzivan razvoj, je primjena fluidnog dizajna, što uključuje korištenje postotaka za vrijednosti dužine kako bi se dimenzije elemenata na stranici mijenjale sukladno veličini prostora za prikaz.“

Fluidan dizajn se odnosi na dizajn koji koristi postotke vrijednosti dužine kako bi prilagodio dimenzije elemenata na stranici dimenzijama ekrana. To je responzivan dizajn.

Dolaskom Bootstrapa 3, implementiran je mobile-first pristup, koji je pomogao kreiranju web stranica koje bi bile efektivne na svim mobilnim uređajima, neovisno o platformi.

Slika 7: Grafičko korisničko sučelje aplikacije na zaslonu prijenosnog računala



Izvor: autor

Slika 8: Grafičko korisničko sučelje aplikacije na zaslonu mobilnog uređaja



Izvor: autor

3. OPIS SUSTAVA

3.1. OPĆI OPIS

U sustavu se vrši evidencija podataka nogometnih klubova. U sustavu se odvijaju slijedeći procesi: evidencija trenera, evidencija članova, evidencija opreme, evidencija treninga, evidencija utakmica, evidencija kategorija, evidencija pogodaka, evidencija kartona, plaćanje članarine i izrada izvještaja.

Evidencija trenera se vrši na osnovu upisnih podataka koje trener dostavlja u sustav. U sustavu se evidentiraju slijedeći podatci o treneru: ime i prezime trenera, OIB trenera, broj trenerske iskaznice, email adresa i kontakt broj trenera.

Upis članova se vrši na osnovu osobnih podataka koje dostavlja član. U sustavu se evidentiraju slijedeći podatci o članu: broj članske iskaznice, ime i prezime člana, OIB člana, kategorija kojoj pripada, email adresa, kontakt broj člana i je li član dostavio rodni list.

Evidencija opreme se odvija tako da skladište dostavlja sustavu popis opreme u skladištu. U sustavu se evidentira sva klubska oprema koja sadrži naziv opreme i šifru svakog komada opreme.

Evidencija treninga se vrši tako da trener upisuje u sustav osnovne podatke o treningu, vlastitu trenersku iskaznicu, popis prisutnih članova i korištene opreme. U sustavu se evidentiraju slijedeći podatci o treningu: datum održavanja treninga, broj trenerske iskaznice, popis prisutnih članova i korištene opreme.

Evidencija utakmica se vrši tako da trener upisuje u sustav podatke o utakmici. Upisuju naziv domaće momčad, gostujuće momčadi, ime i prezime sudca, datum utakmice, i rezultat utakmice.

Evidenciju kartona vrši trener na utakmici. Trener upisuje vrijeme kartona, o kojem se kartonu radi, na kojoj je utakmici zarađen i koji ga je član zaradio.

Evidenciju pogodaka vrši trener na utakmici. Upisuje se vrijeme pogodka, radi li se o penalu, član koji je ostvario pogodak i utakmica na kojoj je pogodak ostvaren.

Evidencija kategorija. Trener upisuje kategorije igrača u sustav kluba. Upisuje se samo naziv kategorije (juniori, seniori...) i svaka kategorija ima pripadajuću šifru.

Evidencija članarina se odvija tako da tajnik kluba evidentira članarine. U sustavu se evidentira mjesec za koji se članarina plaća, je li članarina plaćena i član koji je platio članarinu.

U klubu se pišu izvještaji o upisanim članovima, upisanim trenerima, posjedovanju opreme i izvještaj treninga, upisanim utakmicama, kategorijama, pogodcima, kartonima i penalima. Predsjednik kluba daje zahtjev za izvještajem i rezultati izvještaja se dostavljaju predsjedniku kluba.

3.2. SPECIFIČNI PRIMJERI

a) Novi član se upisuje u klub. Član dostavlja osobne podatke potrebne za upis, a to su ime, prezime, OIB, rodni list, kontakt broj, email adresa. Članu se tada dodjeljuje kategorija i broj članske iskaznice. Član je nakon toga upisan.

b) Održava se trening. Trener bilježi datum treninga, upisuje broj vlastite iskaznice jer on vodi trening, zatim zapisuje sve članove koji sudjeluju na treningu i opremu koja će se koristiti na treningu.

c) Upisuje se nova članarina. Tajnik kluba bilježi za koji je mjesec članarina plaćena, koji ju član treba platiti i je li podmirena ili ne.

d) Upisuje se novi trener u klub. Trener dostavlja svoje osobne podatke, a to su: ime, prezime, OIB, kontakt broj i email adresa.

e) Klub je nabavio novu loptu. Upisuje se naziv lopte (npr. addidas crvena) i oprema se sprema u skladište.

f) U nogometnoj federaciji se uvodi nova kategorija (npr. veterani). Nova kategorija se upisuje u bazu kluba.

g) Klub sudjeluje na utakmici. Trener zapisuje sve potrebne podatke o utakmici: naziv domaće momčadi, naziv gostujuće momčadi, sudca koji sudi utakmicu, datum utakmice i konačno rezultat.

i) Na utakmici je ostvaren pogodak. Trener upisuje vrijeme pogodka, radi li se o penalu, na kojoj je utakmici ostvaren i koji ga je član ostvario.

j) Jedan od članova kluba je napravio prekršaj na utakmici i dobiva karton. Trener upisuje naziv kartona (žuti ili crveni), vrijeme kada je karton zarađen, na kojoj je utakmici zarađen i koji ga je član zaradio.

4. DIJAGRAM DEKOMPOZICIJE FUNKCIJA

Dijagram 1 - Dijagram dekompozicije funkcija



Izvor: autor

Prvi proces sustava je evidencija trenera. Svaki trener koji se upisuje u klub dostavlja upisne podatke. U sustavu se evidentiraju slijedeći podatci o treneru: ime i prezime trenera, OIB trenera, broj trenerske iskaznice, email adresa i kontakt broj trenera.

Drugi proces je evidencija članova. Član dostavlja podatke potrebne za upis. U sustavu se evidentiraju slijedeći podatci o članu: ime i prezime člana, OIB člana, jeli član dostavio rodni list, broj članske iskaznice, email adresa i kontakt broj člana i kategorija člana.

Slijedeći proces je evidencija kategorija. Trener upisuje moguće kategorije u sustav kluba. Upisuje se samo naziv i šifra kategorije kategorije (npr. 1 - juniori, 2 - seniori...).

Slijedeći proces je evidencija opreme. Taj proces se odvija tako da skladište dostavlja sustavu popis opreme u skladištu. U sustavu se evidentira sva klubska oprema koja ima svoj naziv i šifru.

Nakon toga slijedi evidencija treninga. Trener upisuje u sustav popis prisutnih članova, popis korištene opreme, datum održavanja treninga i broj vlastite iskaznice. U sustavu se evidentiraju slijedeći podatci o treningu: datum održavanja treninga, broj trenerske iskaznice, popis članova koji su sudjelovali i opreme koja je korištena na treningu.

Slijedeći proces je Evidencija članarina. Članarine evidentira tajnik. Bilježi se ime i prezime člana na kojeg se članarina obvezuje, za koji je mjesec i je li plaćena.

Evidencija utakmica se vrši tako da trener upisuje u sustav podatke o utakmici. Bilježe se naziv domaće momčad, gostujuće momčadi, ime i prezime sudca, datum utakmice i rezultat utakmice.

Evidenciju pogodaka vrši trener na utakmici. On upisuje osnovne podatke o pogotku. Upisuje se vrijeme svih dobivenih pogodaka, dali je pogodak bio penal, na kojoj je utakmici zabijen i ime i prezime člana koji je zabio pogodak.

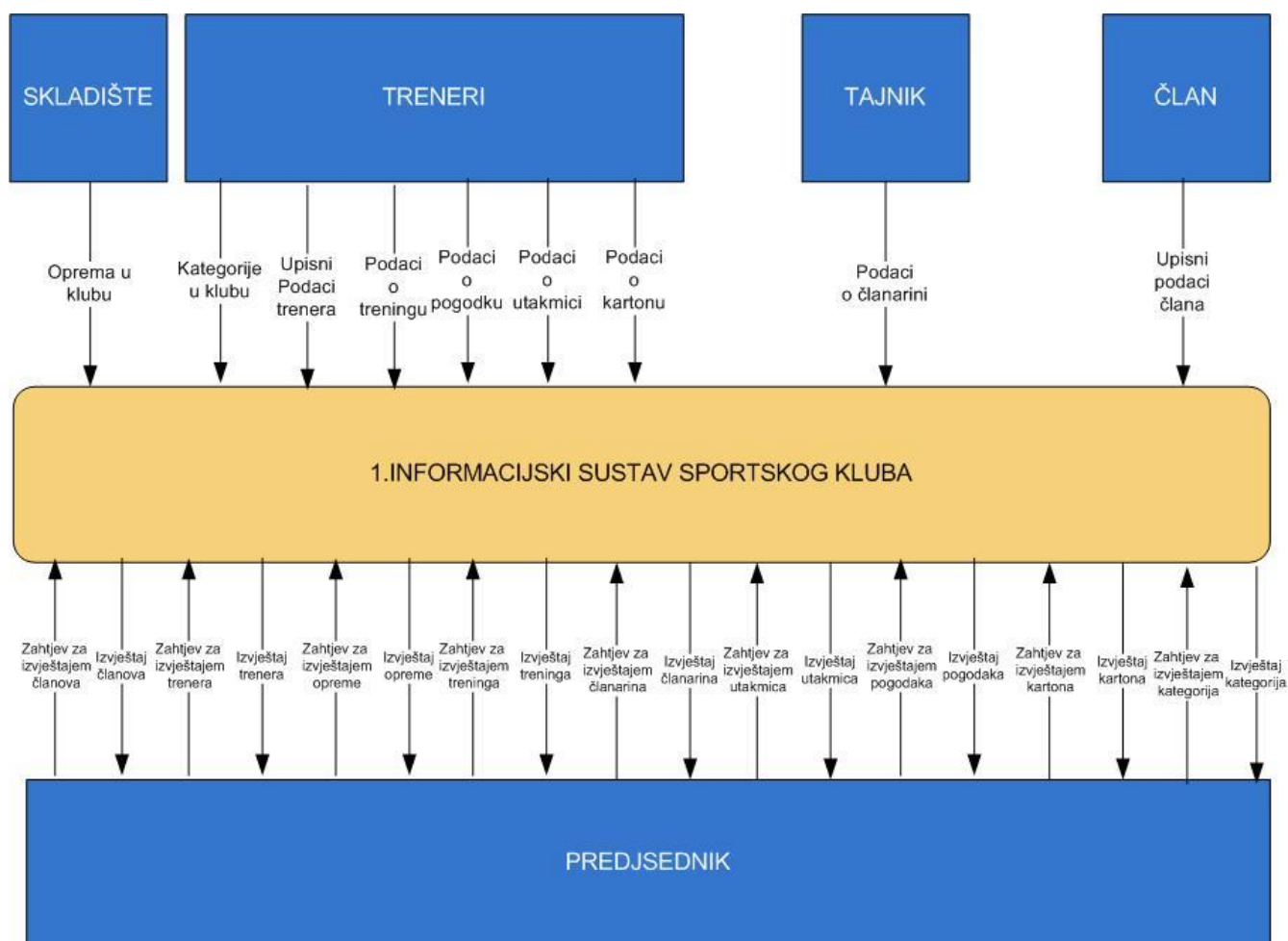
Evidenciju kartona vrši trener na utakmici. Upisuje naziv kartona, vrijeme dobivenog kartona, na kojoj je utakmici karton izdan i koji je član dobio karton.

Posljednji proces je izrada izvještaja. Sustav izrađuje izvještaje koji se traže od njega i dostavlja ih korisniku(predsjedniku kluba).

5. DIJAGRAM TOKA PODATAKA

5.1. DIJAGRAM TOKA PODATAKA NULTE RAZINE

Dijagram 2: Dijagram toka podataka nulte razine



Izvor: autor

Dijagram toka podataka nulte razine opisuje informacijski sustav evidencije podataka sportskog kluba u kojem se odvijaju pod procesi - evidencija trenera, evidencija članova, evidencija opreme, evidencija treninga, plaćanje članarine, evidencija utakmica, evidencija pogodaka, evidencija kartona, evidencije kategorija i izrada izvještaja.

Vanjski sustavi s kojima ovaj sustav komunicira su: trener, tajnik, predsjednik, član i skladište opreme. Svi procesi koji su opisani u dijagramu funkcionalne raščlambe se nalaze i u dijagramu toka podataka nulte razine. Sustav prima podatke iz vanjskih sustava, te sam isporučuje potrebne podatke.

Trener glavnom procesu isporučuje kategorije u klubu, upisne podatke trenera, podatke i treningu, podatke o pogotku, podatke o utakmici podatke o kartonu.

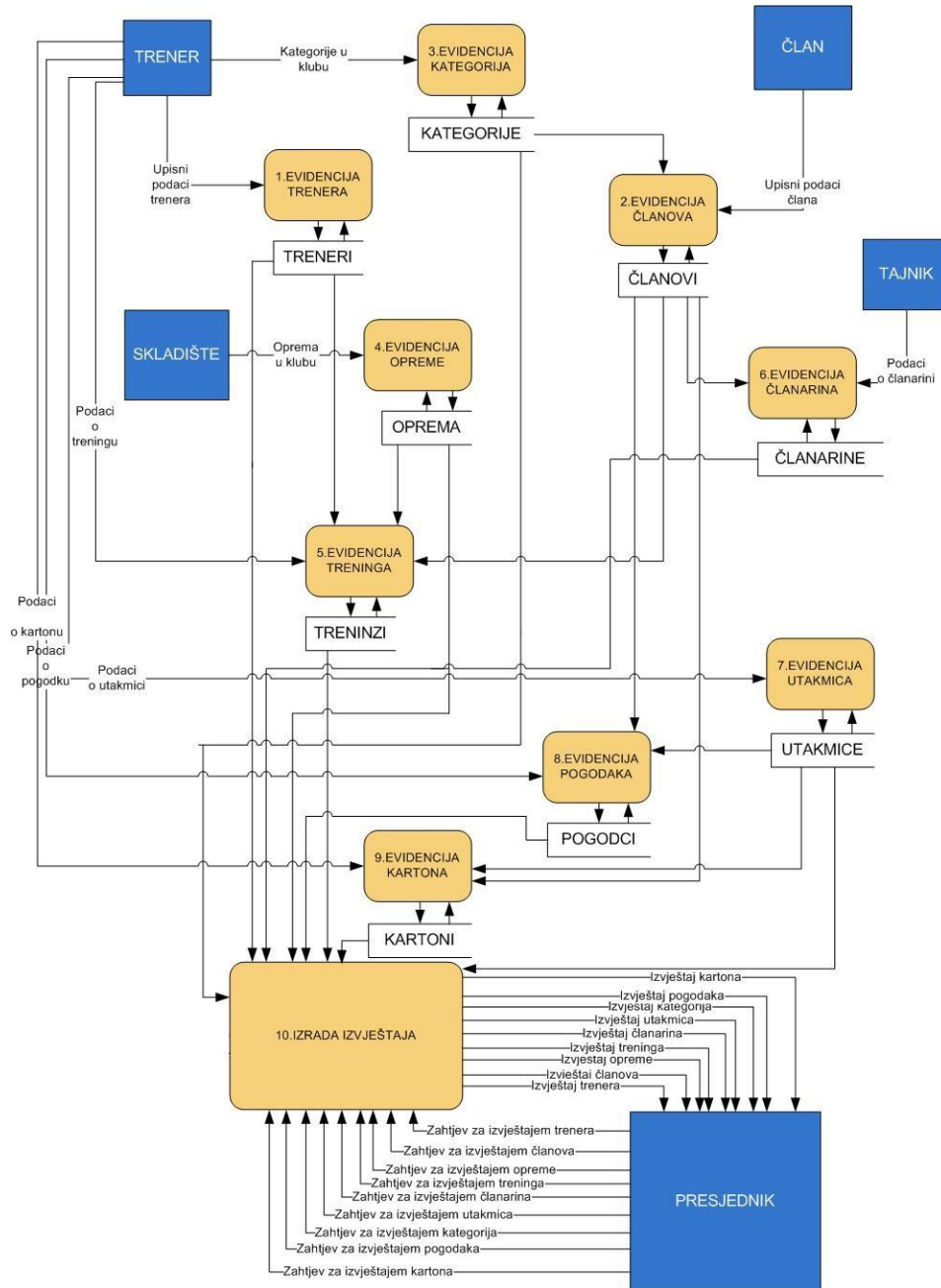
Član glavnom procesu dostavlja upisne podatke člana.

Skladište u glavni proces dostavlja popis opreme u klubu.

Predsjednik glavnom procesu šalje zahtjeve za izradom izvještaja o članovima, trenerima, opremom, treninzima, članarinama, utakmicama, pogotcima, kartonima i kategorijama, a na to iz glavnog procesa dobiva odgovarajuće izvještaje.

5.2. DIJAGRAM TOKA PODATAKA PRVE RAZINE

Dijagram 3: Dijagram toka podataka prve razine



Izvor: autor

Prvi proces opisan u dijagramu toka podataka prve razine je evidencija trenera. Trener procesu dostavlja upisne podatke koji se bilježe u skladištu treneri.

Drugi proces je evidencija članova. Odvija se tako da član procesu dostavlja upisne podatke člana. Podatci se čuvaju u skladištu podataka članovi.

Evidencija kategorija se odvija tako da trener upisuje kategorije u proces što se bilježi u skladištu podataka kategorije.

Slijedeći proces je evidencija opreme. Skladište opreme u sustav dostavlja popis opreme u proces i podatci o opremi se čuvaju u skladištu podataka oprema.

Nakon toga dolazi evidencija treninga. Trener dostavlja u proces podatke o treningu. Ovaj proces je također povezan sa trima skladištima podataka treneri, članovi i oprema. Sve primljene podatke proces evidencije treninga skladišti u skladištu podataka treninzi. Nije moguće upisati veći datum treninga od trenutnog.

Plaćanje članarina se odvija tako da tajnik upisuje podatke o svakoj članarini. Proces plaćanja članarina se također uzima podatke iz skladišta podataka članovi i sve dobivene podatke bilježi u skladištu podataka članarine.

Nakon toga dolazi evidencija utakmica. Proces se odvija tako da trener upisuje podatke o utakmici u sustav i svi podaci se bilježe u skladište podataka utakmice. Nije moguće upisati datum utakmice veći od trenutnog datuma.

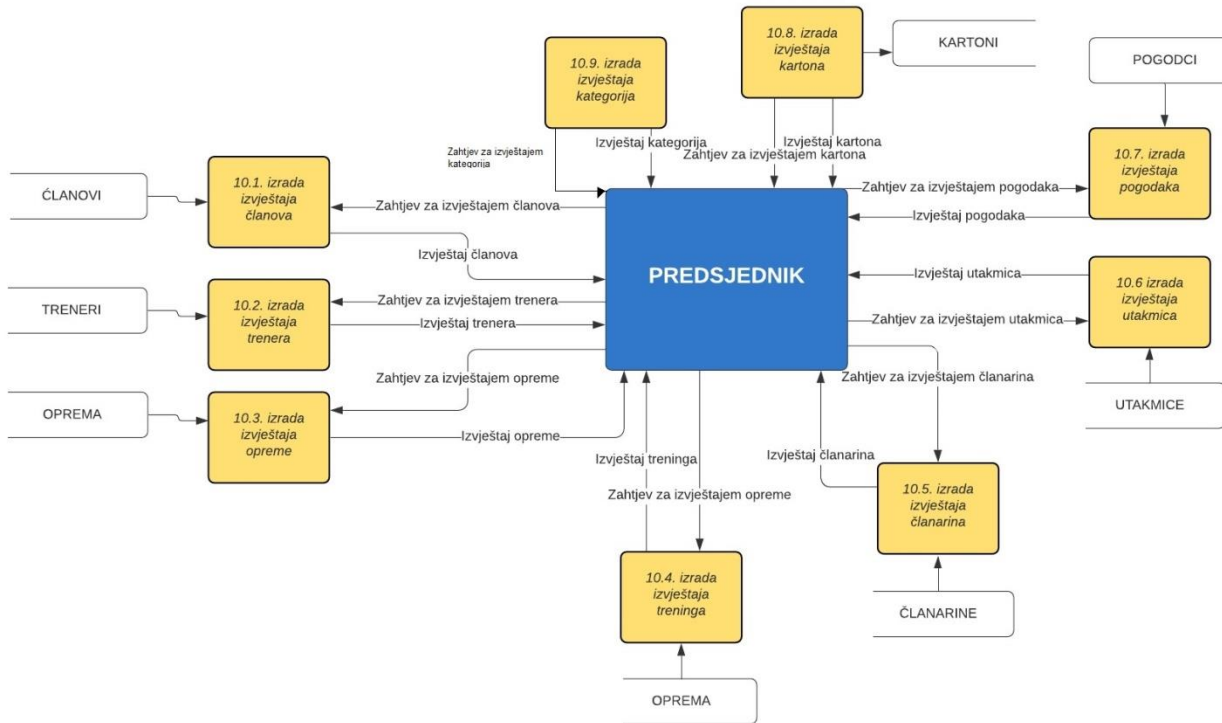
Evidencija pogodaka se izvršava tako da trener u proces upisuje podatke o pogotku. Proces također uzima podatke iz skladišta podataka članovi i utakmice. Svi podaci o evidenciji pogodaka se upisuju u skladište podataka pogodci.

Evidencija kartona se izvršava tako da trener u proces upisuje podatke o kartonu. Proces također uzima podatke iz skladišta podataka članovi i utakmice. Svi podaci o evidenciji kartona se upisuju u skladište podataka kartoni.

Izrada izvještaja komunicira s predsjednikom. Predsjednik procesu šalje zahtjeve za izradom izvještaja trenera, članova, opreme, treninga, članarina, utakmica, kategorije, pogodaka i kartona i na to iz glavnog procesa dobiva izvještaj trenera, članova, opreme, treninga, članarina, utakmica, kategorija, pogodaka i kartona.

5.3. DIJAGRAM TOKA PODATAKA DRUGE RAZINE

Dijagram 4: Dijagram toka podataka druge razine



Izvor: autor

Proces izrade izvještaja se raščlanjuje na devet pod procesa: izrada izvještaja članova, izrada izvještaja trenera, izrada izvještaja opreme, izrada izvještaja treninga, izrada izvještaja članarina, izrada izvještaja utakmica, izrada izvještaja pogodaka, izrada izvještaja kartona, izrada izvještaja kategorija.

Izrada izvještaja članova se odvija tako da predsjednik šalje u proces zahtjev za izvještajem članova. Proces dohvaća podatke iz skladišta podataka članovi i dostavlja predsjedniku izvještaj članova.

Izrada izvještaja trenera se odvija tako da predsjednik šalje u proces zahtjev izvještajem trenera. Proces dohvaća podatke iz skladišta podataka treneri i dostavlja predsjedniku izvještaj o trenerima.

Izrada izvještaja o opreme se odvija tako da predsjednik šalje u proces zahtjev za izvještajem opreme. Proces dohvaća podatke iz skladišta podataka oprema i dostavlja predsjedniku izvještaj opreme.

Izvještaji treninga se odvijaju tako da predsjednik šalje u proces zahtjev za izvještajem treninga. Proces dohvaća podatke iz skladišta podataka treninzi i dostavlja predsjedniku izvještaj treninga.

Izrada izvještaja članarina se odvija tako da predsjednik šalje u proces zahtjev za izvještajem članarina. Proces dohvaća podatke iz skladišta podataka članarine i dostavlja predsjedniku izvještaj članarina.

Izvještaji utakmica se odvijaju tako da predsjednik šalje u proces zahtjev za izvještajem utakmica. Proces dohvaća podatke iz skladišta podataka Utamice i dostavlja predsjedniku izvještaj utakmica.

Izrada izvještaja pogodaka se odvija tako da predsjednik šalje u proces zahtjev za izvještajem pogodaka. Proces dohvaća podatke iz skladišta podataka pogodci i dostavlja predsjedniku izvještaj pogodaka.

Izrada izvještaja kartona se odvija tako da predsjednik šalje u proces zahtjev za izvještajem kartona. Proces dohvaća podatke iz skladišta podataka kartoni i dostavlja predsjedniku izvještaj kartona.

Izrada izvještaja kategorija se odvija tako da predsjednik šalje u proces zahtjev za izvještajem kategorija. Proces dohvaća podatke iz skladišta podataka kategorije i dostavlja predsjedniku izvještaj kategorija.

6. OPIS TOKOVA I SKLADIŠTA PODATAKA

6.1. PROCES - EVIDENCIJA TRENERA

Tablica 1: Tok podataka - Upisni podaci trenera

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Upisni podaci trenera	PK-> Br_trenerske_iskaznice(Samoneriranjje)	Podaci koje trener donosi u klub pri upisu	32	Obavezno
	Ime_trenera(Tekst(30))		Ivan	Obavezno
	Prezime_trenera(Tekst(30))		Ivić	Obavezno
	OIB_trenera(Tekst(11))		13453678987	Obavezno
	Kontakt_broj_trenera(Tekst(20))		092 254 459	Nije obavezno
	E_mail_trenera(Tekst(35))		Ivan.Ivic@Gmail.com	Nije obavezno/mora biti mail

Izvor: autor

Tablica 2: Skladište podataka Treneri

Naziv Skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
TRENERI	PK-> Br_trenerske_iskaznice(Samoneriranjje)	Skladište u kojem se bilježe podaci o trenerima	34	Obavezno
	Ime_trenera(Tekst(30))		Ivan	Obavezno
	Prezime_trenera(Tekst(30))		Ivić	Obavezno
	OIB_trenera(Tekst(11))		13453678987	Obavezno
	Kontakt_broj_trenera(Tekst(20))		092 254 459	Nije obavezno
	E_mail_trenera(Tekst(35))		Ivan.Ivic@Gmail.com	Nije obavezno/mora biti mail

Izvor: autor

6.2. PROCES - EVIDENCIJA ČLANOVA

Tablica 3: Tok podataka - Upisni podaci člana

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Upisni podaci člana	PK-> Br_clanske_iskaznice(Samonumeriranje)	Podaci koje član donosi u klub pri upisu	13	Obavezno
	Ime_clana(Tekst(30))		Ivan	Obavezno
	Prezime_clana(Tekst(30))		Ivić	Obavezno
	OIB_clana(Tekst(11))		13453678987	Obavezno
	Kontakt_br_clana(tekst(20))		092 254 459	Nije obavezno
	E_mail_clana(tekst(30))		Ivan.Ivic@Gmail.com	Nije obavezno/mora biti mail
	Posjedovanje_rodnog_lista(Da/Ne)		Da	Nije obavezno
	FK-> Sifra_Kategorije(broj(11))		32	Obavezno

Izvor: autor

Tablica 4: Skladište podataka - Članovi

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
ČLANOVI	PK-> Br_clanske_iskaznice(Samonumeriranje)	Skladište u kojem se bilježe svi podaci o članovima	34	Obavezno
	Ime_clana(Tekst(30))		Ivan	Obavezno
	Prezime_clana(Tekst(30))		Ivić	Obavezno
	OIB_clana(Tekst(11))		13453678987	Obavezno
	Kontakt_br_clana(tekst(20))		092 254 459	Nije obavezno
	E_mail_clana(tekst(30))		Ivan.Ivic@Gmail.com	Nije obavezno/mora biti mail
	Posjedovanje_rodnog_lista(Da/Ne)		Da	Obavezno
	FK-> Sifra_Kategorije(broj(11))		43	Obavezno

Izvor: autor

6.3 PROCES - EVIDENCIJA KATEGORIJA

Tablica 5: Tok podataka - Kategorije u klubu

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Kategorije u klubu	PK-> Sifra_Kategorije(Samonumeriranje)	Kategorije koje postoje u klubu	34	Obavezno
	Naziv_kategorije(Tekst(30))		Kadeti	Obavezno

Izvor: autor

Tablica 6: Skladište podataka - Kategorije

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
KATEGORIJE	PK-> Sifra_Kategorije(Samonumeriranje)	Skladište u kojem se upisuju svi podaci o kategorijama	34	Obavezno
	Naziv_kategorije(Tekst(30))		Kadeti	Obavezno

Izvor: autor

6.4. PROCES - EVIDENCIJA OPREME

Tablica 7: Tok podataka - Oprema u klubu

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Oprema u klubu	PK-> Sifra_opreme(Samonumeriranje)	Dio opreme koju klub posjeduje	34	Obavezno
	Naziv_opreme(Tekst(30))		Lopta	Obavezno

Izvor: autor

Tablica 8: Skladište podataka - Oprema

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
OPREMA	PK-> Sifra_opreme(Samonumeriranje)	Skladište u kojem se upisuju svi podaci o opremi	34	Obavezno
	Naziv_opreme(Tekst(30))		Lopta	Obavezno

Izvor: autor

6.5. PROCES - EVIDENCIJA TRENINGA

Tablica 9: Tok podataka - Podaci o treningu

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Podaci o treningu	PK-> Sifra_treninga(Samonumeriranje)	Podaci o održanim treninzima	34	Obavezno
	Datum_treninga(Datum)		21.11.2005.	Obavezno
	FK-> Br_trenerske_iskaznice(Broj(11))		45	Nije obavezno
	FK-> Br_clanske_iskaznice(Broj(11))		35	Obavezno
	FK-> Sifra_opreme(Broj(11))		21	Nije obavezno

Izvor: autor

Tablica 10: Skladište podataka - Treninzi

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
TRENINZI	PK-> Sifra_treninga(Samonumeriranje)	Skladište u kojem se bilježe svi podaci o treninzima	34	Obavezno
	Datum_treninga(Datum)		21.11.2005.	Obavezno
	FK-> Br_trenerske_iskaznice(Broj(11))		45	Nije obavezno
	FK-> Br_clanske_iskaznice(Broj(11))		35	Obavezno
	FK-> Sifra_opreme(Broj(11))		21	Nije obavezno

Izvor: autor

6.6. PROCES - EVIDENCIJA ČLANARINA

Tablica 11: Tok podataka - Podaci o članarini

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Podaci o članarini	PK-> Sifra_clanarine(Samoneriranje)	Članarine koje trebaju platiti članovi	34	Obavezno
	Za_mjesec(tekst(2))		05	Obavezno
	Placeno(Da/Ne)		Da	Obavezno
	FK-> Br_clanske_iskaznice(broj(11))		21	Obavezno

Izvor: autor

Tablica 12: Skladište podataka - Članarine

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
ČLARINE	PK-> Sifra_clanarine(Samoneriranje)	Skladište u kojem se bilježe svi podaci o članarinama	34	Obavezno
	Za_mjesec(tekst(2))		05	Obavezno
	Placeno(Da/Ne)		Da	Obavezno
	FK-> Br_clanske_iskaznice(broj(11))		21	Obavezno

Izvor: autor

6.7 PROCES - EVIDENCIJA UTAKMICA

Tablica 13: Tok podataka - Podaci o utakmici

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Podaci o utakmici	PK-> Sifra_utakmice(Samonumeriranje)	Utakmice na kojima je klub sudjelovao	32	Obavezno
	Domaca_momcad(Tekst(30))		NK Grobnik	Obavezno
	Gostujuca_momcad(Tekst(30))		NK Trnava	Obavezno
	Sudac_utakmice(Tekst(30))		Josip Josić	Nije obavezno
	Datum_utakmice(Datum)		21.12.2013.	Obavezno
	Rezultat_utakmice(Tekst(5))		02:01	Obavezno

Izvor: autor

Tablica 14: Skladište podataka - Utakmice

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
UTAKMICE	PK-> Sifra_utakmice(Samonumeriranje)	Skladište u kojem se bilježe svi podaci o utakmici	32	Obavezno
	Domaca_momcad(Tekst(30))		NK Grobnik	Obavezno
	Gostujuca_momcad(Tekst(30))		NK Trnava	Obavezno
	Sudac_utakmice(Tekst(30))		Josip Josić	Nije obavezno
	Datum_utakmice(Datum)		21.12.2013.	Obavezno
	Rezultat_utakmice(Tekst(5))		02:01	Obavezno

Izvor: autor

6.8. PROCES - EVIDENCIJA POGODAKA

Tablica 15: Tok podataka - Podaci o pogodku

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Podaci o pogodku	PK-> Sifra_pogodka(Samonumeriranje)	Pogodak kojeg je određeni član zabio na određenoj utakmici	32	Obavezno
	Vrijeme_pogodka(Tekst(5))		21:30	Obavezno
	Penal(Da/Ne)		Da	Obavezno
	FK-> Br_clanske_iskaznicie(Broj(11))		32	Obavezno
	FK-> Sifra_utakmcie(Broj(11))		11	Obavezno

Izvor: autor

Tablica 16: Skladište podataka - Pogodci

Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
POGODCI	PK-> Sifra_pogodka(Samonumeriranje)	Skladište u kojem se bilježe svi podaci o utakmicama	32	Obavezno
	Vrijeme_pogodka(Tekst(5))		21:30	Obavezno
	Penal(Da/Ne)		Da	Obavezno
	FK-> Br_clanske_iskaznicie(Broj(11))		32	Obavezno
	FK-> Sifra_utakmcie(Broj(11))		11	Obavezno

Izvor: autor

6.9 PROCES - EVIDENCIJA KARTONA

Tablica 17: Tok podataka - Podaci o kartonu

Naziv toka	Tip i duljina podataka	Opis toka	Primjer	Kontrola unosa
Podaci o kartonu	PK-> Sifra_kartona(Samonumeriranje)	Karton koji je odeđeni član zaradio na određenoj utakmici	32	Obavezno
	Vrijeme_kartona(Tekst(5))		21:30	Obavezno
	Naziv_kartona(Tekst(1))		Ž	Obavezno
	FK-> Br_clanske_iskazncie(Broj(11))		32	Obavezno
	FK-> Sifra_utakmcie(Broj(11))		11	Obavezno

Izvor: autor

Tablica 18: Skladište podataka - Kartoni

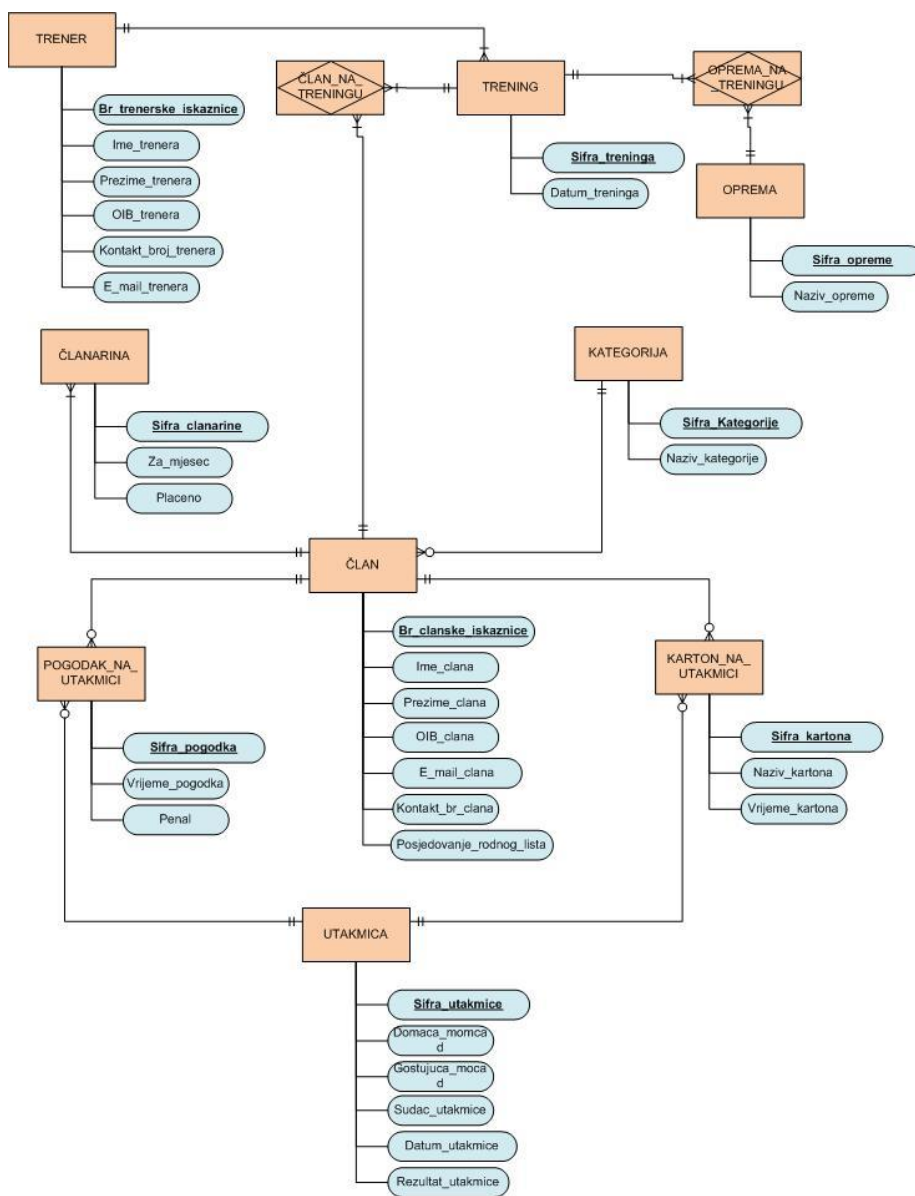
Naziv skladišta	Tip i duljina podataka	Opis skladišta	Primjer	Kontrola unosa
KARTONI	PK-> Sifra_kartona(Samonumeriranje)	Skladište u kojem se bilježe svi podaci o kartonima	32	Obavezno
	Vrijeme_kartona(Tekst(5))		21:30	Obavezno
	Naziv_kartona(Tekst(1))		Ž	Obavezno
	FK-> Br_clanske_iskazncie(Broj(11))		32	Obavezno
	FK-> Sifra_utakmcie(Broj(11))		11	Obavezno

Izvor: autor

7. MODEL PODATAKA - EVA MODEL

7.1. EVA MODEL

Dijagram 5: EVA model



Izvor: autor

7.2. OPIS TIPOVA ENTITETA

Tablica 19: Popis tipova entiteta

Tipovi entiteta	Opis
TRENER	Trener koji vodi treninge u klubu. Njegovi podaci su broj trenerske iskaznice, ime, prezime, OIB, kontakt broj i mail adresa.
ČLAN	Član koji trenira u klubu. Njegovi podaci su broj članske iskaznice, ime, prezime, OIB, kontakt broj, email adresa i posjedovanje rodnog lista.
OPREMA	Oprema koju klub posjeduje. Svaki dio opreme ima svoju šifru i naziv.
TRENING	Treninzi koji se održavaju u klubu. Podaci ovog tipa entiteta su šifra treninga i datum treninga.
ČLAN_NA_TRENINGU	Agregacija koja nastaje iz veze treninga i člana. Označava koji član je bio na kojem treningu.
OPREMA_NA_TRENINGU	Agregacija koja nastaje iz veze treninga i opreme. Označava koja je oprema korištena na kojem treningu.
KATEGORIJA	Označava dobnu kategoriju članova u klubu. Svaka kategorija ima šifru i naziv.
ČLANARINA	Članarine koje plaćaju članovi. Njoj pripadaju podaci šifra članarine, za koji se mjesec plaća i je li plaćena.
UTAKMICA	Utakmice na kojima klub sudjeluje. Podaci ovog tipa entiteta su šifra utakmice, naziv domaće momčadi, naziv gostujuće momčadi, ime i prezime sudca i rezultat utakmice.
POGODAK_NA_UTAKMICI	Pogodak koji određeni član zabija na određenoj utakmici. Sadrži podatke: šifra pogodka, vrijeme pogodka i radi li se o penalu.
KARTON_NA_UTAKMICI	Karton koji je određeni član zaradio na određenoj utakmici. Sadrži podatke: naziv kartona i vrijeme kartona.

Izvor: autor

7.3. OPIS ATRIBUTA TIPOVA ENTIETA

Tablica 20: Atributi tipa entiteta TRENER

Atributi tipa entiteta TRENER	Opis
Br_trenerske_iskaznice	Jedinstveni identifikator svakog trenera u klubu
Ime_trenera	Ovaj atribud obuhvaća uvid u osobno ime svakog trenera
Prezime_trenera	Ovaj atribud obuhvaća uvid u prezime svakog trenera
OIB_trenera	Jedinstveni identifikacijski broj svakog trenera
Kontakt_broj_trenera	Broj mobilnog ili fiksnog telefona trenera
E_mail_trenera	Ovaj atribut obuhvaća uvid u E-mail adresu svakog trenera

Izvor: autor

Tablica 21: Atributi tipa entiteta TRENING

Atributi tipa entiteta TRENING	Opis
Sifra_treninga	Jedinstveni identifikator svakog treninga održanog u klubu
Datum_treninga	Datum kada je održan pojedini trening

Izvor: autor

Tablica 22: Atributi tipa entiteta OPREMA

Atributi tipa entiteta OPREMA	Opis
Sifra_opreme	Jedinstveni identifikator svakog dijela opreme kojeg klub posjeduje
Naziv_opreme	Naziv svakog dijela opreme kojeg klub posjeduje

Izvor: autor

Tablica 23: Atributi tipa entiteta ČLAN

Atributi tipa entiteta ČLAN	Opis
Br_clanske_iskaznice	Jedinstveni identifikator svakog trenera u klubu
Ime_clana	Ovaj atribud obuhvaća uvid u osobno ime svakog upisanog člana
Prezime_clana	Ovaj atribud obuhvaća uvid u prezime svakog upisanog člana
OIB_clana	Jedinstveni identifikacijski broj svakog upisanog člana
Kontakt_br_clana	Broj mobilnog ili fiksnog telefona člana
E_mail_clana	Ovaj atribut obuhvaća uvid u E-mail adresu svakog upisanog člana
Posjedovanje_rodnog_lista	Podatak o tome je li član dostavio rodni list prilikom upisa

Izvor: autor

Tablica 24: Atributi tipa entiteta KATEGORIJA

Atributi tipa entiteta KATEGORIJA	Opis
Sifra_kategorije	Jedinstveni identifikator svake kategorije u klubu
Naziv_kategorije	Naziv kategorije (npr. Junior, senior...)

Izvor: autor

Tablica 25: Atributi tipa entiteta ČLANARINA

Atributi tipa entiteta ČLANARINA	Opis
Sifra_clanarine	Jedinstveni identifikator svake izdane članarine
Za_mjesec	Za koji mjesec je članarina izdana
Placeno	Je li članarina podmirena

Izvor: autor

Tablica 26: Atributi tipa entiteta POGODAK_NA_UTAKMICI

Atributi tipa entiteta POGODAK_NA_UTAKMICI	Opis
Sifra_pogodka	Jedinstveni identifikator svakog ostvarenog pogodka
Vrijeme_pogodka	Minuta i sekunda utakmice u kojoj je pogodak zabijen
Penal	Radi li se o penalu ili običnom pogodku

Izvor: autor

Tablica 27: Atributi tipa entiteta KARTON_NA_UTAKMICI

Atributi tipa entiteta KARTON_NA_UTAKMICI	Opis
Sifra_kartona	Jedinstveni identifikator svakog ostvarenog kartona
Vrijeme_kartona	Minuta i sekunda utakmice u kojoj je karton zarađen
Naziv_kartona	Naziv kartona (Crveni ili žuti)

Izvor: autor

Tablica 28: Atributi tipa entiteta UTAKMICA

Atributi tipa entiteta UTAKMICA	Opis
Sifra_utakmice	Jedinstveni identifikator svake odigrane utakmice
Domaca_momcad	Naziv domaće momčadi
Gostujuca_momcad	Naziv godstujuće momčadi
Sudac_utakmice	Ime i prezime sudca koji sudi utakmicu
Datum_utakmice	Dan kada se utakmica održala
Rezultat_utakmice	Rezultat utakmice (npr. 01:01)

Izvor: autor

7.4. OPIS VEZA EVA MODELA

Tablica 29: Veze EVA modela

Naziv veze	Opis
TRENER - TRENING	Jedan trener može voditi jedan ili više treninga, a jedan ili više treninga može biti vođeno od strane jednog trenera.
TRENING - ČLAN	Jedan član može prisustvovati na više treninga, a više treninga može biti prisustvovano od više članova, tako da nastaje agregacija ČLAN_NA_TRENINGU
TRENING - OPREMA	Jedna oprema može biti korištena na više treninga, a na više treninga se može koristiti od više opreme, tako da nastaje agregacija OPREMA_NA_TRENINGU
ČLAN - ČLANARINA	Jedna ili više članarina, može dužiti jednog člana, a jedan član može biti zadužen pod jednu ili više članarina
ČLAN - KATEGORIJA	Jedan ili nula članova mogu imati jednu kategoriju, a jedna kategorija može biti posjedovana od nula ili više članova
ČLAN - POGODAK_NA_UTAKMICI	Jedan član može ostvariti nula ili više pogodaka na jednoj utakmici, a nula ili više pogodaka može biti ostvareno od jednog člana
UTAKMICA - POGODAK_NA_UTAKMICI	Nula ili više pogodaka može biti ostvareno na jednoj utakmici, a na jednoj utakmici može biti ostvareno nula ili više pogodaka
ČLAN - KARTON_NA_UTAKMICI	Jedan član može zaraditi nula ili više kartona na jednoj utakmici, a nula ili više kartona na jednoj utakmici može biti zarađeno od strane jednog člana.
UTAKMICA - KARTON_NA_UTAKMICI	Na jednoj utakmici može biti ostvareno nula ili više kartona, a nula ili više kartona može biti ostvareno na jednoj utakmici.

Izvor: autor

8. RELACIJSKI MODEL PODATAKA

8.1. POPIS RELACIJSKIH SHEMA

TRENER(**Br trenerske iskaznice(PK)**, Ime_trenera, Prezime_trenera, OIB_trenera, Kontakt_broj_trenera, E_mail_trenera),

TRENING(**Sifra treninga(PK)**, Datum_treninga, Br_trenerske_iskaznice(FK)),

OPREMA_NA_TRENINGU(**Sifra treninga(PK,FK)**, **Sifra opreme(PK,FK)**),

OPREMA(**Sifra opreme**, Naziv_opreme),

ČLAN_NA_TRENINGU(**Sifra treninga(PK,FK)**, **Br clanske iskaznice(PK,FK)**),

ČLAN(**Br clanske iskaznice(PK)**, Ime_clana, Prezime_clana, OIB_clana, E_mail_clana, Kontakt_broj_clana, Posjedovanje_rodnog_lista, **Sifra kategorije(FK)**),

KARTON_NA_UTAKMICI(**Sifra kartona(PK)**, Naziv_kartona, Vrijeme_kartona, **Br clanske iskaznice(FK1)**, **Sifra utakmice(FK2)**),

POGODAK_NA_UTAKMICI(**Sifra pogodka(PK)**, Vrijeme_pogodka, Penal, **Br clanske iskaznice(FK1)**, **Br clanske iskaznice(FK2)**),

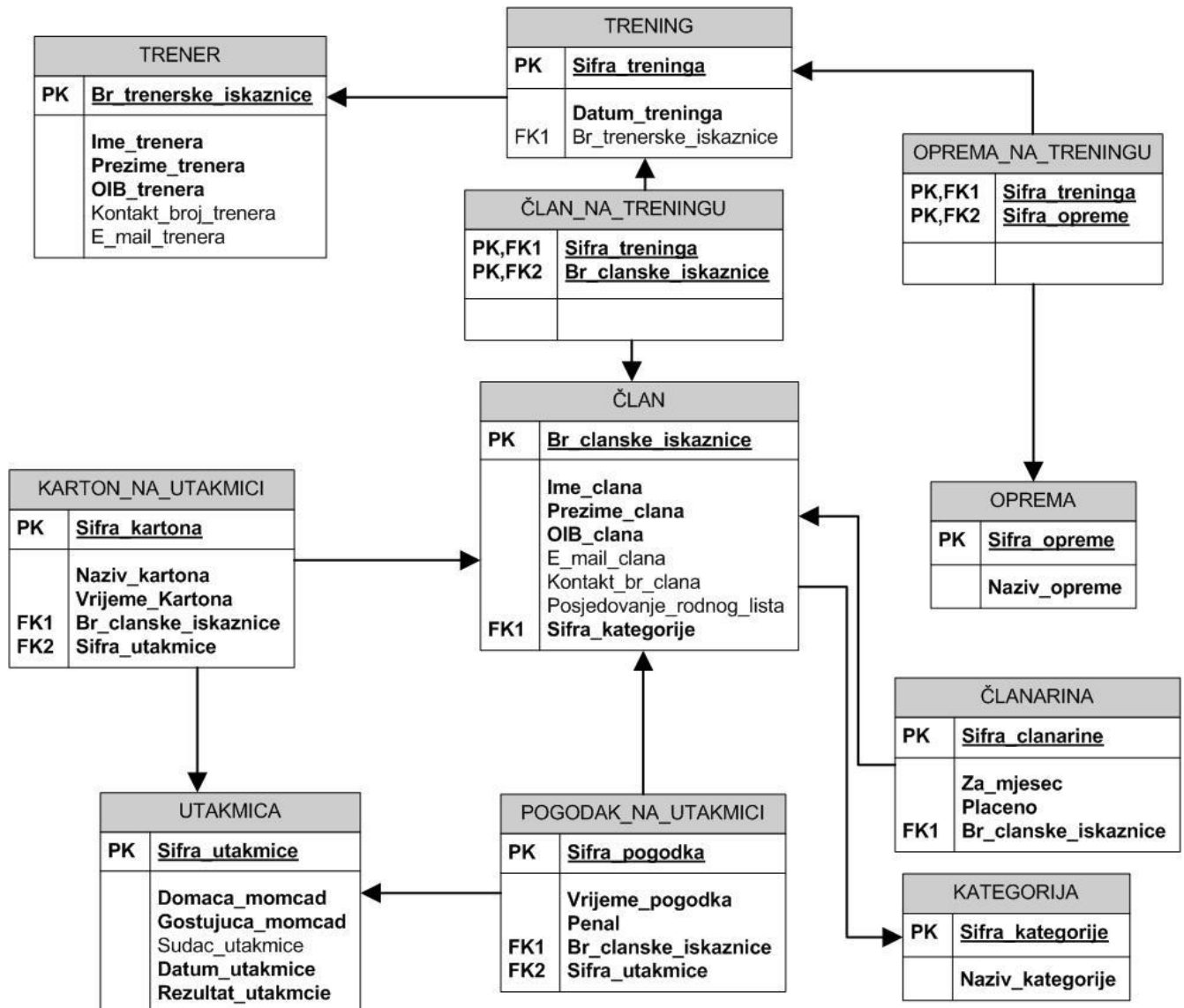
UTAKMICA(Sifra utakmice(PK), Domaca_momcad, Gostujuca_momcad,
Sudac_utakmice, Datum_utakmice, Rezultat_utakmice),

ČLANARINA(Sifra clanarine(PK), Za_mjesec, Placeno, Br_clanske_iskaznice),

KATEGORIJA(Sifra kategorije(PK), Naziv_kategorije)

8.2. RELACIJSKI MODEL

Dijagram 6: Relacijski model

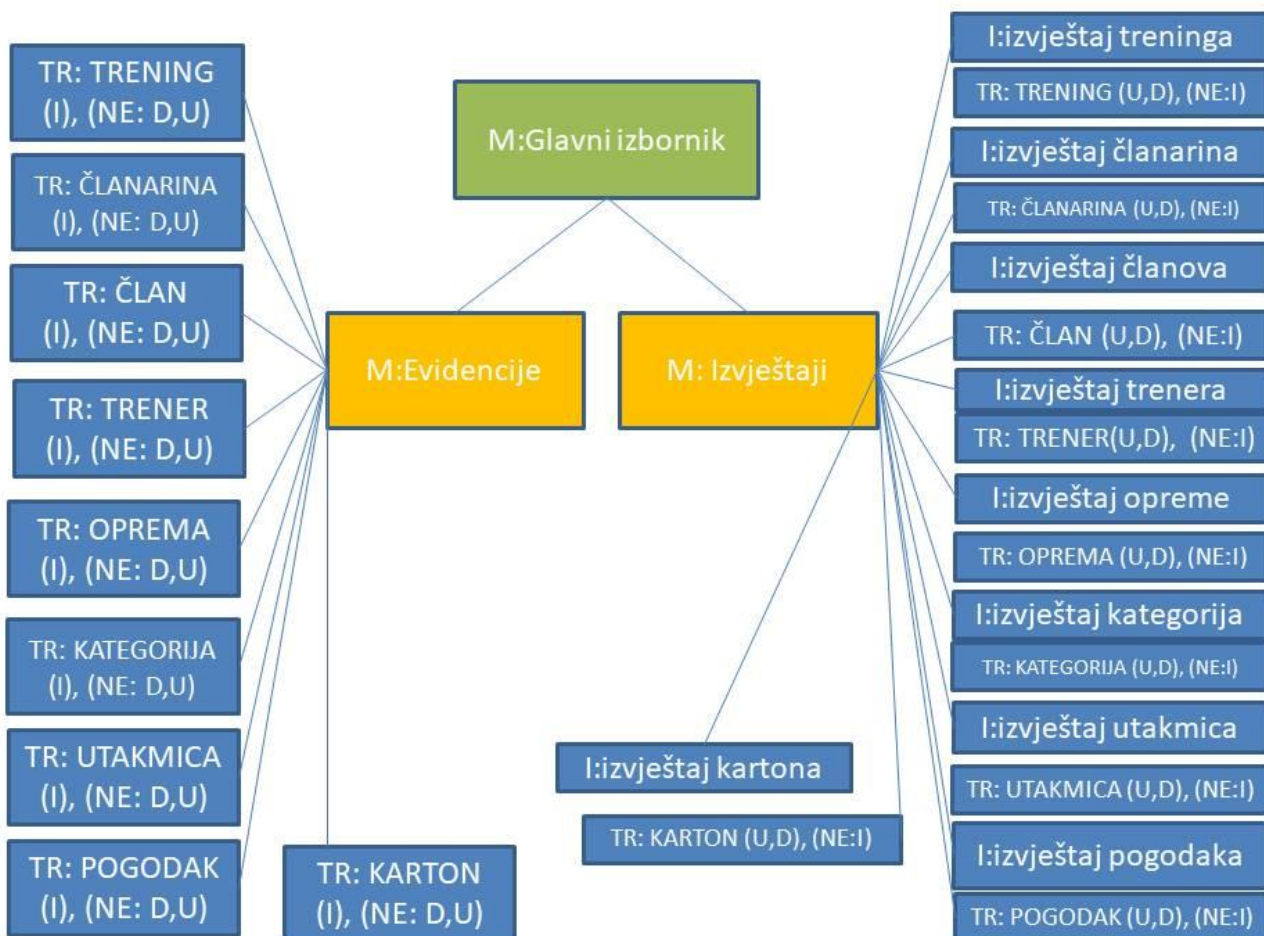


Izvor: autor

9. ARHITEKTURA PROGRAMSKE PODRŠKE

9.1. DIJAGRAM PROGRAMSKE PODRŠKE (DPP)

Dijagram 7: DPP dijagram



Izvor: autor

9.2. TEKSTUALNI OPIS DPP DIJAGRAMA

TR: TRENING - modul prikazuje pojedinačno pojavljivanje tipa entiteta TRENING, nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: ČLANARINA - modul prikazuje pojedinačno pojavljivanje tipa entiteta ČLANARINA, nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: ČLAN - modul prikazuje pojedinačno pojavljivanje tipa entiteta ČLAN nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: TRENER - modul prikazuje pojedinačno pojavljivanje tipa entiteta TRENER nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: OPREMA - modul prikazuje pojedinačno pojavljivanje tipa entiteta OPREMA nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: KATEGORIJA - modul prikazuje pojedinačno pojavljivanje tipa entiteta KATEGORIJA nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: UTAKMICA - modul prikazuje pojedinačno pojavljivanje tipa entiteta UTAKMICA nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: POGODAK - modul prikazuje pojedinačno pojavljivanje tipa entiteta POGODAK nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

TR: KARTON - modul prikazuje pojedinačno pojavljivanje tipa entiteta KARTON nad kojem se izvršava operacija unosa. Operacije brisanja i uređivanja nisu omogućene.

10. IMPLEMENTACIJA POSLOVNE LOGIKE

Prilikom upisa svake utakmice, potrebno je unijeti datum kada je utakmica bila održana. Ako je datum veći od trenutnog datuma, onemogućuje se upis i korisniku se prikazuje poruka koja ga obavještava kako je upisao pogrešan datum. Pseudokod:

```
IF (uneseni datum <= trenutni datum)
    THEN Upiši utakmicu;
    ELSE Ispiši poruku upozorenja;
ENDIF;
```

Prilikom upisa svakog treninga, potrebno je unijeti datum kada je trening bio održan. Ako je datum veći od trenutnog datuma, onemogućuje se upis i korisniku se prikazuje poruka koja ga obavještava kako je upisao pogrešan datum. Pseudokod:

```
IF (uneseni datum <= trenutni datum)
    THEN Upiši trening;
    ELSE Ispiši poruku upozorenja;
ENDIF;
```

11. OPIS I IMPLEMENTACIJA IZVJEŠTAJA

11.1. IZVJEŠTAJ TRENINGA

Izveštaj treninga prikazuje korisniku sve upisane treninge sa popisom članova koji su sudjelovali na treningu i opreme koja je korištena na treningu, također omogućuje brisanje ili uređivanje pojedinačnih treninga.

SQL upit: `SELECT * FROM 'Tening';`

11.2. IZVJEŠTAJ ČLANARINA

Izveštaj članarina prikazuje korisniku sve upisane članarine, također omogućuje brisanje ili uređivanje pojedinačnih članarina.

SQL upit: `SELECT * FROM 'Clanarina';`

11.3. IZVJEŠTAJ ČLANOVA

Izveštaj članova prikazuje korisniku sve upisane članove, također omogućuje brisanje ili uređivanje pojedinačnih članova.

SQL upit: `SELECT * FROM 'Clan';`

11.3. IZVJEŠTAJ TRENERA

Izvještaj trenera prikazuje korisniku sve upisane trenere, također omogućuje brisanje ili uređivanje pojedinačnih trenera.

SQL upit: `SELECT * FROM 'Trener';`

11.4. IZVJEŠTAJ OPREME

Izvještaj opreme prikazuje korisniku svu upisanu opremu, također omogućuje brisanje ili uređivanje pojedinačnih dijelova opreme.

SQL upit: `SELECT * FROM 'Oprema';`

11.5. IZVJEŠTAJ KATEGORIJA

Izvještaj kategorija prikazuje korisniku sve upisane kategorije, također omogućuje brisanje ili uređivanje pojedinačnih kategorija.

SQL upit: `SELECT * FROM 'Kategorija';`

11.6. IZVJEŠTAJ UTAKMICA

Izvještaj utakmica prikazuje korisniku sve upisane utakmice, također omogućuje brisanje ili uređivanje pojedinačnih utakmica.

SQL upit: `SELECT * FROM 'Utakmica';`

11.7. IZVJEŠTAJ POGODAKA

Izvještaj pogodaka prikazuje korisniku sve upisane pogotke, također omogućuje brisanje ili uređivanje pojedinačnih pogodaka.

SQL upit: `SELECT * FROM Pogodak_na_utakmici;`

11.8. IZVJEŠTAJ KARTONA

Izvještaj kartona prikazuje korisniku sve upisane kartone, također omogućuje brisanje ili uređivanje pojedinačnih kartona.

SQL upit: `SELECT * FROM Karton_na_utakmici;`

12. PRIKAZ KORIŠTENJA PROGRAMSKOG RJEŠENJA

Slika 9: Primjer unosa podataka

encija trenera Evidencija opreme Evidencija kategorija Evidencija utakmica Evid

Evidencija trenera

Ime:

Prezime:

OIB:

Kontakt broj:

Email address:

Informacijski sustav sportskog kluba

Izvor: autor

Programsko rješenje se sastoji od niza formi koje služe za unos podataka u bazu. Na primjeru (Slika 1) vidimo formu za upis trenera u klub. Kada se novi upisuje novi trener, dostavit će osobne podatke koje će jedan od administratora kluba upisati u formu i klikom na gumb Potvrdi, upisati u bazu podataka. Svaki trener dobiva svoj broj iskaznice koji mu se automatski dodjeljuje u bazi pritiskom na gumb Potvrdi. Novi podaci se također mogu upisivati i u izvještajima. Pritiskom na gumb se otvara izbornik.

Slika 10: Primjer izvještaja

treninga Izvještaj članarina Izvještaj članova Izvještaj trenera Izvještaj opreme Izvještaj kategorija Izvještaj utakmica Izvještaj pogodaka Izvještaj kartona

Izvještaj trenera

Upisani treneri:

Broj iskaznice	Ime	Prezime	OIB	Kontakt broj	E-mail	Opcije
32	Anto	Antić	123	321	Anto.A@gmail.com	Uredi Izbriši
33	Ivan	Ivanović	123	321	Test@t.com	Uredi Izbriši

Izvor: autor

Svi izvještaji se mogu dobiti odabiranjem bilo koje od opcija u izborniku izvještaji. U ovom primjeru aplikacija dohvaća sve upisane trenere iz baze podataka i prikazuje ih u tablici.

Slika 11: Primjer - brisanje podataka 1

treninga Izvještaj članarina Izvještaj članova Izvještaj trenera Izvještaj opreme Izvještaj kategorija Izvještaj utakmica Izvještaj pogodaka Izvještaj kartona

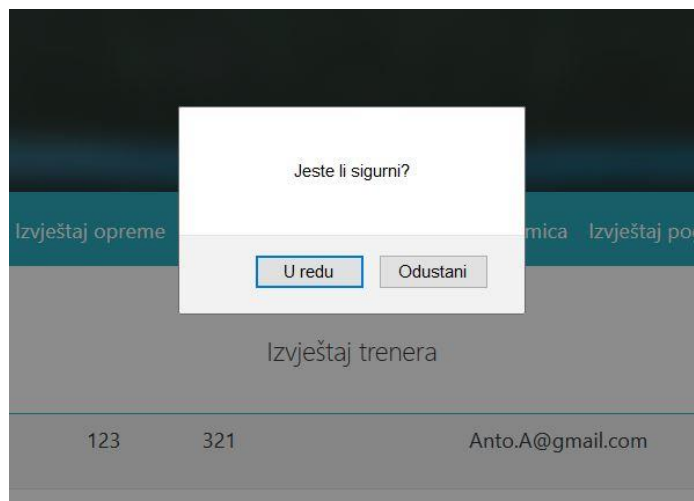
Izvještaj trenera

Upisani treneri:

Broj iskaznice	Ime	Prezime	OIB	Kontakt broj	E-mail	Opcije
32	Anto	Antić	123	321	Anto.A@gmail.com	Uredi Izbriši
33	Ivan	Ivanović	123	321	Test@t.com	Uredi Izbriši

Izvor: autor

Slika 12: Primjer - brisanje podataka 2



Izvor: autor

Unutar svakog izvještaja imamo gumbе za brisanje svakog pojedinačnog upisanog redka. Pritiskom na gumb Izbriši, dobivamo upozorenje koje od nas traži da potvrdimo svoju odluku nakon čega će određeni redak biti izbrisan.

Slika 13: Primjer uređivanja podataka

The image shows a web application interface. At the top, there is a modal window titled "Uredi trenera" (Edit coach) with a close button (x). The form contains the following fields:

- Ime: Anto
- Prezime: Antic
- OIB: 123
- Kontakt broj: 321
- Email address: Anto.A@gmail.com

Below the form is a blue button labeled "Potvrdi".

Below the form is a table titled "Izveštaj trenera" (Coach report). The table has two rows of data:

Izveštaj trenera		
321	Anto.A@gmail.com	Uredi
321	Test@t.com	Uredi

Izvor: autor

Unutar svakog izvještaja također možemo pronaći gumb za uređivanje svakog specifičnog retka. Pritiskom na gumb se otvara se forma za upis podataka odabranog retka. Svi podaci određenog retka su automatski upisani u formu, potrebno je samo izmijeniti željene atribute.

13. ZAKLJUČAK

Sustav sustav je razvijen koristeći Vue, Laravel i Bootstrap frameworke i SQL baze podataka. Baza podataka je nastala na temelju procesa modeliranja procesa i podataka iz relacijske sheme. Sustav uspješno obavlja funkcionalnosti potrebne nogometnom klubu i baziran je na webu. Korisnik može koristiti sustav na bilo kojem mjestu u bilo koje vrijeme putem preglednika.

Za izradu čitavog rada bilo je potrebno u prosjeku tri mjeseca ne računajući vrijeme potrebno za učenje do tada nepoznatih tehnologija koje su korištene u razvoju aplikacije. Najviše je vremena bilo potrebno za samu pripremu za izradu aplikacije, planiranje i izrada dijagrama.

Najveći izazov u izgradnji ovog sustava bio je upoznavanje sa do tada mi nepoznatim tehnologijama frameworka i svladavanje do sada nepoznatih situacija vezanih za web development. Bilo je potrebno uložiti dobar dio vremena na istraživanje tih tehnologija prije upuštanja u razvoj samog sustava.

Mislim da bi se sustav mogao poboljšati uvođenjem dodatnih sigurnosnih mjera, autorizacije korisnika i dodavanjem složenijih izvještaja. Mogli bi se izraditi računi sa određenom razinom autorizacije. Tako bi predsjednik kluba mogao pristupiti svim funkcionalnostima sustava, tajnik, na primjer samo onim funkcionalnostima koje se tiču članarina, upisa novih trenera. Trener bi mogao imati ovlasti samo upisivati nove članove, treninge, utakmice, a skladište bi moglo pristupiti samo operacijama s opremom.

Kada bi se uveli složeniji izvještaji, korisnici sustava bi mogli donositi bolje zaključke na temelju izvještaja. Na primjer, bilo bi dobro da se napravi izvještaj u kojem je moguće prikazati sve

članove koji su u određenom mjesecu platili članarinu, dodati analize podataka, grafikone, izračune prosjeka itd.

Osim toga, ovakav sustav bi bio dovoljan malim nogometnim klubovima. Kako bi služio ozbiljnijem klubu, bilo bi ga potrebno nadograditi s više funkcionalnosti i s profesionalnijim pristupom, obraćajući pažnju na sve dokumente i zakonske odredbe koje jedan takav klub mora poštivati.

Popis literature

1. Baniček, D, © Pro-MIL d.o.o., 2004, Tiskara Varteks, Varaždin.
2. Bud, A, Collins, S, Moll, C, CSS Mastery, ©2006, Apress L.P., 2560 Ninth Street, Suite 219, Berkley, CA 94710, USA.
3. Duckett, J, HTML & CSS: Design and Build Web Sites, ©2011, John Wiley & Sons, Inc., Indianapolis, Indiana
4. Gasston, P, The Morden Web With HTML5, CSS3, and JavaScript, ©2013, No Starch Press.
5. Hayes, D, Sams Tech HTML and XHTML in 10 Minutes, ©2002, Miš, Zagreb.
6. Lerdorf, R, MacIntyre, P, Tatroe, K, Programing PHP, 2nd edition, ©2006, O'Reilly Media, Inc. 1005 Gravenstein, Highway North, Sebastopol, CA 95472.
7. Macrae, C, Vue.js:Up and Running, ©2018, O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472.
8. Powers, S, Learning JavaScript, 2nd edition,©2009, O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472.
9. Shenoy, A, Ulrich Sossou, Learning Bootstrap,©2014, Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK.
10. Stauffer, M, Laravel:Up and Running,©2017, O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Popis kratica

HTML (<i>Hypertext markap language</i>).....	2
CSS (<i>Cascading style sheets</i>).....	3
PHP (Hypertext procesor).....	7
ORM (Object-relational mapping).....	10

Popis tablica

Tablica 1: Tok Podataka - Upisni podaci trenera.....	32
Tablica 2: Skladište podataka - Treneri.....	32
Tablica 3: Tok podataka - Upisni podaci člana.....	33
Tablica 4: Skladište podataka - Članovi.....	33
Tablica 5: Tok podataka - Kategorije u klubu.....	34
Tablica 6: Skladište - Kategorije.....	34
Tablica 7: Tok podataka - Oprema u klubu.....	35
Tablica 8: Skladište - Oprema.....	35
Tablica 9: Tok podataka - Podaci o treningu.....	36
Tablica 10: Skladište - Treninzi.....	36
Tablica 11: Tok podataka - Podaci o članarini.....	37
Tablica 12: Skladište - Članarine.....	37
Tablica 13: Tok podataka - Podaci o utakmici.....	38
Tablica 14: Skladište - Utakmice.....	38
Tablica 15: Tok podataka - Podaci o pogodku.....	39
Tablica 16: Skladište - Pogodci.....	39
Tablica 17: Tok podataka - Podaci o kartonu.....	40
Tablica 18: Skladište - Kartoni.....	40
Tablica 19: Popis tipova entiteta.....	42
Tablica 20: Atributi tipa entiteta TRENER.....	43

Tablica 21: Atributi tipa entiteta TRENING.....	43
Tablica 22: Atributi tipa entiteta OPREMA.....	43
Tablica 23: Atributi tipa entiteta ČLAN.....	44
Tablica 24: Atributi tipa entiteta KATEGORIJA.....	44
Tablica 25: Atributi tipa entiteta ČLANARINA.....	44
Tablica 26: Atributi tipa entiteta POGODAK_NA_UTAKMICI.....	45
Tablica 27: Atributi tipa entiteta KARTON_NA_UTAKMICI.....	45
Tablica 28: Atributi tipa entiteta UTAKMICA.....	45
Tablica 29: Veze EVA modela.....	46

Popis slika

Slika 1: Primjer uobičajenih slektora.....	3
Slika 2: Primjer metoda povezivanja CSS-a i HTML-a unutar dokumenta.....	5
Slika 3: Primjer metoda povezivanja CSS-a i HTML-a unutar elementa.....	5
Slika 4: Primjer metoda povezivanja CSS-a i HTML-a iz vanjskog dokumenta.....	6
Slika 5: Grafičko korisničko sučelje Visual Studio Code-a.....	9
Slika 6: Prikaz dodjeljivanja klasa putem v-bind direktive.....	14
Slika 7: Grafičko korisničko sučelje aplikacije na zaslonu prijenosnog računala.....	16
Slika 8: Grafičko korisničko sučelje aplikacije na zaslonu mobilnog uređaja.....	17
Slika 9: Primjer unosa podataka.....	57
Slika 10: Primjer izvještaja.....	58
Slika 11: Primjer - brisanje podataka 1.....	59
Slika 12: Primjer - brisanje podataka 2.....	59
Slika 13: Primjer uređivanja podataka.....	60

Popis dijagrama

Dijagram 1: Dijagram dekompozicije funkcija.....	22
Dijagram 2: Dijagram toka podataka nulte razine.....	24
Dijagram 3: Dijagram toka podataka prve razine.....	26
Dijagram 4: Dijagram toka podataka druge razine.....	29
Dijagram 5: EVA model.....	41
Dijagram 6: Relacijski model.....	49
Dijagram 7: DPP dijagram.....	50