

# DIZAJN SISTEMA

## STONOTENISKI KLUB

Dragan Ilić  
Dejan Gajić  
Đorđe Turjačanin  
Maja Mihajlović  
Marko Malinović  
Helena Brkanlić  
Petar Stojanović

## Sadržaj

1. Uvod .....	3
1.1. Namjena sistema.....	3
1.2. Projektni ciljevi.....	3
1.3. Definicije i skraćenice.....	3
1.4. Reference .....	3
1.5. Pregled .....	3
2. Pregled postojećih arhitektura .....	4
3. Predložena arhitektura .....	4
3.1. Kratak pregled aritekture i funkcionalnosti podsistema .....	4
Funkcionalnosti podsistema: .....	5
3.2. Dekompozicija sistema.....	5
Dijagram paketa .....	5
3.3. HW/SW mapiranje .....	6
Dijagram komponenata .....	6
Dijagram razmještaja: .....	6
3.4. Perzistentni sloj .....	7
Konceptualna šema baze podataka sistema .....	7
3.5. Kontrola prava pristupa i sigurnost.....	8
3.6. Kontrola toka.....	8
3.7. Granična stanja sistema .....	9
Dijagram slučajeva upotrebe za granična stanja .....	9

## 1. Uvod

Ovaj dokument opisuje način dizajniranja sistema, šablone koji će se koristiti prilikom njegovog projektovanja, kao i dodatne funkcionalnosti koje softver treba da ima, a koji nisu navedeni u specifikaciji softverskih zahtjeva.

### 1.1. Namjena sistema

„Stonoteniski klub“ je softver koji se koristi u stonoteniskom klubu za lakše obavljanje svih obaveza vezanih za klub.

### 1.2. Projektni ciljevi

Softver će omogućiti zaposlenima da vode evidenciju o članovima, treninzima i turnirima, vođenje administrativnih poslova i evidentiranje novčanih sredstava.

### 1.3. Definicije i skraćenice

Pojam	Definicija
Korisnik	Osoba koja koristi softver
Administrator	Korisnik koji je zadužen za održavanje i kontrolisanje softvera
MVC	Model View Controller
JRE	Java Runtime Enviroment
DBMS	Sistem za upravljanje bazom podataka

### 1.4. Reference

[1] Specifikacija softverskih zahtjeva - Stonoteniski klub, 2017.

### 1.5. Pregled

Ostatak dokumenta je podijeljen u dvije glave, gdje sljedeća glava opisuje postojeće arhitekturne stilovi koji su razmatrani za korištenje , a u trećoj glavi se nalaze detaljni opisi djelova predložene arhitekture finalnog sistema.

## 2. Pregled postojećih arhitektura

Najčešće korišteni arhitekturni stilovi su:

- MVC
- 3-slojni (3-Layer) arhitekturni stil
- 4-slojni (4-Layer) arhitekturni stil

MVC klasifikuje podsisteme u tri kategorije:

- Model : reprezentacija i pristup domenskim objektima
- View : predstavljanje podataka krajnjem korisniku
- Controller : dio sistema odgovaran za interakciju između view-a i controller-a

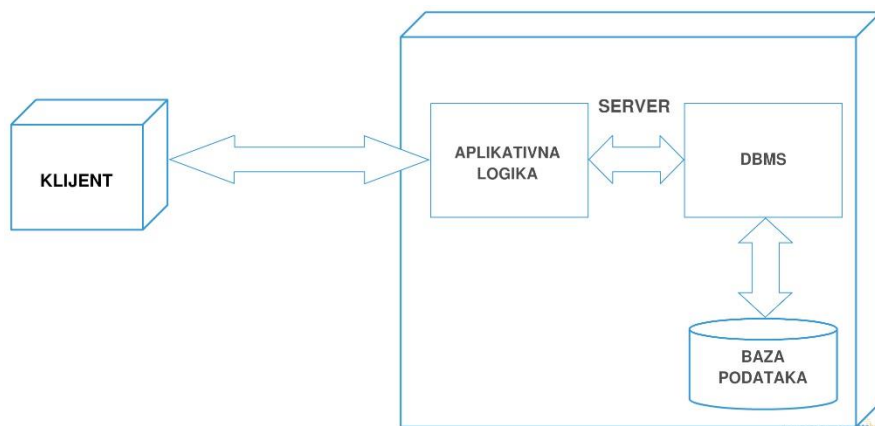
3-slojni arhitekturni stil podrazumijeva tri virtuelne mašine kod kojih postoji sloj prezentacije, sloj aplikacije i sloj podataka. Tipično se koristi u web programiranju.

4-slojni arhitekturni stil podrazumijeva četiri hijerarijska podsistema. Najčešće se koristi kod velikih poslovnih internet sistema.

## 3. Predložena arhitektura

### 3.1. Kratak pregled aritekture i funkcionalnosti podsistema

Sistem se sastoji iz klijentskog i serverskog dijela. Klijentski dio predstavlja desktop aplikaciju, koja će komunicirati sa serverom i od njega dobavljati potrebne podatke. Serverski dio sadrži bazu podataka i aplikativnu logiku preko koje klijent komunicira sa DBMS-om. Ovaj proizvod nije dio nekog drugog većeg sistema, već je u potpunosti nezavisan proizvod.



## Funkcionalnosti podsistema:

Dio softvera je zadužen za praćenje toka i organizovanja turnira. Kalendar aktivnosti kluba je servis dostupan svim korisnicima softvera pružajući im uvid u termine održavanja različitih klupskih dešavanja.

Evidencija o zaposlenima i članovima kluba se ažurira prilikom učlanjivanja, iščlanjivanja, zapošljavanja, te prekida radnog odnosa. Administrativni detalji o kojima se vodi evidencija su detalji o finansijama, sponzorima, distributerima opreme, ugovorima i zapisnicima sa sjednica.

Dio softvera zadužen za rad sa statistikom podrazumijeva omogućavanje prikaza i dodavanja podataka o treninzima, praćenje i ažuriranje stanja rang liste i čuvanje ostvarenih rezultata na turnirima za svakog korisnika ponaosob.

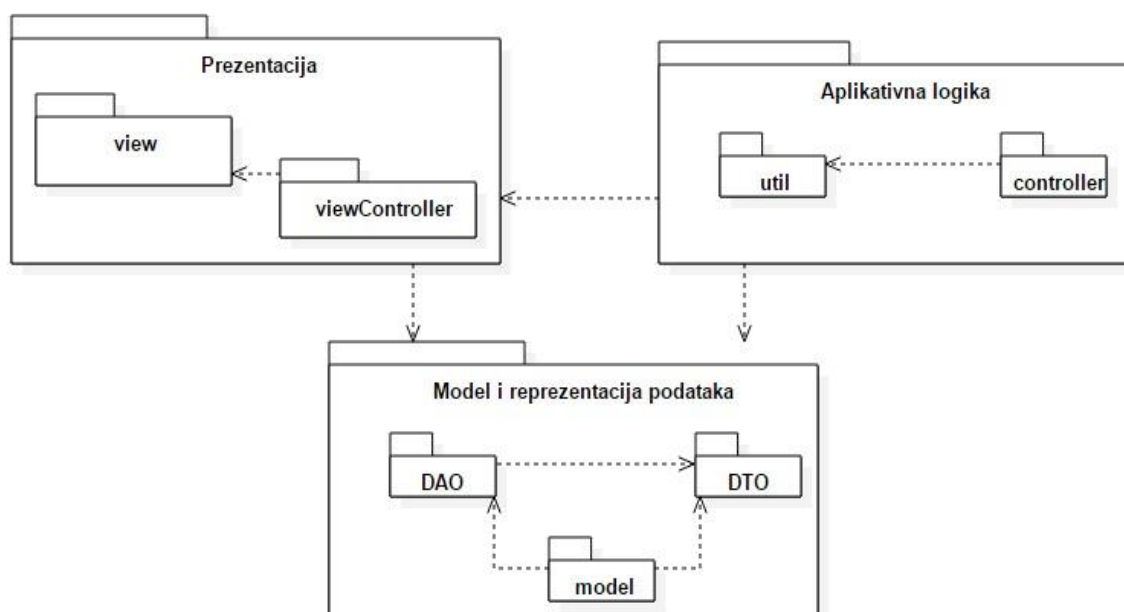
Dokumenti koje softver kreira i priprema za štampanje su različite vrste potvrda, dnevni redovi i zapisnici sjednica, finansijski izvještaji, te zahtjevi za registraciju igrača. Svi navedeni dokumenti se kreiraju po odgovarajućem šablonu.

Softver pruža korisnicima mogućnost evidentiranja učesnika turnira kao i kreiranja i ažuriranja žrijeba po kojem se odigravaju mečevi na turniru.

## 3.2. Dekompozicija sistema

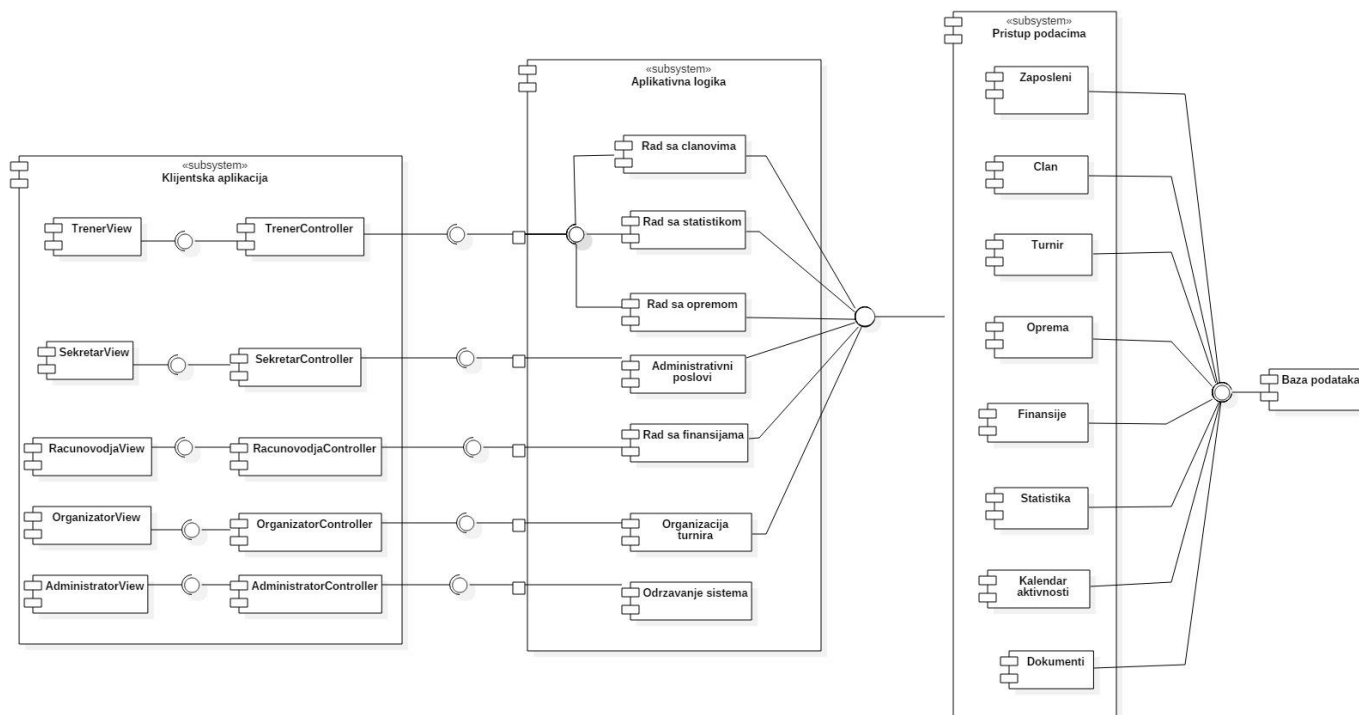
### Dijagram paketa

Sistem je modelovan prema principu MVC arhitekture.



### 3.3. HW/SW mapiranje

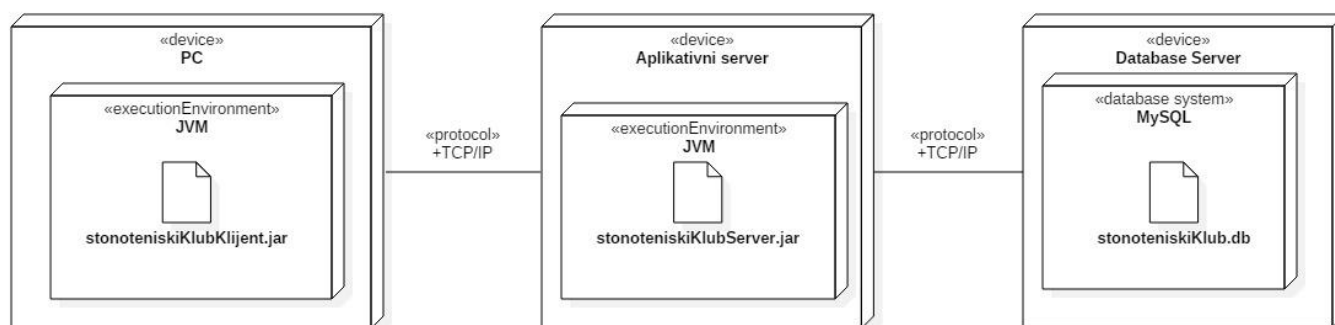
#### Dijagram komponenta



#### Dijagram razmještaja:

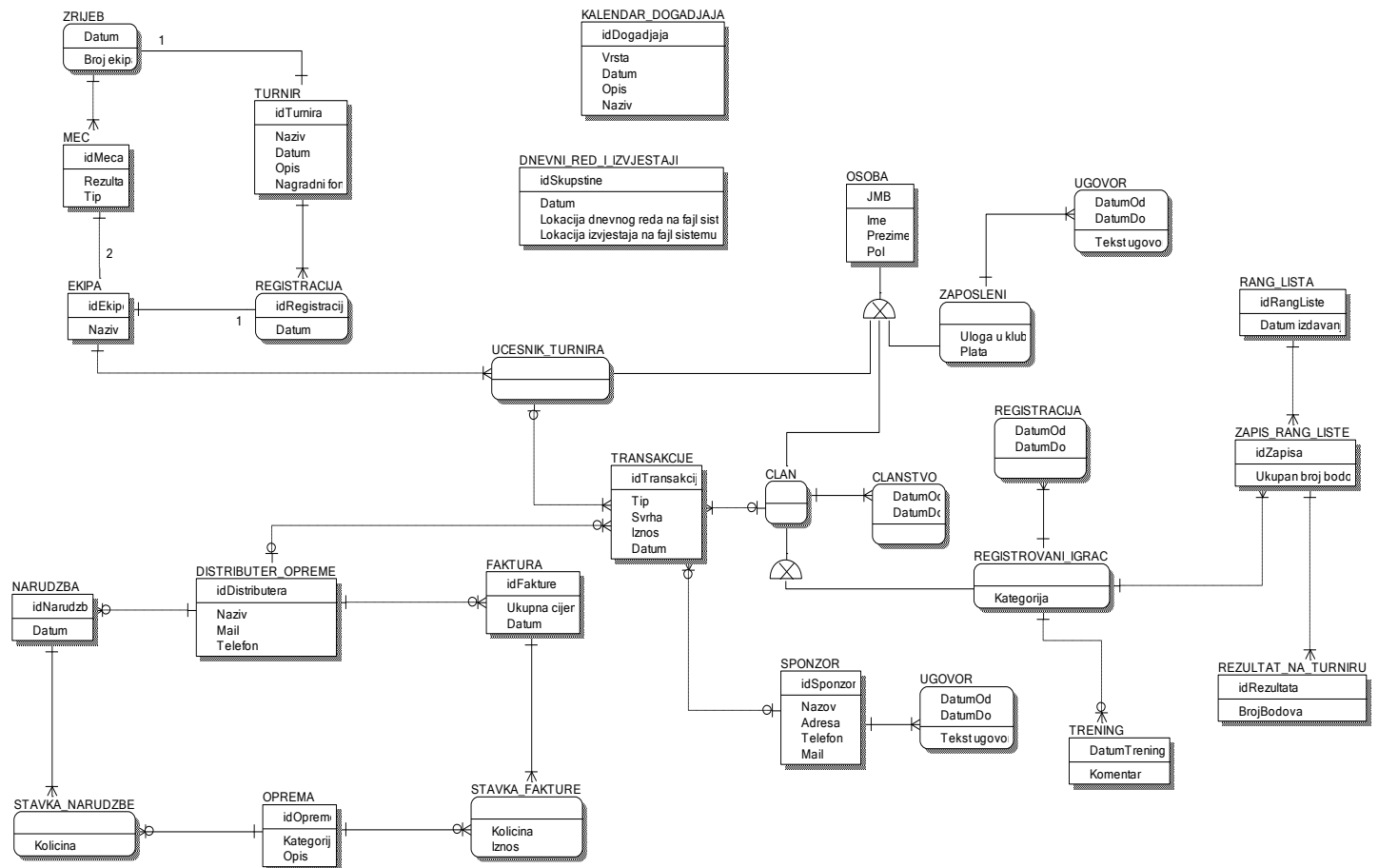
Na klijentskom računaru se nalazi Java aplikacija koja se sastoji iz dijela namijenjog za kreiranje grafičkog korisničkog interfejsa i interakciju sa korisnikom, dok je drugi dio aplikacije namijenjen za komunikaciju sa serverom.

Na serveru se nalazi aplikativna logika namijenjena za rad sa bazom podataka, dobavljanjem potrebnih podataka sa interneta, komunikacijom sa klijentima i obrađivanjem klijentskih zahtjeva.



### 3.4. Perzistentni sloj

#### Konceptualna šema baze podataka sistema



### 3.5. Kontrola prava pristupa i sigurnost

Korisnik treba da se autentifikuje prije nego što dobije dozvolu da koristi aplikaciju. Autentifikacija korisnika se vrši na osnovu korisničkog imena i lozinke, pri čemu korisničko ime mora biti jedinstveno.

Prava pristupa učesnika metodama pojedinih klasa su prikazana preko globalne matrice pristupa.

KLASE	UČESNICI				
	Administrator	Sekretar	Trener	Računovođa	Organizator
Klub	IzmjeniPodatke()	X	X	X	X
Zaposleni	dodajNalog() obrisiNalog() azurirajNalog()	pregledaj ()	X		X
Potvrda	X	stampaj ()	X	X	X
PodaciOBudzetu	X	X	X	dodajUplatu() dodajIsplatu()	X
DnevniRedSkupstine	X	dodajTacku () obrisiTacku () stampaj ()	X	X	X
Izvjestaj	X	stampaj ()	X	X	X
ZahtjevZaRegistracijulgraca	X	X	stampaj()	X	X
Clan	X	X	dodaj() obrisi() azurirajClanstvo() azurirajPodatke()	X	X
UcesnikTurnira	X	X	X	X	dodaj()
Turnir	X	X	X	X	dodajZrijeb() azurirajZrijeb()
Mec	X	X	X	X	dodajUcesnike()
Zrijeb	X	X	X	X	dodajMec()
Registrovanilgrac	X	X	dodajTrening() dodajRezTurnira() azurirajRegistraciju()	X	X
RangLista	X	X	azuriraj()	X	X
RezultatNaTurniru	X	X	azuriraj()	X	X
Sponzor	X	dodaj()	X	X	X
KalendarDogadjaja	X	dodajDogadjaj() obrisiDogadjaj()	X	X	X
Oprema	X	X	pregledaj() dodaj() azuriraj()	X	X
Narudzba	X	X	dodajStavku() brisiStavku() odaberiDistributera() dodajDistributera()	X	X
Faktura	X	dodajStavku() brisiStavku()	X	X	X

### 3.6. Kontrola toka

Kontrola toka predstavlja redoslijed izvršavanja akcija. U objektno orijentisanim sistemima ona uključuje redoslijed izvršavanja metoda.

Kontrola toka ovoga sistema je centralizovana, i koristi mehanizam upravljanja događajima.



### 3.7. Granična stanja sistema

#### Dijagram slučajeva upotrebe za granična stanja

Prije početka korištenja sistema potrebno je kreirati početni (administratorski) nalog. Prilikom prvog pokretanja aplikacije prikazaće se forma za kreiranje početnog naloga. Tek nakon kreiranog administratorskog naloga, moguće je upravljati bazom podataka i dodavati nove korisničke naloge.

Administrator ima mogućnost unosa podataka o klubu. Podaci koji se trebaju unijeti su:

- Naziv
- Adresa
- Telefon
- E-mail

Po potrebi, podaci o klubu se mogu mijenjati.

Administrator ima mogućnost upravljanja bazom podataka.

