

# Baza podataka i web-aplikacija za teniski klub

---

Bešenić, Luka

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:598019>

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

Download date / Datum preuzimanja: **2025-03-28**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1469

# **BAZA PODATAKA I WEB-APLIKACIJA ZA TENISKI KLUB**

Luka Bešenić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1469

# **BAZA PODATAKA I WEB-APLIKACIJA ZA TENISKI KLUB**

Luka Bešenić

Zagreb, lipanj 2024.

## ZAVRŠNI ZADATAK br. 1469

Pristupnik: **Luka Bešenić (0036538565)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: prof. dr. sc. Boris Vrdoljak

Zadatak: **Baza podataka i web-aplikacija za teniski klub**

### Opis zadatka:

U sklopu ovog završnog rada potrebno je izraditi bazu podataka u koju će se pohranjivati podaci o članovima teniskog kluba, njihovim aktivnostima i sportskim rezultatima. Nakon oblikovanja modela entiteti-veze i odgovarajućeg relacijskog modela baze podataka, treba implementirati bazu podataka koristeći sustav za upravljanje bazom podataka PostgreSQL. Potrebno je zatim korištenjem radnih okvira React i Spring te programskih jezika Java i Typescript napraviti web-aplikaciju koja omogućava pregled, unos, brisanje i izmjenu podataka vezanih za vođenje evidencije rada sportskog kluba.

Rok za predaju rada: 14. lipnja 2024.



# Sadržaj

Uvod .....	1
1. Korišteni alati i tehnologije .....	2
1.1. IntelliJ IDEA .....	2
1.2. React .....	2
1.3. Spring Boot .....	3
1.4. PostgreSQL .....	3
2. Oblikovanje modela baze podataka .....	4
2.1. ER model baze podataka .....	4
2.1.1. Opis ER modela baze podataka .....	5
2.1.2. Popis entiteta i atributa .....	6
2.1.3. Popis veza .....	7
2.2. Relacijski model baze podataka .....	8
3. Specifikacija programske potpore .....	17
3.1. Funkcionalni zahtjevi .....	17
3.2. Arhitektura i dizajn sustava .....	31
3.3. Dijagrami razreda .....	32
4. Korisničke upute .....	35
4.1. Početni zaslon .....	35
4.2. Pregled podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima .....	35
4.3. Pretraživanje podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima .....	38
4.4. Unos podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima .....	39
4.5. Izmjena i brisanje podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima .....	40
Zaključak .....	42
Literatura .....	43
Sažetak .....	44
Summary .....	44

## Uvod

U suvremenom svijetu, informacijske tehnologije igraju ključnu ulogu u unapređenju poslovanja i organizacije različitih institucija, uključujući sportske klubove. Teniski klubovi, koji okupljaju velik broj članova i organiziraju brojne aktivnosti i natjecanja, suočavaju se s izazovom vođenja točne i ažurirane evidencije o članovima, njihovim aktivnostima i sportskim rezultatima. U tu svrhu, izrada baze podataka i web aplikacije predstavlja efikasno rješenje koje omogućava lakše upravljanje i pristup relevantnim informacijama.

Proces izrade ovakvog integriranog sustava započinje dizajniranjem baze podataka, što uključuje stvaranje relacijske baze podataka, koja će služiti za pohranu podataka. Relacijska baza podataka omogućava strukturirano spremanje informacija te olakšava pretraživanje i manipulaciju podacima. U ovom projektu, za implementaciju baze podataka odabran je PostgreSQL, poznat po svojoj pouzdanosti, skalabilnosti i bogatom skupu funkcionalnosti.

Kao posredničko sredstvo između korisnika i baze podataka, stvorit će se jednostavno i intuitivno korisničko sučelje putem kojeg će biti moguće pregledavati, pohranjivati, brisati i mijenjati podatke vezane uz teniske klubove, njihove članove i pripadajuće aktivnosti. Korisničko će sučelje biti razvijeno korištenjem modernih web tehnologija i radnih okvira. Za *frontend* aplikacije koristit će se React, dok će se za *backend* koristiti Spring Java. Korištenjem TypeScripta umjesto JavaScripta dodatno će se osigurati statička tipizacija, što će doprinijeti većoj pouzdanosti i lakšem održavanju koda. Integracija svih ovih tehnologija omogućit će izradu sveobuhvatnog sustava za upravljanje teniskim klubom, koji će pružiti brojne funkcionalnosti.

U prvome poglavlju završnog rada opisani su korišteni alati i tehnologije pri implementaciji aplikacije. Drugo poglavlje opisuje proces nastanka baze podataka za aplikaciju, a treće poglavlje specifikaciju programske potpore, što uključuje funkcionalne zahtjeve i dijagram slučaja uporabe, arhitekturu sustava i dijagrame razreda. U posljednjem, četvrtom poglavlju dane su korisničke upute za korištenje aplikacije za teniski klub. Konačno, u zaključku su sažeti rezultati rada.

# 1. Korišteni alati i tehnologije

## 1.1. IntelliJ IDEA

IntelliJ IDEA integrirano je razvojno okruženje (engl. *Integrated Development Environment*) koje služi kako bi se razvijao softver, između ostalog, u Javi. Pokrenuto je 2001. godine, a do danas je steklo veliku popularnost zahvaljujući svojim sposobnostima da poboljša produktivnost razvijatelja programske podrške.

Glavne su karakteristike ovog okruženja mogućnost pametnog dovršavanja koda, tako da na temelju konteksta predlaže najvjerojatnije opcije za dovršavanje dijela koda; zatim analiza programskog koda i dojava mogućih pogrešaka s predloženim ispravcima. IDE podržava mnoge programske jezike i okvire poput Kotlin, Groovy, Scala, te integraciju sa sustavima kao što su Git i SVN. Build alati kao Maven i Gradle olakšavaju automatizaciju izgradnje aplikacija. Dodatno, IntelliJ IDEA može se prilagoditi specifičnim potrebama projekta kroz bogatu biblioteku dodataka.

## 1.2. React

React je JavaScript biblioteka koja služi za izgradnju *frontend* dijela aplikacije, odnosno korisničkih sučelja. Pokrenuta je 2013. godine i jedan je od poznatijih alata zbog jednostavnosti i efikasnosti u izradi dinamičkih i responzivnih web aplikacija.

Važno svojstvo React-a je korištenje komponenti, koje predstavljaju ponovno iskoristive dijelove koda koji predstavljaju dijelove korisničkog sučelja. One mogu biti vrlo jednostavne, kao što su gumb, link ili polje za unos, a mogu biti i vrlo složene, kao što su navigacijske trake ili već gotove forme. JavaScript XML (JSX) je sintaksa s pomoću koje je moguće pisati HTML kod unutar JavaScript koda, a s kojom je dobiveno na većoj čitljivosti koda. Još jedna važna karakteristika jest Virtual DOM, kojeg React ažurira kada se stanje aplikacije promijeni te sinkronizira stvarni DOM, čime se poboljšavaju performanse aplikacije.

React se lako integrira s različitim *backend* tehnologijama, često u kombinaciji s Node.js za optimizirano dohvaćanje podataka. Zahvaljujući aktivnoj zajednici i stalnim ažuriranjima, React ostaje nerijetki izbor za web developere.



### 1.3. Spring Boot

Spring Boot je okvir otvorenog koda u Javi korišten za stvaranje i programiranje aplikacija koje se mogu jednostavno pokrenuti. Prvo je izdanje pušteno u pogon 2014. godine, a izvorni je autor Rod Johnson.

Prednosti koje pruža Spring Boot su razumljivost i jednostavnost u razvijanju aplikacija, povećanje produktivnosti i smanjenje vremena razvoja tako da se nudi lakši način za početak rada s aplikacijom. Spring Boot upotrebljava predefinirane konfiguracije te tako omogućava razvoj aplikacija s minimalno XML konfiguracije, što u velikoj mjeri smanjuje složenost razvoja. Podržane su također brojne tehnologije i radni okviri poput Hibernate ili Spring Security što doprinosi povećanju produktivnosti. Spring Boot nadalje pruža snažnu podršku za rad s bazama podataka i odlično se integrira s popularnim ORM (engl. *Object-Relational Mapping*) alatima, čime se ostvaruje mapiranje Java objekata na relacijske baze podataka. Time je pojednostavljeno povezivanje, konfiguracija i upravljanje bazama podataka.

### 1.4. PostgreSQL

PostgreSQL je sustav koji služi za jednostavno upravljanje bazama podataka koji podržava standard SQL. Inicijalno je pušten u javnost 1996. godine, a razvijen je od strane PostgreSQL Global Development Group.

Važna je karakteristika PostgreSQL-a njegova proširivost. Pod time je podrazumijevana mogućnost definiranja vlastitih podatkovnih tipova, funkcija i operatora. Na taj je način moguće prilagoditi bazu podataka odgovarajućim potrebama pojedine aplikacije. Za aplikacije za koje je nužno osigurati visoku razinu sigurnosti podataka bitne su ACID (engl. *atomicity, consistency, isolation, durability*) transakcije. One osiguravaju pouzdanost transakcija, kao i integritet podataka. PostgreSQL također podržava mnogobrojne programske jezike i alate, kao što su Java, Python i C++ te je shodno tome čest odabir programera.

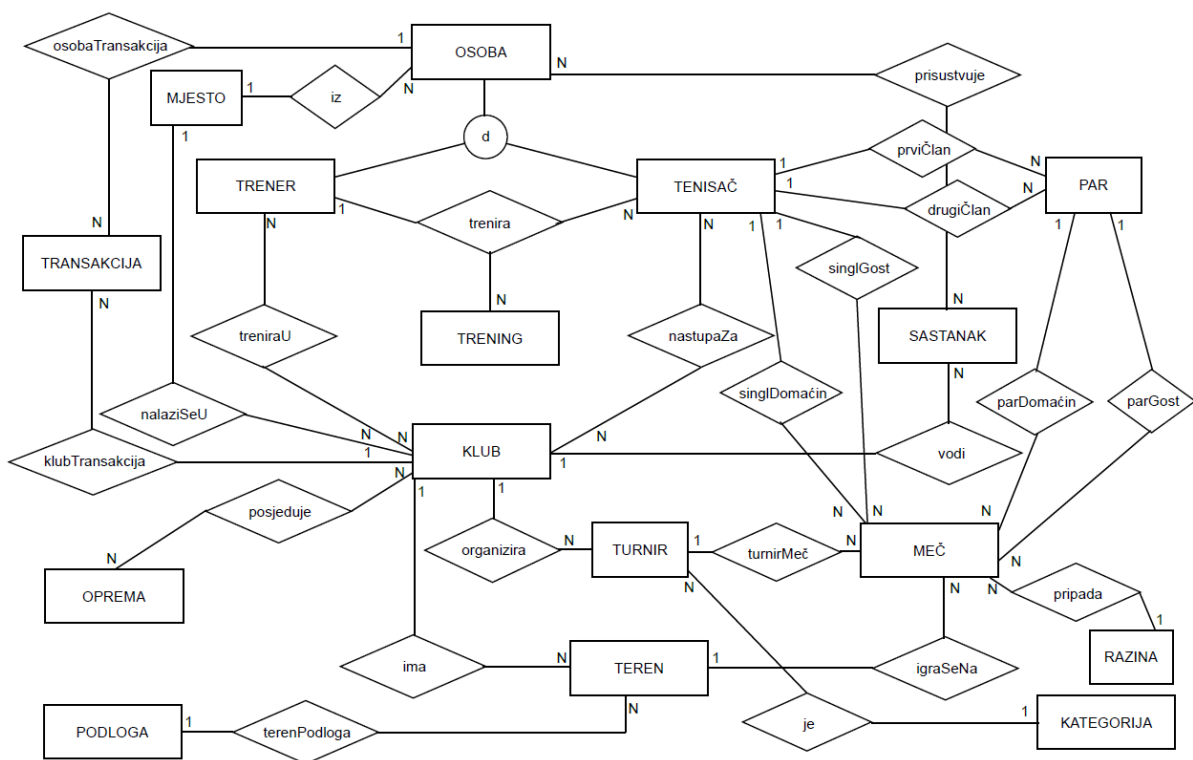
Uz sve navedeno, redovita ažuriranja i aktivna zajednica osiguravaju da je PostgreSQL jedan od najpouzdanijih i najkorištenijih sustava za upravljanje bazama podataka.

## 2. Oblikovanje modela baze podataka

### 2.1. ER model baze podataka

Baza podataka kao organiziran i logički strukturiran skup podataka čini temeljni dio bilo koje aplikacije, pa tako i aplikacije za teniski klub. Stoga, prvotno valja oblikovati model iste.

ER (engl. *Entity-Relationship Model*) model jedan je od osnovnih načina na koji se grafički može prikazati model baze podataka. Navedeni se model sastoji od niza entiteta, odnosno osnovnih elemenata koji modeliraju određene podatke pomoću odgovarajućih atributa, te veza koje ih međusobno smisljeno povezuju. Na slici 2-1 prikazan je ER model baze podataka aplikacije za teniski klub.



Slika 2-1 ER dijagram baze podataka za teniski klub

### 2.1.1. Opis ER modela baze podataka

Baza podataka za teniski klub sadrži entitet klub, koji je opisan jedinstvenom šifrom, svojim nazivom, godinom osnutka, email-om, telefonskim brojem, web-stranicom (ako ju ima) te budžetom. Za svaki teniski klub nastupa više tenisača i svaki klub može imati više trenera. Tenisači i treneri specijalizacija su entiteta osoba, koji ima jedinstveni oib, ime, prezime, datum rođenja i spol. Tenisač je dodatno opisan atributima visine, težine, snažnije ruke, ranga i podatkom o ozlijedi, ako je ima. Trener je, s druge strane, dodatno opisan godinama iskustva i specijalizacijom. Svaka osoba može u jednom trenutku biti u samo jednom klub, dok kroz karijeru može mijenjati klubove, s pretpostavkom da će klubovi kroz karijeru biti različiti. S time, svakoj se osobi za određeni klub definira vremenski period u kojemu je član kluba. Za svaku se osobu i klub bilježi iz kojeg je mjesta, koje ima jedinstveni atribut poštanski broj i atribut naziv mjesta.

U bazi se također vodi evidencija treninga, koji ima jedinstvenu šifru, datum i vrijeme održavanja, trajanje, opis i bilješke vezane uz pojedini trening. Jedan trening provodi jedan trener, a na njemu sudjeluje više tenisača. Osobe i klubovi mogu izvršavati transakcije, koje imaju datum i vrijeme, iznos, način uplate/isplate i opis. Klub može voditi sastanke, opisane jedinstvenom šifrom, datumom i vremenom sastanka, dnevnim redom i bilješkama sa sastanka. Na sastanku prisustvuju osobe iz kluba.

Svaki teniski klub posjeduje određenu opremu, koja ima šifru, naziv i cijenu. Također, bilježi se količina pojedine opreme koju klub posjeduje. Teniski klub može organizirati turnire, za koji su definirani jedinstvena šifra i naziv. Svaki turnir je određene kategorije, koja je definirana šifrom, tipom (singl / parovi), dobnim ograničenjem i spolnim ograničenjem. Turniri se sastoje od mečeva, koji imaju jedinstvenu šifru, datum i vrijeme održavanja, rezultat i trajanje meča te su određene razine, koja je posebni entitet. Meč mogu igrati igrač protiv igrača ili par protiv para. Par je zaseban entitet i ima jedinstvenu šifru, rang i opcionalni atribut datuma do kada je par aktivno igrao. Igrač, odnosno par u svakome meču je ili nominalni domaćin ili nominalni gost. Svaki se meč igra na terenu, koji je u posjedu kluba koji organizira turnir, i koji ima jedinstvenu šifru i naziv. Teren može biti različite podloge (travnata, zemljana, tvrda), a podloga je opisana šifrom i nazivom.

## 2.1.2. Popis entiteta i atributa

Za svaki od entiteta s ER dijagrama potrebno je definirati njegove attribute koje će ga pobliže opisati. To podrazumijeva identificiranje svih bitnih karakteristika koje određuju entitet i koje će omogućiti detaljno praćenje podataka u sustavu. Atributi predstavljaju specifične informacije koje su vezane uz pojedini entitet i ključni su za pravilno funkcioniranje baze podataka.

Svaki entitet može imati jedan ili više atributa, pri čemu je potrebno jasno razlikovati primarne ključeve od ostalih atributa. Primarni su ključevi atributi koji jedinstveno identificiraju svaki zapis unutar entiteta, čime se osigurava integritet podataka unutar baze.

U nastavku slijedi detaljan popis svih entiteta i njihovih atributa, pri čemu su primarni ključevi svakog od entiteta masno otisnuti. Ovaj popis omogućit će bolji uvid u strukturu podataka i olakšati razumijevanje odnosa među entitetima.

OSOBA: **oib**, ime, prezime, datum\_rodjenja, spol

TRENER: **oib**, godine\_iskustva, specijalizacija

TENISAČ: **oib**, visina, tezina, snaznija\_ruka, rang, ozljeda

TRENING: **sifra\_trening**, datum\_vrijeme\_trening, trajanje, opis, biljeske

KLUB: **sifra\_klub**, naziv\_klub, godina\_osnutka, email, telefon, web\_stranica, budzet

MJESTO: **post\_broj**, naziv\_mjesto

SASTANAK: **sifra\_sastanak**, datum\_vrijeme\_sastanak, dnevni\_red, biljeske

TRANSAKCIJA: **sifra\_transakcija**, datum\_vrijeme\_transakcija, iznos, nacin\_placanja, opis

OPREMA: **sifra\_oprema**, naziv\_oprema, cijena

TURNIR: **sifra\_turnir**, naziv\_turnir

KATEGORIJA: **sifra\_kategorija**, tip, dobno\_ogranicenje, spolno\_ogranicenje

MEČ: **sifra\_mec**, datum\_vrijeme\_mec, rezultat, trajanje

TEREN: **sifra\_teren**, naziv\_teren

PODLOGA: **sifra\_podloga**, naziv\_podloga

PAR: **sifra\_par**, rang, datum\_prestanka

RAZINA: **sifra\_razina**, naziv\_razina

### 2.1.3. Popis veza

Entiteti su na ER dijagramu međusobno povezani vezama, koje imaju određene atribute. Veze između entiteta predstavljaju logičke odnose među podacima i omogućuju njihovu integraciju unutar baze podataka. Svaka veza može imati svoje specifične atribute koji pružaju dodatne informacije o odnosu između povezanih entiteta. Primarni ključevi veza jedinstveno identificiraju svaku instancu veze i naglašeni su masnim otiskom. U nastavku slijedi popis svih veza s ER modela baze podataka za teniski klub.

trenira: oib\_trener, **oib\_tenisac**, **sifra\_trening**

treniraU: **oib**, **sifra\_klub**, od, do

nastupaZa: **oib**, **sifra\_klub**, od, do

iz: **oib**, post\_broj

nalaziSeU: **sifra\_klub**, post\_broj

osobaTransakcija: **sifra\_transakcija**, oib

klubTransakcija: **sifra\_transakcija**, sifra\_klub

vodi: **sifra\_sastanak**, sifra\_klub

prisustvuje: **oib**, **sifra\_sastanak**

organizira: **sifra\_turnir**, sifra\_klub

turnirMeč: **sifra\_mec**, sifra\_turnir

posjeduje: **sifra\_klub**, **sifra\_oprema**, dostupna\_kolicina

igraSeNa: **sifra\_mec**, sifra\_teren

ima: **sifra\_teren**, sifra\_klub

terenPodloga: **sifra\_teren**, sifra\_podloga

je: **sifra\_turnir**, sifra\_kategorija

prviČlan: **šifra\_par**, oib

drugiČlan: **šifra\_par**, oib

singlDomaćin: **sifra\_mec**, oib

singlGost: **sifra\_mec**, oib

parDomaćin: **sifra\_mec**, sifra\_par

parGost: **sifra\_mec**, sifra\_par

pripada: **sifra\_mec**, sifra\_razina

## 2.2. Relacijski model baze podataka

Kako bi se baza podataka mogla implementirati u aplikaciju, potrebno je napravljene ER model pretvoriti u relacijski model baze podataka. Proces pretvaranja ER modela u relacijski model obuhvaća nekoliko ključnih koraka kako bi se osiguralo da baza podataka bude pravilno strukturirana.

Prva razlika između ER modela i relacijskog modela je način na koji se tretiraju veze između entiteta. Na primjer, veze tipa N:N i ternarne veze (koje uključuju tri entiteta) ne mogu se direktno implementirati u relacijskom modelu. Umjesto toga, takve se veze preslikavaju u zasebne entitete. Ove tablice će sadržavati primarne ključeve iz entiteta koji sudjeluju u vezi kao strane ključeve, čime se omogućuje održavanje pravilnih veza između originalnih entiteta.

Dalje, primarne ključeve iz veza tipa 1:N potrebno je integrirati na odgovarajući način. U ovim slučajevima, primarni ključ entiteta na strani „jedan“ postaje strani ključ u entitetu na strani „više“. To znači da će se strani ključ smjestiti u tablicu koja predstavlja entitet s višestrukom spojnošću.

Rezultat ovog postupka je skup dobivenih relacija, odnosno tablica, koje jasno prikazuju strukturu baze podataka. Ove relacije su prikazane u tablicama i vizualno predstavljene na relacijskom dijagramu, koji je označen s 2-2.

### Osoba

Atribut	Opis	Tip podatka	Ograničenja
oib	jedinstveni identifikator osobe	char(11)	primarni ključ
ime	ime osobe	varchar	
prezime	prezime osobe	varchar	
datum_rođenja	datum rođenja osobe	date	
spol	spol osobe	char(1)	
post_broj	poštanski broj mjesta stanovanja	int	strani ključ

## Trener

Atribut	Opis	Tip podatka	Ograničenja
oib	jedinstveni identifikator osobe	char(11)	primarni ključ, strani ključ
godine_iskustva	godine trenerovog iskustva	int	
specijalizacija	specijalizacija trenera	varchar	

## Tenisač

Atribut	Opis	Tip podatka	Ograničenja
oib	jedinstveni identifikator osobe	char(11)	primarni ključ, strani ključ
visina	tenisačeva visina	double	
tezina	tenisačeva težina	double	
snaznija_ruka	snažnija/preferirana ruka tenisača	char(1)	
rang	tenisačev rang na službenoj ljestvici	int	
ozljeda	opis ozljede	varchar	

## Klub

Atribut	Opis	Tip podatka	Ograničenja
sifra_klub	jedinstvena šifra kluba	int	primarni ključ
naziv_klub	naziv kluba	varchar	jedinstvena vrijednost
godina_osnutka	godina osnutka kluba	int	
email	klupski email	varchar	
telefon	broj klupskog telefona	varchar	
post_broj	poštanski broj mjesta iz kojeg je klub	int	strani ključ

## Mjesto

Atribut	Opis	Tip podatka	Ograničenja
post_broj	poštanski broj	int	primarni ključ
naziv_mjesto	naziv mjesta	varchar	

## Trenira\_u

Atribut	Opis	Tip podatka	Ograničenja	
oib	jedinstveni identifikator trenera	char(11)	primarni ključ	strani ključ
sifra_klub	jedinstvena šifra kluba	int		strani ključ
od	datum od kada trener trenira u klubu	date	NotNull	
do	datum do kada je trener trenirao u klubu	date		

## Nastupa\_za

Atribut	Opis	Tip podatka	Ograničenja	
oib	jedinstveni identifikator tenisača	char(11)	primarni ključ	strani ključ
sifra_klub	jedinstvena šifra kluba	int		strani ključ
od	datum od kada nastupa za klub	date	NotNull	
do	datum do kada je nastupao za klub	date		



## Trening

Atribut	Opis	Tip podatka	Ograničenja
sifra_trening	jedinstvena šifra treninga	int	primarni ključ
datum_vrijeme_trening	datum i vrijeme treninga	timestamp	
trajanje	trajanje treninga	varchar	
opis	opis karakteristika treninga	varchar	
biljeske	bilješke s treninga	varchar	

## Trenira

Atribut	Opis	Tip podatka	Ograničenja
oib_tenisac	jedinstveni identifikator tenisača	char(11)	primarni ključ strani ključ
sifra_trening	jedinstvena šifra treninga	int	strani ključ
oib_trener	jedinstveni identifikator trenera	char(11)	strani ključ

## Transakcija

Atribut	Opis	Tip podatka	Ograničenja
sifra_transakcija	jedinstvena šifra transakcije	int	primarni ključ
datum_vrijeme_transakcija	datum i vrijeme transakcije	timestamp	NotNull
iznos	iznos transakcije	double	NotNull
nacin_placanja	način plaćanja transakcije	varchar	
opis	opis transakcije	varchar	
oib	jedinstveni identifikator osobe	char(11)	strani ključ
sifra_klub	jedinstvena šifra kluba	int	strani ključ

## Oprema

Atribut	Opis	Tip podatka	Ograničenja
sifra_oprema	jedinstvena šifra opreme	int	primarni ključ
naziv_oprema	naziv opreme	varchar	jedinstvena vrijednost
cijena	cijena opreme	double	

## Posjeduje

Atribut	Opis	Tip podatka	Ograničenja
sifra_klub	jedinstvena šifra kluba	int	primarni ključ strani ključ
sifra_oprema	jedinstvena šifra opreme	int	
dostupna_kolicina	dostupna količina opreme u klubu	int	

## Sastanak

Atribut	Opis	Tip podatka	Ograničenja
sifra_sastanak	jedinstvena šifra sastanka	int	primarni ključ
datum_vrijeme_sastanak	datum i vrijeme sastanka	timestamp	NotNull
dnevni_red	dnevni red sastanka	varchar	NotNull
biljeske	bilješke sa sastanka	varchar	
sifra_klub	jedinstvena šifra kluba	int	strani ključ

## Prisustvuje

Atribut	Opis	Tip podatka	Ograničenja	
oib	jedinstveni identifikator osobe	char(11)	primarni ključ	strani ključ
sifra_sastanak	jedinstvena šifra sastanka	int		strani ključ

## Podloga

Atribut	Opis	Tip podatka	Ograničenja
sifra_podloga	jedinstvena šifra podloge	int	primarni ključ
naziv_podloga	naziv podloge	varchar	

## Teren

Atribut	Opis	Tip podatka	Ograničenja
sifra_teren	jedinstvena šifra terena	int	primarni ključ
naziv_teren	naziv terena	varchar	
sifra_klub	jedinstvena šifra kluba	int	strani ključ
sifra_podloga	jedinstvena šifra podloge	int	strani ključ

## Kategorija

Atribut	Opis	Tip podatka	Ograničenja
sifra_kategorija	jedinstvena šifra kategorije	int	primarni ključ
tip	tip kategorije (singl ili parovi)	char(1)	NotNull
dobno_ogranicenje	dobno ograničenje	varchar	
spolno_ogranicenje	spolno ograničenje	char(1)	

## Turnir

Atribut	Opis	Tip podatka	Ograničenja
sifra_turnir	jedinstvena šifra turnira	int	primarni ključ
naziv_turnir	naziv turnira	varchar	jedinstvena vrijednost
sifra_klub	jedinstvena šifra kluba	int	strani ključ
sifra_kategorija	jedinstvena šifra kategorije	int	strani ključ

## Razina

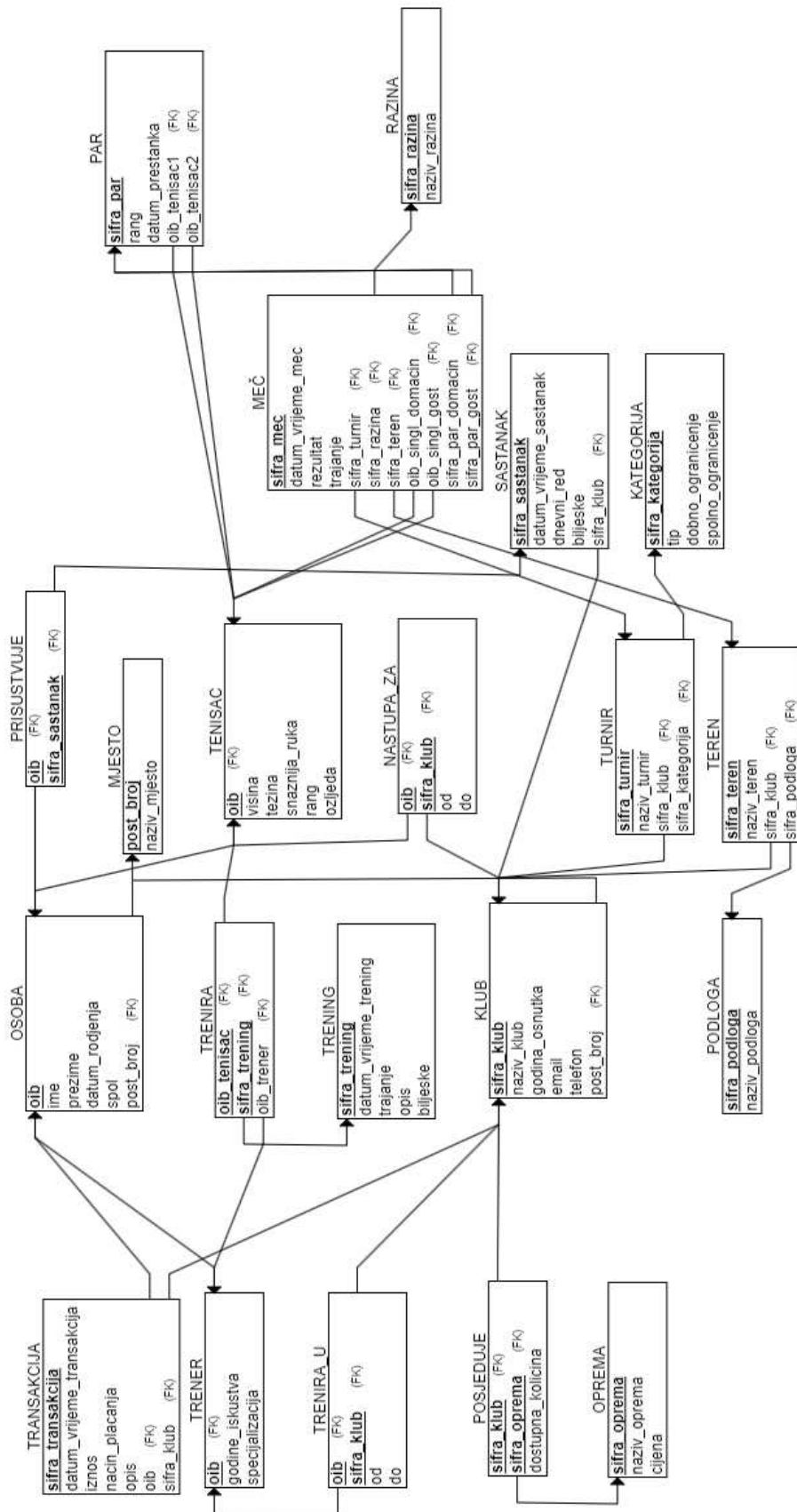
Atribut	Opis	Tip podatka	Ograničenja
sifra_razina	jedinstvena šifra razine	int	primarni ključ
naziv_razina	naziv razine	varchar	

## Par

Atribut	Opis	Tip podatka	Ograničenja
sifra_par	jedinstvena šifra para	int	primarni ključ
rang	rang para na službenoj ljestvici	int	
datum_prestanka	datum prestanka igranja para	date	
oib_tenisac1	jedinstveni identifikator prvog tenisača u paru	char(11)	strani ključ
oib_tenisac2	jedinstveni identifikator drugog tenisača u paru	char(11)	strani ključ

## Meč

Atribut	Opis	Tip podatka	Ograničenja
sifra_mec	jedinstvena šifra meča	int	primarni ključ
datum_vrijeme_mec	datum i vrijeme meča	timestamp	
rezultat	rezultat meča	varchar	
trajanje	trajanje meča	varchar	
sifra_turnir	jedinstvena šifra turnira	int	strani ključ
sifra_razina	jedinstvena šifra razine	int	strani ključ
sifra_teren	jedinstvena šifra terena	int	strani ključ
oib_singl_domacin	jedinstveni identifikator tenisača domaćina	char(11)	strani ključ
oib_singl_gost	jedinstveni identifikator tenisača gosta	char(11)	strani ključ
sifra_par_domacin	jedinstvena šifra para domaćina	int	strani ključ
sifra_par_gost	jedinstvena šifra para gosta	int	strani ključ



Slika 2-2 Relacijski dijagram baze podataka za teniski klub

### 3. Specifikacija programske potpore

#### 3.1. Funkcionalni zahtjevi

Programski sustav za vođenje evidencije rada teniskog kluba svoje funkcionalnosti definira određenim zahtjevima. Potrebno je korisniku omogućiti dodavanje podataka o klubu, njegovim igračima, trenerima, parovima i turnirima. Također, potrebno je omogućiti mijenjanje i brisanje postojećih podataka na zahtjev korisnika. U nastavku slijedi razrada svih funkcionalnih zahtjeva sustava, s prikazom na dijagramu slučaja uporabe na slici 3-1.

#### Aktori i njihovi funkcionalni zahtjevi:

1. Korisnik (inicijator) može:

- unositi nove podatke o teniskim klubovima, njihovim igračima, trenerima, parovima i turnirima
- izmjenjivati postojeće podatke o teniskim klubovima, njihovim igračima, trenerima, parovima i turnirima
- brisati postojeće podatke o teniskim klubovima, njihovim igračima, trenerima, parovima i turnirima

2. Baza podataka (sudionik):

- pohranjuje podatke o teniskim klubovima, njihovim igračima, trenerima, parovima i turnirima

#### Obrasci uporabe:

UC1 – Pregled teniskih klubova

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pregledavanje podataka o postojećim teniskim klubovima.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji barem jedan teniski klub u bazi podataka.
- **Rezultat:** Prikazani podaci za teniski klub.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz odgovarajućih podataka o teniskom klubu (općenito, transakcije, oprema, sastanci ili tereni).
  2. Otvara se stranica s podacima.

## UC2 – Pretraživanje teniskih klubova

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pretraživanje teniskih klubova po dostupnim kategorijama podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Prikaz teniskih klubova koji zadovoljavaju zadane kriterije.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategorije podataka po kojima obavlja pretraživanje.
  2. Korisnik unosi podatke za odabrane kategorije
  3. Prikazuje se popis filtriranih i/ili sortiranih teniskih klubova.

## UC3 – Unos podataka o teniskom klubu

- **Glavni sudionik:** Korisnik.
- **Cilj:** Dodavanje podataka o novom teniskom klubu u bazu podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Dodan novi zapis o teniskom klubu u bazu podataka.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za unos odgovarajućih podataka o teniskom klubu (općenito, transakcije, oprema, sastanci ili tereni).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    3. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    4. Korisnik unosi zahtijevane podatke ili odustaje



#### UC4 – Izmjena podataka o teniskom klubu

- **Glavni sudionik:** Korisnik.
- **Cilj:** Izmjena podataka o teniskom klubu.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski klub za kojeg se mijenjaju podaci.
- **Rezultat:** Izmijenjen zapis o teniskom klubu.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za izmjenu odgovarajućih podataka o teniskom klubu (općenito, transakcije, oprema, sastanci ili tereni).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC5 – Brisanje podataka o teniskom klubu

- **Glavni sudionik:** Korisnik.
- **Cilj:** Brisanje podataka o teniskom klubu.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski klub za kojeg se brišu podaci.
- **Rezultat:** Izbrisan zapis o teniskom klubu.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire odgovarajuće podatke o teniskom klubu (općenito, transakcije, oprema, sastanci ili tereni).

2. Korisnik odabire opciju za brisanje.
3. Nestaju svi podaci iz popisa vidljivih teniskih klubova.
4. Korisnik nastavlja ili prekida rad sa sustavom.

#### UC6 – Pregled tenisača

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pregledavanje podataka o postojećim tenisačima.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji barem jedan tenisač u bazi podataka.
- **Rezultat:** Prikazani podaci za tenisača.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz odgovarajućih podataka o tenisaču (općenito ili mečevi).
  2. Otvara se stranica s podacima.

#### UC7 – Pretraživanje tenisača

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pretraživanje tenisača po dostupnim kategorijama podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Prikaz tenisača koji zadovoljavaju zadane kriterije.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategorije podataka po kojima obavlja pretraživanje.
  2. Korisnik unosi podatke za odabrane kategorije
  3. Prikazuje se popis filtriranih i/ili sortiranih tenisača.

#### UC8 – Unos podataka o tenisaču

- **Glavni sudionik:** Korisnik.
- **Cilj:** Dodavanje podataka o novom tenisaču u bazu podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Dodan novi zapis o tenisaču u bazu podataka.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za unos odgovarajućih podataka o tenisaču (općenito ili mečevi).

2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
    1. Korisnik je unio invalidne podatke u određenu kategoriju
      1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
      2. Korisnik unosi ispravne podatke ili odustaje
    2. Korisnik nije unio potrebne podatke
      1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
      2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC9 – Izmjena podataka o tenisaču

- **Glavni sudionik:** Korisnik.
- **Cilj:** Izmjena podataka o tenisaču.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji tenisač za kojeg se mijenjaju podaci.
- **Rezultat:** Izmijenjen zapis o tenisaču.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za izmjenu odgovarajućih podataka o tenisaču (općenito ili mečevi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke

1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC10 – Brisanje podataka o tenisaču

- **Glavni sudionik:** Korisnik.
- **Cilj:** Brisanje podataka o tenisaču.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji tenisač za kojeg se brišu podaci.
- **Rezultat:** Izbrisan zapis o tenisaču.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire odgovarajuće podatke o tenisaču (općenito ili mečevi).
  2. Korisnik odabire opciju za brisanje.
  3. Nestaju svi podaci iz popisa vidljivih tenisača.
  4. Korisnik nastavlja ili prekida rad sa sustavom.

#### UC11 – Pregled trenera

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pregledavanje podataka o postojećim trenerima.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji barem jedan trener u bazi podataka.
- **Rezultat:** Prikazani podaci za trener.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz odgovarajućih podataka o treneru (općenito ili treninzi).
  2. Otvara se stranica s podacima.

#### UC12 – Pretraživanje trenera

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pretraživanje trenera po dostupnim kategorijama podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Prikaz trenera koji zadovoljavaju zadane kriterije.

- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategorije podataka po kojima obavlja pretraživanje.
  2. Korisnik unosi podatke za odabrane kategorije
  3. Prikazuje se popis filtriranih i/ili sortiranih trenera.

#### UC13 – Unos podataka o treneru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Dodavanje podataka o novom treneru u bazu podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Dodan novi zapis o treneru u bazu podataka.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za unos odgovarajućih podataka o treneru (općenito ili treninzi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC14 – Izmjena podataka o treneru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Izmjena podataka o treneru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji trener za kojeg se mijenjaju podaci.
- **Rezultat:** Izmijenjen zapis o treneru.

- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za izmjenu odgovarajućih podataka o treneru (općenito ili treninzi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC15 – Brisanje podataka o treneru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Brisanje podataka o treneru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji trener za kojeg se brišu podaci.
- **Rezultat:** Izbrisan zapis o treneru.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire odgovarajuće podatke o treneru (općenito ili treninzi).
  2. Korisnik odabire opciju za brisanje.
  3. Nestaju svi podaci iz popisa vidljivih trenera.
  4. Korisnik nastavlja ili prekida rad sa sustavom.

#### UC16 – Pregled teniskih parova

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pregledavanje podataka o postojećim teniskim parovima.
- **Sudionici:** Korisnik, baza podataka.

- **Preduvjeti:** Postoji barem jedan teniski par u bazi podataka.
- **Rezultat:** Prikazani podaci za teniski par.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz odgovarajućih podataka o teniskom paru (općenito ili mečevi).
  2. Otvara se stranica s podacima.

#### UC17 – Pretraživanje teniskih parova

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pretraživanje teniskih parova po dostupnim kategorijama podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Prikaz teniskih parova koji zadovoljavaju zadane kriterije.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategorije podataka po kojima obavlja pretraživanje.
  2. Korisnik unosi podatke za odabrane kategorije
  3. Prikazuje se popis filtriranih i/ili sortiranih teniskih parova.

#### UC18 – Unos podataka o teniskom paru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Dodavanje podataka o novom teniskom paru u bazu podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoje tenisači u bazi podataka koji čine novi teniski par.
- **Rezultat:** Dodan novi zapis o teniskom paru u bazu podataka.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za unos odgovarajućih podataka o teniskom paru (općenito ili mečevi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.

- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

UC19 – Izmjena podataka o teniskom paru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Izmjena podataka o teniskom paru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski par za kojeg se mijenjaju podaci.
- **Rezultat:** Izmijenjen zapis o teniskom paru.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za izmjenu odgovarajućih podataka o teniskom paru (općenito ili mečevi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje



#### UC20 – Brisanje podataka o teniskom paru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Brisanje podataka o teniskom paru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski par za kojeg se brišu podaci.
- **Rezultat:** Izbrisan zapis o teniskom paru.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire odgovarajuće podatke o teniskom paru (općenito ili mečevi).
  2. Korisnik odabire opciju za brisanje.
  3. Nestaju svi podaci iz popisa vidljivih teniskih parova.
  4. Korisnik nastavlja ili prekida rad sa sustavom.

#### UC21 – Pregled teniskih turnira

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pregledavanje podataka o postojećim teniskim turnirima.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji barem jedan teniski turnir u bazi podataka.
- **Rezultat:** Prikazani podaci za teniski turnir.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz odgovarajućih podataka o teniskom turniru (općenito ili mečevi).
  2. Otvara se stranica s podacima.

#### UC22 – Pretraživanje teniskih turnira

- **Glavni sudionik:** Korisnik.
- **Cilj:** Pretraživanje teniskih turnira po dostupnim kategorijama podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Prikaz teniskih turnira koji zadovoljavaju zadane kriterije.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategorije podataka po kojima obavlja pretraživanje.
  2. Korisnik unosi podatke za odabrane kategorije

3. Prikazuje se popis filtriranih i/ili sortiranih teniskih turnira.

#### UC23 – Unos podataka o turniru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Dodavanje podataka o novom turniru u bazu podataka.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji klub u bazi podataka koji organizira turnir.
- **Rezultat:** Dodan novi zapis o turniru u bazu podataka.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za unos odgovarajućih podataka o turniru (općenito ili mečevi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

#### UC24 – Izmjena podataka o turniru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Izmjena podataka o teniskom turniru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski turnir za kojeg se mijenjaju podaci.
- **Rezultat:** Izmijenjen zapis o teniskom turniru.

- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za izmjenu odgovarajućih podataka o teniskom turniru (općenito ili mečevi).
  2. Otvara se stranica gdje se unose podaci.
  3. Uz validne podatke, isti se spremaju u bazu te sustav o tome obavještava korisnika.
  4. Korisnik nastavlja ili prekida rad sa sustavom.
- **Opis mogućih odstupanja:**
  1. Korisnik je unio invalidne podatke u određenu kategoriju
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi ispravne podatke ili odustaje
  2. Korisnik nije unio potrebne podatke
    1. Pojavljuje se poruka o pogrešci i zahtjev za ponovnim unosom podataka
    2. Korisnik unosi zahtijevane podatke ili odustaje

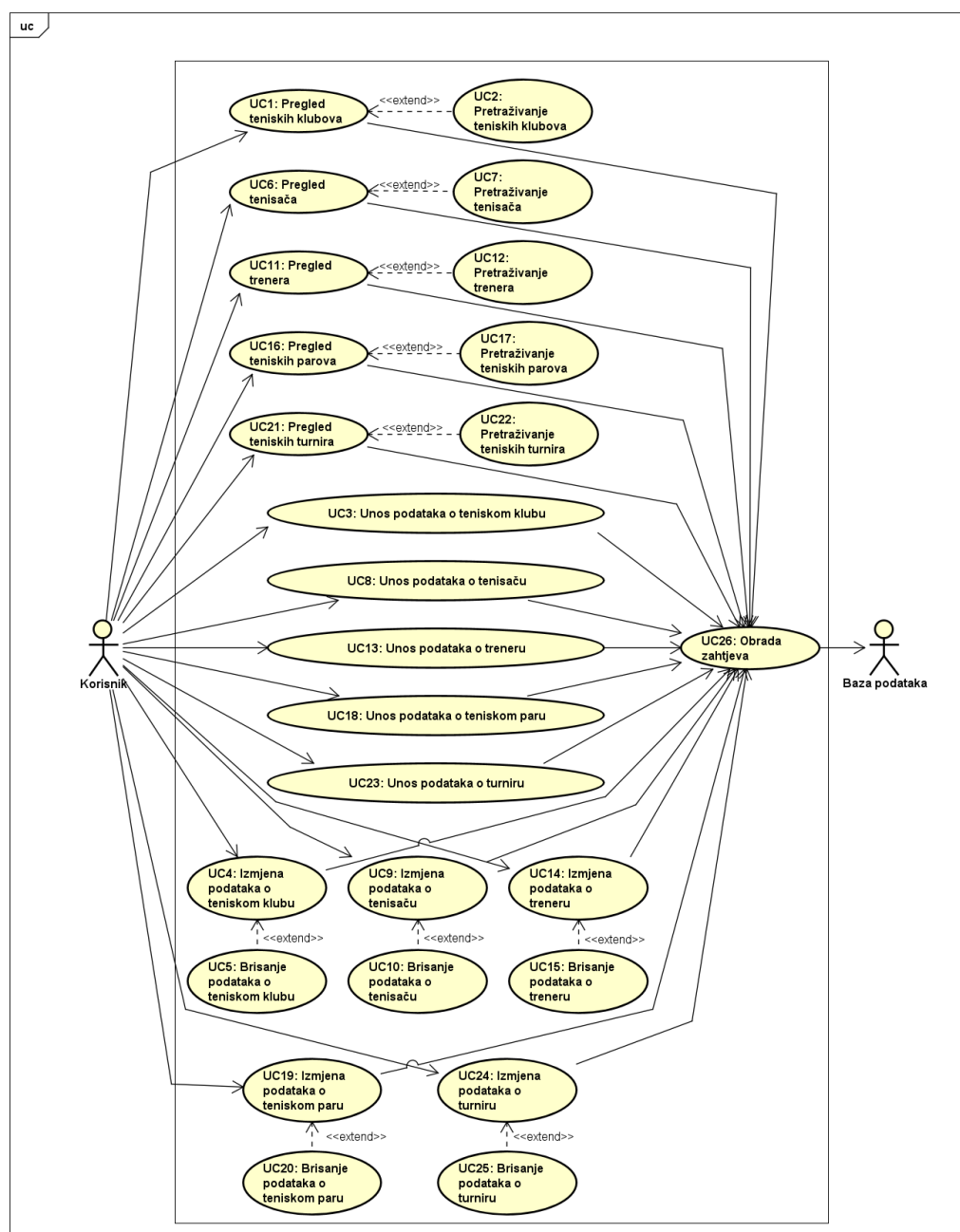
#### UC25 – Brisanje podataka o turniru

- **Glavni sudionik:** Korisnik.
- **Cilj:** Brisanje podataka o teniskom turniru.
- **Sudionici:** Korisnik, baza podataka.
- **Preduvjeti:** Postoji teniski turnir za kojeg se brišu podaci.
- **Rezultat:** Izbrisan zapis o teniskom turniru.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire odgovarajuće podatke o teniskom turniru (općenito ili mečevi).
  2. Korisnik odabire opciju za brisanje.
  3. Nestaju svi podaci iz popisa vidljivih teniskih turnira.
  4. Korisnik nastavlja ili prekida rad sa sustavom.

#### UC26 – Obrada zahtjeva

- **Glavni sudionik:** Korisnik.
- **Cilj:** Obrada zahtjeva prema i iz baze podataka.

- **Sudionici:** Baza podataka.
- **Rezultat:** Zahtjev prema ili iz baze podataka se obrađuje te se podaci dohvaćaju ili spremaju.
- **Opis osnovnog tijeka:**
  1. Korisnik zatraži podatke iz baze podataka ili pošalje podatke koje treba spremiti u bazu podataka.
  2. Pomoću aplikacijskog programskog sučelja (API) izvršava se prosljeđivanje podataka.
  3. Korisnik nastavlja ili prekida rad sa sustavom.



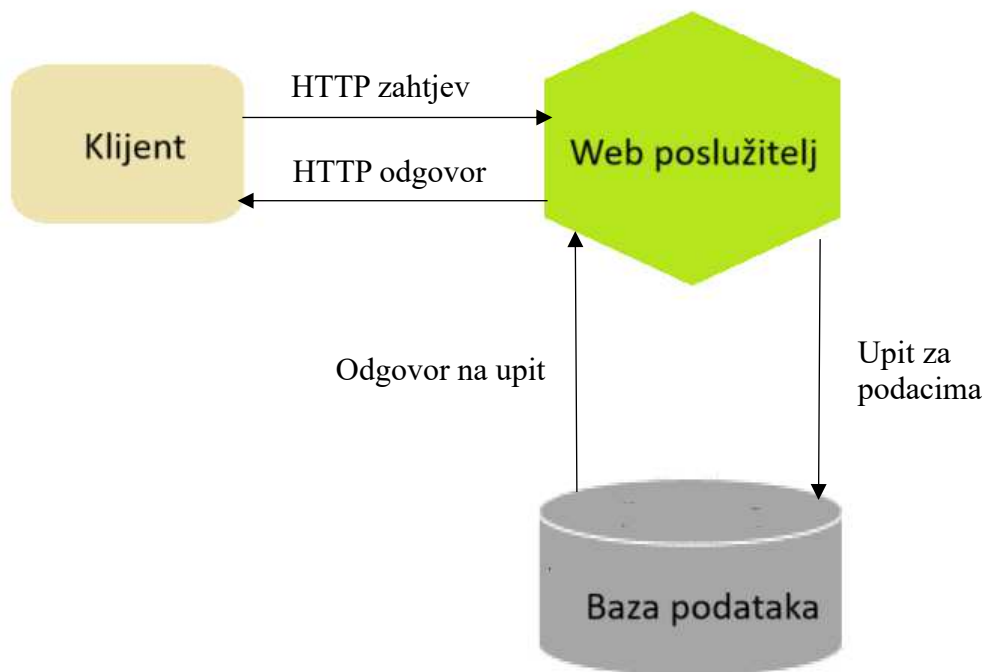
Slika 3-1 Dijagram obrasca uporabe

## 3.2. Arhitektura i dizajn sustava

Arhitektura je sustava apstraktni strukturirani opis sustava koji definira njegove komponente, njihove međusobne odnose i načina na koji međusobno komuniciraju. Za potrebe razvoja aplikacije za teniski klub korištena je MVC (*Model-View-Controller*) arhitektura.

Ta se arhitektura sastoji od tri glavne komponente: modela, pogleda i kontrolera. Model predstavlja podatke aplikacije i poslovnu logiku te upravlja podacima, pravilima i logikom koja definira način na koji se ti podaci obrađuju. Pogled prikuplja podatke iz modela i odgovoran je za prezentaciju tih podataka korisniku na razumljiv način. Konačno, kontroler predstavlja sponu između modela i pogleda na način da obrađuje korisničke unose, komunicira s modelom kako bi manipulirao podacima i odabire pogled za prikaz podataka.

Sustav će, stoga, funkcionirati kao što je prikazano na slici 3-2. Klijent će u većini slučajeva slati HTTP zahtjeve prema web poslužitelju. Tada će kontroler komponenta poslati zahtjev model komponenti za dohvat podataka, što zapravo predstavlja dohvat podataka iz baze podataka. Dohvaćeno će biti vraćeno na web poslužitelj, koji će uz pogled komponentu prezentirati isto klijentu.



Slika 3-2 Prikaz osnovnog rada sustava

### 3.3. Dijagrami razreda

Na slikama 3-3, 3-4 i 3-5 prikazani su dijagrami razreda koji pripadaju *backend* dijelu MVC arhitekture.

Na slici 3-3 prikazan je paket *controller*, u kojem je implementirana kontroler komponenta kroz razrede. Metode u tim razredima koriste DTO-ove (*Data Transfer Object*), kojima pristupamo putem metoda koje su implementirane u razredima paketa *entity*. Povratne su vrijednosti u JSON formatu s pripadajućim HTTP status kodovima.



Slika 3-3 Dijagram razreda paketa *controller*

Na slici 3-4 prikazan je paket *service*, a odnosi se na servisni sloj. U njemu je razvijena poslovna logika i pristup podacima za različite funkcionalnosti aplikacije. Razredi navedenog paketa koriste repozitorije za pristup podacima. Rezultati ovih servisa koriste se u razredima koji se nalaze u paketu *controller* kako bi se generirali odgovori na zahtjeve korisnika.



Slika 3-4 Dijagram razreda paketa *service*

Na slici 3-5 prikazan je paket *entity*, čiji razredi odražavaju strukturu baze podataka u aplikaciji. Metode unutar ovih razreda komuniciraju izravno s bazom podataka kako bi dohvatili željene podatke. Na dijagramu su ignorirani *getteri* i *setteri* za atribute.





## 4. Korisničke upute

### 4.1. Početni zaslon

Pokretanjem web aplikacije korisnika se pozicionira na početni zaslon (slika 4-1). Na njemu se nalazi navigacijska traka, koja se nalazi i na svakoj drugoj stranici aplikacije, i pozdravna poruka s kratkim opisom aplikacije. Pristup dijelovima aplikacije za daljnji rad omogućen je nakon klika na odgovarajuću poveznicu u navigacijskoj traci. Moguće je odabrati pregled i uređivanje podataka za kategorije: klubovi, treneri, tenisači, parovi i turniri. Unutar kategorije klubovi moguće je detaljnije promatrati podatke o klupskim transakcijama, sastancima, opremi i terenima. Kategorija treneri dodatno nudi opširnije podatke za treninge, dok su kategorije tenisači i parovi dodatno opisani podacima o odigranim mečevima za tenisača, odnosno par. Konačno, za svaki turnir može se detaljno vidjeti specifičnosti mečeva tog turnira. Sve navedene podatke moguće je na jednostavan način pregledavati, pretraživati, unositi nove te mijenjati i brisati postojeće.

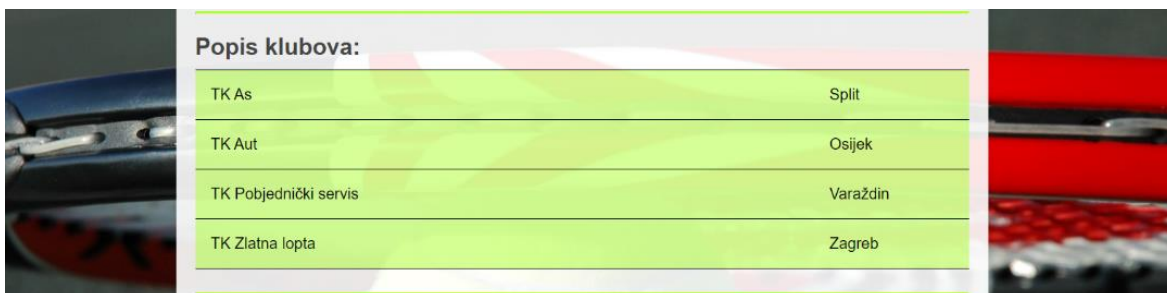


Slika 4-1 Početni zaslon

### 4.2. Pregled podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima

Podatke o teniskim klubovima, trenerima, tenisačima, parovima i turnirima moguće je pregledavati klikom na naziv kategorije na navigacijskoj traci. Time se otvara stranica na kojoj se, između ostalog, nalazi i rubrika *Popis klubova*, u slučaju u kojemu pregledavamo teniske klubove (slika 4-2). U navedenom se popisu pojavljuju svi dosad uneseni klubovi

zapisani unutar tablice, gdje s lijeve strane pišu imena klubova, a s desne imena gradova u kojima se nalaze.



Popis klubova:	
TK As	Split
TK Aut	Osijek
TK Pobjednički servis	Varaždin
TK Zlatna lopta	Zagreb

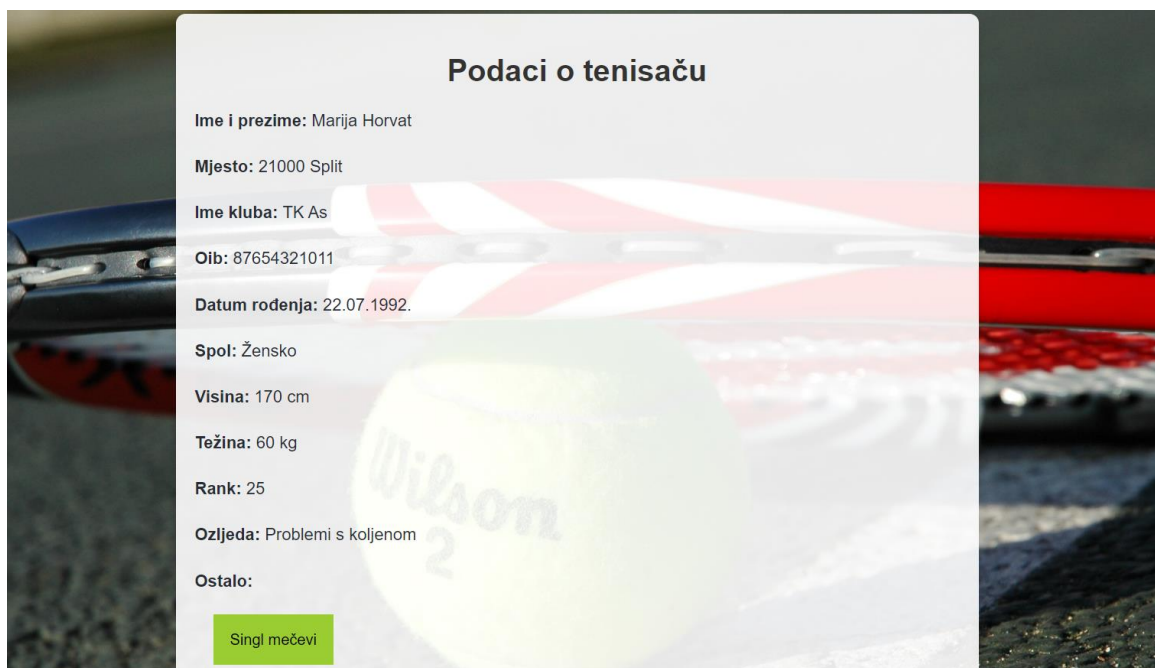
Slika 4-2 Prikaz teniskih klubova

Klikom na određenu stavku unutar tablice, otvorit će se nova stranica čija će glavina sadržaja biti detaljniji podaci o određenom entitetu, primjerice kao što je prikazano na slikama 4-3 i 4-4, gdje je moguće vidjeti podatke o treneru, to jest tenisaču. Unutar podataka postoji i polje *Ostalo*, u kojem se nalaze poveznice na stranice sa sadržajno povezanim skupinama podataka, kao što su u slučaju trenera treninzi koje je vodio ili u slučaju tenisača singl mečevi koje je odigrao. Pritom se klikom na te gumbe otvaraju stranice slične kao što je na slici 4-2.



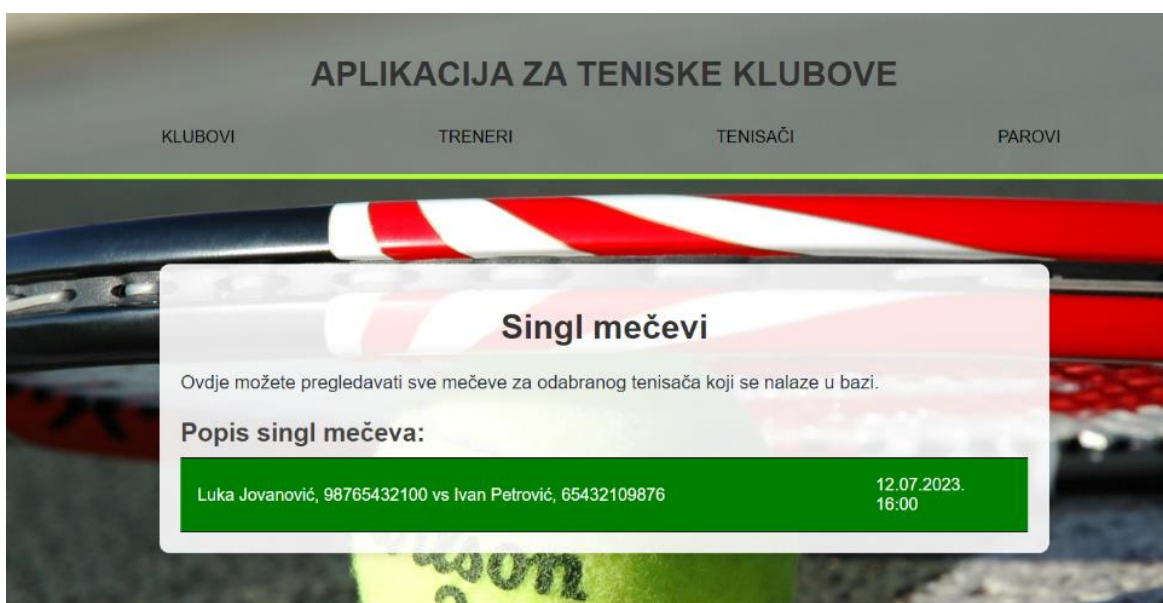
Podaci o treneru	
Ime i prezime:	Marko Petrić
Mjesto:	51000 Rijeka
Ime kluba:	TK Zlatna lopta
Oib:	34567890123
Datum rođenja:	01.11.1975.
Spol:	Muško
Specijalizacija:	Fizička priprema i kondicija
Godine iskustva:	20
Prijašnji klubovi:	
	1. TK As: 20. 3. 2012. - 11. 7. 2015.
Ostalo:	
	<a href="#">Treninzi</a>

Slika 4-3 Prikaz podataka o treneru



Slika 4-4 Prikaz podataka o tenisaču

Kod pregleda mečeva tenisača ili teniskog para zelenom su bojom označeni mečevi koje je tenisač, odnosno teniski par pobijedio, a crvenom su bojom označeni mečevi koje je izgubio (slika 4-5 i 4-6).



Slika 4-5 Prikaz dobivenog meča

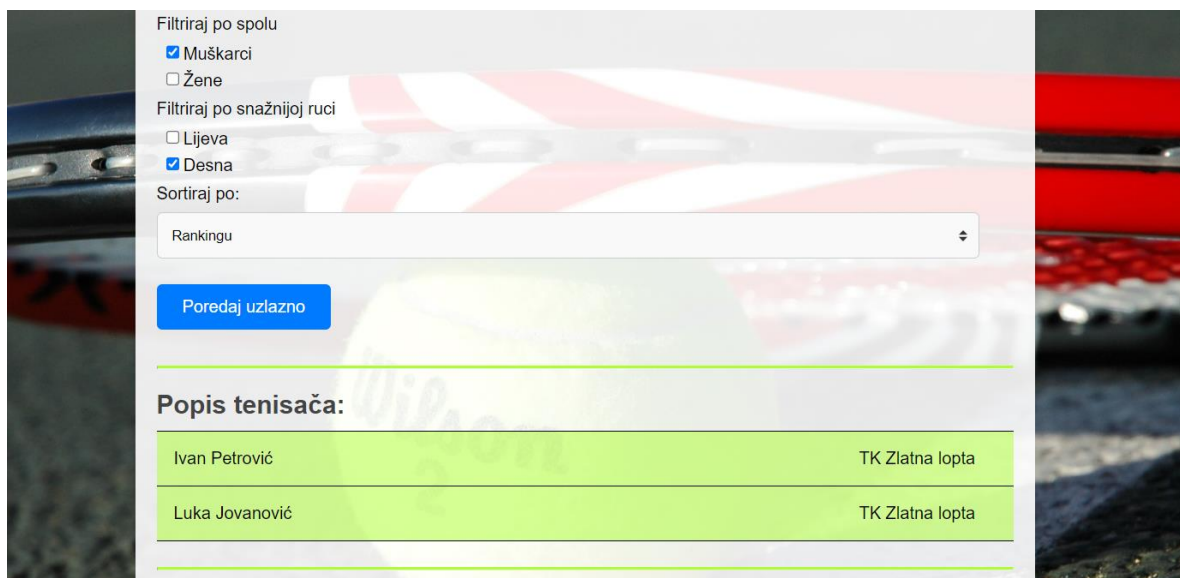


Slika 4-6 Prikaz izgubljenog meča

### 4.3. Pretraživanje podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima

Pretraživanje podataka odnosi se na mogućnosti filtriranja i sortiranja određenog skupa podataka. Navedene su opcije opazive ispred popisa svakog od entiteta koji se pregledava. Najprije su prikazana polja po kojima je moguće filtrirati podatke, a zatim polje u kojem se odabire po kojoj se varijabli sortira. Na primjeru tenisača (slika 4-7), neka od polja za filtriranje su spol i snažnija ruka, a moguće je sortiranje prema rangu. Klikom na gumb *Poredaj uzlazno* tenisači se poredaju uzlazno prema odabranoj varijabli te se tekst gumba mijenja u *Poredaj silazno*, čime se pak tenisači poredaju silazno prema odabranoj varijabli.





Slika 4-7 Filtriranje i sortiranje tenisača

#### 4.4.Unos podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima

Unos podataka za teniske klubove, trenere, tenisače, teniske parove i turnire moguće je ostvariti ispunjavajući formu koja se nalazi ispod popisa svakog od entiteta. Na slici 4-8 prikazana je prazna forma, čijim ispunjavanjem i klikom na *Dodaj sastanak* možemo dodati novi klupski sastanak u bazu podataka. Klikom na gumb *Odustani* ispunjena polja će se isprazniti.

**Dodajte novi sastanak:**

Datum i vrijeme sastanka

dd. mm. gggg. --:--

Dnevni red

Upišite dnevni red sastanka

Bilješke

Upišite bilješke sa sastanka

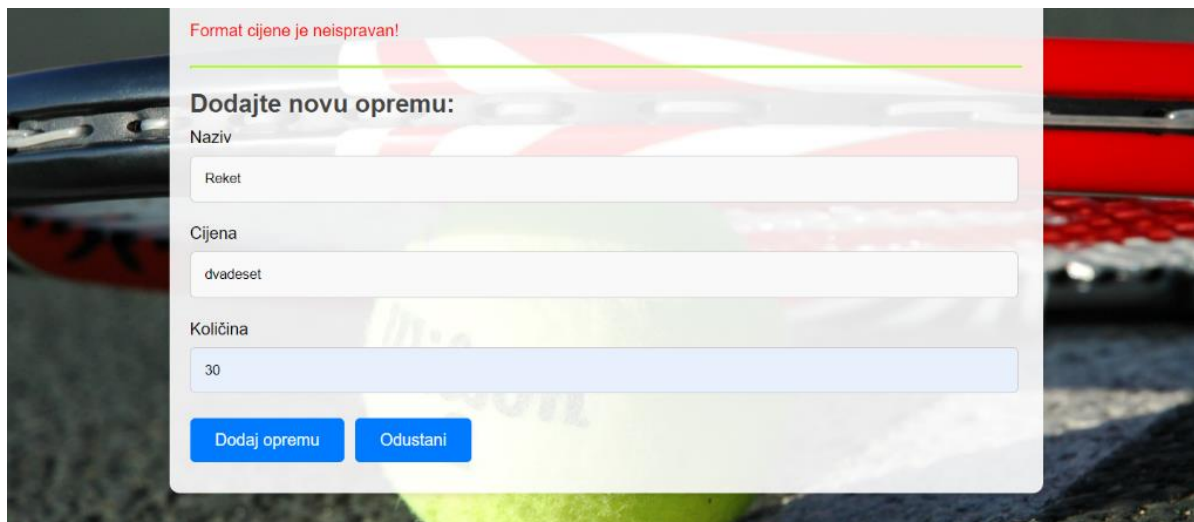
Sudionik 1

Izaberi osobu

Dodaj sudionika   Dodaj sastanak   Odustani

Slika 4-8 Forma za dodavanje novog sastanka

Važno je da se unose samo validni podaci, stoga postoji validacija unesenog koja će onemogućiti unos podataka koji nisu u skladu s očekivanim formatom (slika 4-9).



Format cijene je neispravan!

**Dodajte novu opremu:**

Naziv  
Reket

Cijena  
dvadeset

Količina  
30

Dodaj opremu Odustani

Slika 4-9 Poruka o grešci zbog neispravno unesenih podataka

#### 4.5. Izmjena i brisanje podataka o teniskim klubovima, trenerima, tenisačima, parovima i turnirima

Izmjenu i brisanje podataka moguće je ostvariti klikom na manji skup podataka koji želimo mijenjati. Ponovno se otvara stranica na kojoj se nalaze detaljniji podaci za odabrano, no polja za izmjenu i brisanje nalaze se ispod toga. Postoje dva gumba, *Promijeni podatke o paru* i *Izbriši podatke o paru* (u slučaju na slikama 4-10 i 4-11). Klikom na prvi ponovno se unose podatci zajedno s promjenom koja se želi napraviti, a klikom na drugi odabire se želja za brisanjem podataka. U oba slučaja, odabir se potvrđuje klikom na gumb *Potvrđi*.



**Uredi podatke:**

Promijeni podatke o paru  Izbriši podatke o paru

Igrač 1  
Izaberi prvog igrača u paru

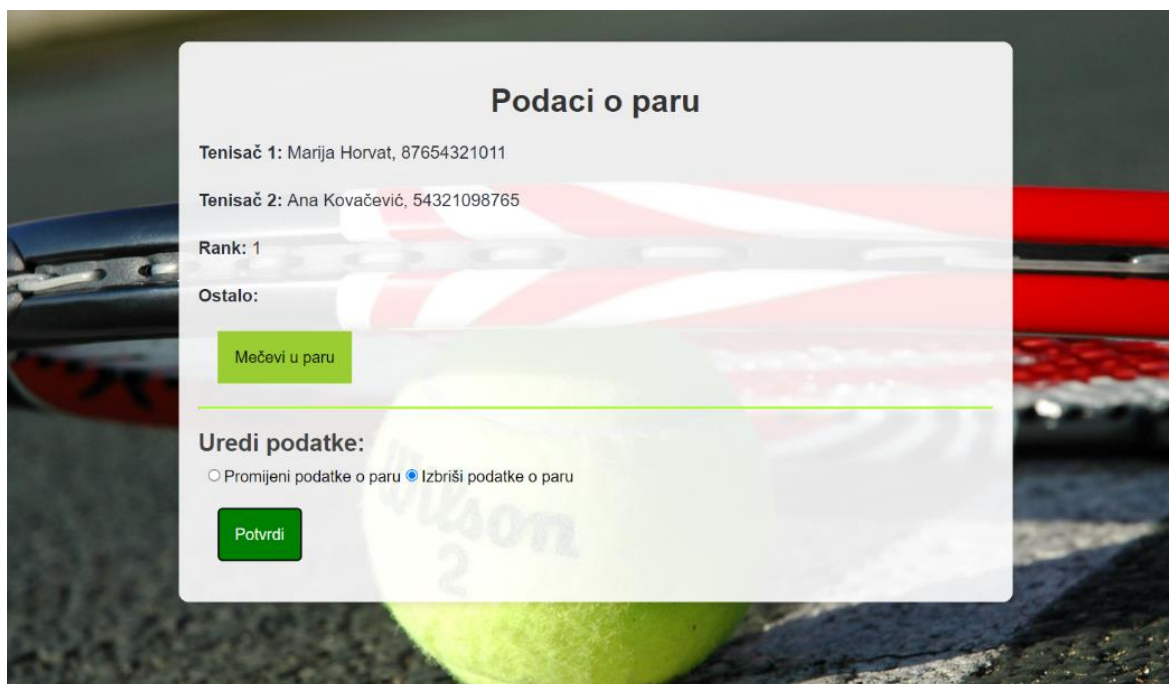
Igrač 2  
Izaberi drugog igrača u paru

Rank  
12

Datum prestanka djelovanja para  
10.07.2024.

Potvrđi

Slika 4-10 Izmjena podataka o paru



Slika 4-11 Brisanje podataka o paru

## Zaključak

Zadatak i cilj ovog je završnog rada bila izrada baze podataka i web aplikacije za teniski klub, što podrazumijeva mogućnost vođenja evidencije za teniski klub, njegove članove, aktivnosti i rezultate. Korištenjem u radu navedenih tehnologija isti je programski sustav ostvaren te je napisana odgovarajuća dokumentacija.

Razvoj se ovog programskog sustava odvijao u dvije veće etape: dizajniranje relacijske baze podataka i implementacija web aplikacije s grafičkim sučeljem temeljene na napravljenom modelu baze podataka. Aplikacija pritom pruža mogućnosti pregleda, unosa, brisanja i izmjene odgovarajućih podataka.

Daljnji rad na ovoj bi aplikaciji podrazumijevao detaljniju razradu baze podataka, gdje bi bilo moguće upravljanje podacima kao što su detaljnije statistike mečeva i slično. Nadalje, implementacija mogućnosti kategorizacije korisnika i pripadajuće autentifikacije doprinijela bi većoj sigurnosti i autentičnosti podataka. Konačno, daljnji razvoj može uključivati i mogućnost grafičkog prikaza podataka na svrsishodnim dijagramima.

Izrada ovog završnog rada dala mi je dodatno iskustvo pri izradi projekta koji obuhvaća oblikovanje i stvaranje programskog sustava. Proširio sam svoja znanja o bazama podataka, kao i pojačao vještine rada u radnim okvirima React i Spring Boot te programskim jezicima TypeScript i Java. Cjelokupni rad zasigurno će biti značajan poticaj za daljnje studiranje i rad te veliki čimbenik u svim daljnjim postignućima.



## Literatura

[1] Nastavni materijali iz kolegija Baze podataka, Fakultet elektrotehnike i računarstva Zagreb, veljača 2023.

[2] Nastavni materijali iz kolegija Programsko inženjerstvo, Fakultet elektrotehnike i računarstva Zagreb, 2023.

[3] Stack Overflow. Poveznica: <https://stackoverflow.com/>; pristupljeno 21.04.2024.

[4] Thorben Janssen. Poveznica: <https://thorben-janssen.com/>; pristupljeno 22.04.2024.

[5] Baeldung. Poveznica: <https://www.baeldung.com/>; pristupljeno 22.04.2024.

## Sažetak

U završnom su radu izrađena baza podataka i web aplikacija za teniski klub, koja obuhvaća pohranu podataka o klubu, njegovim članovima, njihovim aktivnostima i sportskim rezultatima.

Napravljeni su ER model i relacijski model podataka kako bi se implementirala baza podataka, za koju se koristi PostgreSQL. Kreirana je web aplikacija koja omogućuje unos, pregled, promjenu i brisanje teniskih klubova i njihovih podataka. U svim je slojevima aplikacije provedena validacija unesenih podataka. Korištene tehnologije za izradu aplikacije su radni okviri Spring Boot i React te programski jezici Java i TypeScript.

*Ključne riječi: teniski klub, baza podataka, web aplikacija, Spring Boot, React, TypeScript, PostgreSQL, Java*

## Summary

In this paper, a database and a web application for a tennis club were created, which includes the storage of data about the club, its members, their activities and sports results.

An ER model and a relational data model were created to implement the database, which uses PostgreSQL. A web application has been created that enables entering, viewing, changing and deleting tennis clubs and their data. Validation of entered data was carried out in all layers of the application. The technologies used to create the application are the Spring Boot and React frameworks and the Java and TypeScript programming languages.

*Keywords: tennis club, database, web application, Spring Boot, React, TypeScript, PostgreSQL, Java*