

Sustav praćenja rada službenih osoba u vaterpolu

Ljubić, Vlaho

Master's thesis / Diplomski rad

2025

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:168:016473>

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

Download date / Datum preuzimanja: **2025-03-17**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 732

**SUSTAV PRAĆENJA RADA SLUŽBENIH OSOBA U
VATERPOLU**

Vlaho Ljubić

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 732

**SUSTAV PRAĆENJA RADA SLUŽBENIH OSOBA U
VATERPOLU**

Vlaho Ljubić

Zagreb, veljača 2025.

DIPLOMSKI ZADATAK br. 732

Pristupnik: **Vlaho Ljubić (0036503954)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: prof. dr. sc. Vedran Mornar

Zadatak: **Sustav praćenja rada službenih osoba u vaterpolu**

Opis zadatka:

Izgraditi administracijski sustav koji će omogućiti efikasno vođenje vaterpolskih natjecanja i praćenje rada službenih osoba. Sustav će omogućiti automatsko generiranje ključnih dokumenata, poput delegiranja, disciplinskih prijava i izvještaja o nastupima klubova po natjecanjima te njihovu javnu dostupnost. Digitalizirat će se proces potvrde, otkazivanja i javljanja dostupnosti za utakmice radi lakše komunikacije između službenih osoba i organizatora natjecanja. Također, aplikacija će pratiti ocjene, komentare i angažman službenih osoba, što će omogućiti kvalitetniju evidenciju njihovog rada. Ovaj sustav doprinosi povećanju učinkovitosti, transparentnosti i profesionalizacije u organizaciji vaterpolskih natjecanja. Sustav će se razviti kao web aplikacija koristeći programski okvir Vue.js na klijentu i ASP.NET Core na poslužitelju. Koristit će se sustav upravljanja bazom podataka PostgreSQL.

Rok za predaju rada: 14. veljače 2025.

Sadržaj

Uvod	1
1. Opis sustava.....	2
2. Korištene tehnologije.....	4
2.1. Vue.js.....	4
2.2. ASP.NET Core	4
2.3. Entity Framework Core	5
2.4. PostgreSQL.....	5
3. Arhitektura aplikacije	6
3.1. Klijent-poslužitelj arhitektura.....	6
3.2. REST API komunikacija	7
3.3. Entiteti	13
3.3.1. Korisnik	14
3.3.2. Utakmica.....	14
3.3.3. Natjecanje	14
3.3.4. Kategorija	14
3.3.5. Sezona.....	14
3.3.6. Kolo	14
3.3.7. Klub	14
3.3.8. Bazen	14
3.3.9. Regija.....	15
3.3.10. Sudačka lista.....	15
3.3.11. Status službenih osoba.....	15
3.3.12. Dostupnost službenih osoba	15
3.3.13. Evaluacija	15
3.3.14. Disciplinska prijava	15

3.4.	Klijentska aplikacija	16
3.4.1.	Pinia	16
3.4.2.	Provjera pristupa.....	18
3.4.3.	Korisničke poruke	20
3.4.4.	API pozivi.....	20
3.4.5.	JWT	22
3.5.	Poslužiteljska aplikacija	23
3.5.1.	Modeli.....	24
3.5.2.	Kontroleri	25
3.5.3.	DTO	26
3.5.4.	Registracija korisnika	27
3.5.5.	Prijava korisnika.....	27
4.	Korisničko sučelje	29
4.1.	Prijava u sustav	29
4.2.	Kalendar.....	29
4.3.	Službene osobe	31
4.4.	Dodavanje ili ažuriranje utakmice.....	33
4.5.	Popis utakmica.....	34
4.6.	Bazeni	35
4.7.	Korisnici	35
4.8.	Sezone.....	36
4.9.	Klubovi.....	36
4.10.	Klupska statistika sudjelovanja	37
4.11.	Sudačke liste	37
4.12.	Natjecanja	38
4.13.	Profil	38

4.14.	Delegiranja.....	39
4.15.	Slanje delegiranja	40
4.16.	Detalji o utakmici	41
4.17.	Evaluacije	42
4.18.	Disciplinske prijave	43
4.19.	Javljanje nedostupnosti.....	43
4.20.	Klupska funkcionalnost	44
5.	Ponašanje sustava	45
5.1.	Obrasci uporabe.....	46
	Zaključak	60
	Literatura	61
	Sažetak.....	62
	Summary.....	63

Uvod

U današnjem svijetu ubrzanog života gdje sve želimo odmah i na dlanu, moderne tehnologije nam uvelike omogućavaju unaprjeđenje raznih životnih aspekata. Najtromiji prilagodbi novom načinu života su administrativni poslovi – papirnati dokumenti i ručno upravljanje podacima samo su neki od primjera poslova koji zaostaju u praćenju digitalnih inovacija.

Pri vođenju jednog vaterpolskog natjecanja i praćenja rada službenih osoba ključna je učinkovita administracija kako bi se natjecanje odvijalo nesmetano i bez poteškoća te kako bi koordinacija službenih osoba bila efikasna i pravovremena.

Cilj ovog diplomskog rada je razvoj web-aplikacije koja služi kao administracijski sustav koji omogućuje efikasno vođenje vaterpolskog natjecanja i praćenja rada službenih osoba. Sustav omogućuje digitalizaciju i automatizaciju ključnih procesa za efikasno provođenje natjecanja, poput javljanja dostupnosti za natjecanja, delegiranja službenih osoba i kreiranje dokumenata za utakmicu. Osim toga omogućuje praćenje angažmana i ocjena sudaca, što doprinosi objektivnijoj evaluaciji njihovog rada.

Klijentska strana aplikacije razvijena je u Vue.js JavaScript okviru, dok poslužiteljska strana koristi ASP.NET Core. Pohrana podataka se vrši preko PostgreSQL, sustava za upravljanje bazama podataka.

1. Opis sustava

Fokus svake sportske priče uvijek bi trebao biti na igračima i igri, neovisno o kojem sportu se radi. Nažalost, često to nije tako te se nakon odigrane utakmice često priča o sucima. Njihovom dobro, ili manje dobro odrađenom poslu. Ovaj sustav bavi se upravo tom sferom jedne sportske priče - radom službenih osoba. Konkretno, praćenjem rada službenih osoba i organizacijom te provođenjem vaterpolskog natjecanja.

Provođenje natjecanja zahtjeva niz administrativnih zadataka i međusobnu koordinaciju dionika različitih procesa tijekom natjecanja za pojedinu utakmicu. Natjecanje kreće pripremom kalendara odigravanja utakmica. Ključno je da se kalendar utvrdi na vrijeme i da za vrijeme trajanja natjecanja ima što manje promjena i otkazivanja čime se osigurava pouzdanost i mogućnost pravovremene pripreme raznih procesa ključnih za utakmicu. Nakon kreiranja inicijalnog rasporeda, sljedeći korak iz kola u kolo je prikupljanje podataka o satnici utakmice, kako bi se moglo pripremiti delegiranje službenih osoba i dogovoriti televizijski prijenosi. Nakon saznavanja satnica svih utakmica u datom kolu, slijedi proces delegiranja službenih osoba. Povjerenik natjecanja, vodeći računa o težini utakmice, mjestu odigravanja, protivnicima i brojnim drugim faktorima, određuje službene osobe, suce i delegata, za svaku utakmicu. Da bi znao jesu li službene osobe u mogućnosti obavljati dužnost u tom kolu, mora ih kontaktirati. Nakon složenog rasporeda i definiranih službenih osoba povjerenik isto delegiranje mora javno objaviti i obavijestiti sve dionike, čime je završen prvi dio priče. Drugi dio priče odvija se nakon utakmice. Po završetku utakmice, delegat ima dužnost ispuniti evaluaciju za suce predmetne utakmice. Također su dužni, ako je bilo nekih nesportskih događanja, ispuniti disciplinske prijave, kako bi se osobe koje su napravile prekršaj na utakmici sankcioniralo. Ispunjavanjem evaluacija i eventualnih disciplinskih prijava te slanjem dokumentacije povjereniku završava se proces za jednu utakmicu te se u proces kreće ispočetka. Povjerenik potom prikuplja navedene dokumente i izrađuje sudačku statistiku koja mu pomaže pri sljedećem delegiranju i odabiru službenih osoba.

Sustav koji je izgrađen za potrebe ovog diplomskog rada dotiče se svih dionika gore navedenog procesa. Naglasak i primarna funkcionalnost sustava stavljena je na suce i pogled

na funkcioniranje sustava kroz sudačku perspektivu, uz potporu drugih dionika - delegata te povjerenika natjecanja i predstavnike klubova. Iz sudačke perspektive, procesi koje oni obavljaju su javljanje nedostupnosti za utakmice na određeni datum, potvrda ili naknadno otkazivanje već delegirane utakmice, te ispunjavanje disciplinskih prijava i pregled evaluacija i statistike suđenja. Delegati svoju ulogu ostvaruju kroz ispunjanje evaluacija, te također javljanje nedostupnosti i potvrđivanje ili otkazivanje utakmica. Kroz sustav se također ostvaruje i informatizacija načina prijave satnice domaće utakmice određenog kluba. U sustavu su uvedene klupske osobe koje imaju uvid u svoje domaće utakmice i mogućnost prijave satnice za odabranu utakmicu. Naposljetku, administrator sustava - povjerenik natjecanja - ima potpuni uvid u službene osobe, informacije o dostupnosti i statistiku suđenja, raspored utakmica s prijavljenom satnicom, pregled evaluacija i disciplinskih prijava te proces slanja digitalnog delegiranja kreiranog iz sustava, bez potrebe za ručnim pisanjem dokumenta.

Povjerenik sada u sustavu ima pregled cijelog natjecanja, lako dostupan i pregledan prikaz pojedinih utakmica. Pritom za svako kolo ne mora kontaktirati svaki pojedini klub domaćin za dostavom satnice, već je klupskim osobama omogućena samostalna prijava satnice čime se smanjuje količina administrativnih mailova i gubljenje vremena. Povjerenik također ima pregled svih službenih osoba i njihove dostupnosti za točno određene dane i utakmice te ih nije potrebno provjeravati ponaosob. Generiranje dokumenta delegiranja iz sustava iznimno olakšava i ubrzava cijeli proces, smanjuje mogućnost ljudske pogreške pri pisanju svake informacije iznova te automatizira cijeli proces. Proces evaluacije sudaca i transparentnost pri prikazu i obradi tih podataka također je veliki napredak pri praćenju rada službenih osoba čime se unaprjeđuje kvaliteta i profesionalnost svih dionika.

Uvođenjem informatizacije i digitaliziranjem navedenih procesa značajno se unaprjeđuje dosadašnje procese u vidu automatizacije, transparentnosti, brzini i preciznosti komuniciranja i obrade podataka te podizanjem ukupne kvalitete praćenja i valoriziranja rada službenih osoba i vođenja vaterpolskog natjecanja.

2. Korištene tehnologije

2.1. Vue.js

Vue.js je progresivni JavaScript okvir za izradu korisničkih sučelja. Omogućava modularni i reaktivni razvoj aplikacija, što znači da se promjene u podacima automatski odražavaju na korisničkom sučelju bez potrebe za ručnim ažuriranjem. Vue.js koristi virtualni Dom (eng. Document Object Model) za optimizaciju performansi te omogućava deklarativno definiranje korisničkog sučelja koristeći web tehnologije kao što su HTML, CSS i JavaScript.

Vue.js također ima pozamašnu lepezu dodataka, među kojima su Vue Router koji služi za upravljanje rutama i Pinia za upravljanje stanjem aplikacije. Zbog svoje fleksibilnosti jednostavan je za integraciju s već postojećim projektima ili drugim JavaScript bibliotekama.

Izabrao sam ga zbog jednostavnosti korištenja, fleksibilnosti i mogućnošću integracije s drugim alatima. Filozofija grupiranja koda u građevne jedinice – komponente – omogućava ponovnu upotrebu koda i lakšu organizaciju aplikacije što je ključno za održavanje i kasniju naknadnu proširivost. Također okuplja veliku zajednicu korisnika što omogućava aktivni razvoj i dugoročnu podršku i stabilnost. [1]

2.2. ASP.NET Core

ASP.NET Core je moderan, visokoučinkoviti, multiplatformski okvir za razvoj web aplikacija i API-ja, kojeg je razvio Microsoft. Omogućava izgradnju sigurnih, skalabilnih i fleksibilnih aplikacija koje mogu raditi na Windows, Linux i macOS operacijskim sustavima. ASP.NET Core je nadogradnja na stariju verziju ASP.NET-a s modularnom arhitekturom i poboljšanim performansama.

Jedna od ključnih prednosti je što okvir omogućava jednostavnu konfiguraciju posredničkih komponenti, čime se postiže fleksibilnost i učinkovitost u obradi HTTP zahtjeva. Uz razvoj web aplikacija koju ovaj okvir pruža, koristio sam funkcionalnost stvaranja API točaka koristeći arhitekturu kontrolera i modela uz poštivanje REST principa. Osim izrade API točaka, ASP.NET Core pruža ugrađenu podršku za autentifikaciju i autorizaciju koristeći

JWT – JSON Web Token tehnologiju. Pri radu s bazom podataka koristi Entity Framework Core [2].

ASP.NET Core sam odabrao zbog njegovih performansi, multiplatformske podrške i fleksibilnosti u razvoju API-ja. Kao i Vue.js, ASP.NET Core okuplja veliku zajednicu korisnika koji, uz Microsoftovu podršku, garantiraju dugoročnu stabilnost i jednostavnu nadogradnju aplikacije.

2.3. Entity Framework Core

Entity Framework Core je objektno-relacijski mapirajući alat koji omogućava rad s bazom podataka koristeći C# jezik. On zamjenjuje pisanje upita prema bazi koristeći SQL jezik, čime olakšava integraciju baze podataka s ASP.NET Core aplikacijom, omogućavajući rad s podacima u objektnom okruženju, na apstraktan i intuitivan način.

Upiti nad bazom obavljaju se s pomoću C# LINQ (eng. Language Integrated Query) izraza, što omogućava fleksibilnu i čitljivu manipulaciju podacima bez potrebe za „čistim“ SQL upitima. EF Core podržava različite strategije pristupa podacima, poput Code-First i Database-First pristupa, uz jednostavno mapiranje klasa u tablice baze podataka. Također omogućava upravljanje verzijama baze podataka i promjenama u strukturi baze [3].

Dodatna prednost EF Core alata je multiplatformska podrška koja omogućava rad s različitim bazama podataka kao što su PostgreSQL, SQL Server, MySQL i SQLite.

Odabrao sam Entity Framework Core zbog jednostavne integracije s ASP.NET Core poslužiteljskom aplikacijom i korištenja LINQ izraza za brže i intuitivnije baratanje s podacima.

2.4. PostgreSQL

PostgreSQL je sustav za upravljanje relacijskim bazama podataka koji nudi visoku pouzdanost, sigurnost i performanse. Ima podršku za transakcije, indeksiranje, replikaciju i naprednije SQL značajke. Ističe se osiguravanjem integriteta podataka kroz transakcije, radu s NoSQL podacima, naprednim indeksima za optimiziranje pretrage [4].

Odabrao sam PostgreSQL zbog stabilnosti, visokih performansi i dobre integracije s ASP.NET Core koja pospješuje efikasnost upravljanja podacima unutar aplikacije.

3. Arhitektura aplikacije

Web aplikacija kao administracijski sustav za praćenje rada službenih osoba u vaterpolu se sastoji od klijentske i poslužiteljske aplikacije koje komuniciraju REST API pozivima, te baze podataka na koju se spaja poslužiteljska aplikacija (Slika 3.1).



Slika 3.1 Klijent-poslužitelj arhitektura

3.1. Klijent-poslužitelj arhitektura

Klijent-poslužitelj arhitektura je naziv koji se koristi za način strukturiranja komponenti u raspodijeljenim sustavima. To je najčešće korišten model pri razvoju softverskih rješenja u web aplikacijama. U ovoj arhitekturi sustav dijelimo na dvije komponente: klijent, koji šalje zahtjeve za resursima i prima odgovore, i poslužitelj, komponenta koja obrađuje te zahtjeve, pristupa resursima unutar baze podataka i vraća dohvaćene podatke klijentu [9].

Klijent-poslužitelj arhitektura koristi se u raspodijeljenim sustavima kako bi se jasno odvojili arhitekturni slojevi. Tipična arhitektura aplikacije sastoji se od tri sloja - sloja prezentacije, sloja aplikacijske logike i sloja podataka.

U dvoslojnoj klijent-poslužitelj arhitekturi funkcionalni slojevi grupirani su u dva zasebna sloja. To su klijentska aplikacija koja sadrži sloj prezentacije i aplikacijske logike, dok poslužiteljska aplikacija sadrži sloj podataka te mogućnost pružanja usluga jednoj ili više klijentskih aplikacija. Klijentski se sloj može dodatno raščlaniti pa umjesto gore opisanog klijenta koji sadrži sloj prezentacije i aplikacijske logike i naziva se debeli klijent, imamo tanki klijent koji sadrži samo sloj prezentacije. Time dobivamo zapravo troslojnu arhitekturu gdje su svi funkcionalni slojevi grupirani u zasebne arhitekturne slojeve pa imamo klijentsku

aplikaciju koja sadrži sloj prezentacije, poslužiteljsku aplikaciju koja obrađuje sloj aplikacijske logike i bazu podataka koja služi kao sloj za pristup i dohvat podataka. Ovu troslojnu klijent-poslužitelj arhitekturu koristio sam u ovom radu.

Za razliku od klijent-poslužitelj arhitekture postoje i drugi tipovi arhitektura. Jedna od njih je monolitna arhitektura koja sve funkcionalne slojeve aplikacije sadrži unutar jedne strukture što u početku olakšava razvoj radi jednostavnosti, ali u kasnijim fazama razvoja otežava skaliranje i održavanje. Za razliku od dvoslojne ili troslojne arhitekture, postoje i višeslojne arhitekture koje sadrže višestruke aplikacijske poslužitelje i omogućuju ravnomjerniju raspodjelu opterećenja ali i zahtijevaju veću komunikacijsku infrastrukturu. Također postoje i arhitekture u kojima svaka komponenta može imati ulogu i klijenta i poslužitelja, ne postoje centralizirani poslužitelji. To su Peer-to-Peer (eng. P2P) arhitekture.

Ono što odlikuje klijent-poslužitelj arhitekturu je centralizacija podataka, gdje su svi podaci spremljeni na jednom mjestu - u bazi podataka - čime se olakšava upravljanje pristupom i dostupnost podataka. S obzirom na raspodijeljenost i dobru odvojenost slojeva, arhitektura je skalabilna te je moguće povećavati broj klijenata koji pristupaju resursima preko poslužiteljske aplikacije bazi podataka.

Odabrao sam klijent-poslužitelj arhitekturu jer se pokazala kao najbolji izbor za sustav koji je trebalo izgraditi gdje su jasno podijeljeni funkcionalni slojevi u klijentskoj aplikaciji koja obuhvaća prezentacijski sloj, poslužiteljskoj aplikaciji koja obrađuje aplikacijsku logiku te centralizirano mjesto za spremanje i upravljanje podacima - bazi podataka.

3.2. REST API komunikacija

API (eng. application programming interface) je naziv za skup pravila i definicija kojima se omogućuje komunikacija između različitih aplikacija. API djeluje kao posrednik u komunikaciji dviju aplikacija koje pritom ne znaju ništa o arhitekturi ili implementaciji one druge.

REST API (eng. Representational State Transfer) je jedan od najčešće korištenih načina komunikacije između klijentskih i poslužiteljskih aplikacija u raspodijeljenim sustavima. To je skup arhitektonskih pravila i ograničenja koji se temelji na standardnim HTTP (eng. Hyper Text Transfer Protocol) metodama za manipulaciju resursima. Kada kažemo da je klijent napravio REST API zahtjev, to znači da se stanje resursa prenijelo onom koji je taj zahtjev izvršio. Taj prijenos se može izvršiti u nekom od brojnih formata: JSON, HTML, XML itd. Metode koje se koriste u HTTP-u su GET (dohvaćanje podataka), POST (kreiranje novog resursa), PUT (ažuriranje postojećeg resursa), PATCH (djelomično ažuriranje resursa) i DELETE (brisanje resursa).

Da bi se API smatrao REST mora zadovoljavati određene uvjete. Arhitektura mora biti klijent-poslužitelj i zahtjevi i odgovori moraju biti obrađeni preko HTTP. Komunikacija između klijenta i poslužitelja mora biti bez pamćenja stanja. Svaki zahtjev za resursom je odvojen i nepovezan s prethodnim i budućim. Odgovori poslužitelja mogu se pospremiti (eng. caching) kako bi se smanjilo opterećenje pri ponavljajućem pozivanju istog resursa te kako bi se povećala brzina rada aplikacije. Svakom resursu pristupa se putem jedinstvene adrese - jedinstveni identifikator resursa (eng. URI - Uniform Resource Identifier) [10].

Prednosti korištenja REST API arhitekture su jednostavnost zbog korištenja dobro poznatih i definiranih protokola i struktura podataka (HTTP i JSON, XML), neovisnost klijenta i poslužitelja te interoperabilnost u korištenju različitih tehnologija i programskih jezika.

Za komunikaciju između klijentske i poslužiteljske aplikacije u ovom radu koristio sam REST API. Na svaki korisnikov zahtjev za podacima unutar klijentske aplikacije API-jem se šalje REST zahtjev. Ako korisnik mijenja određeni podatak, te ga želi spremiti u bazu podataka šalje se zahtjev PUT, dok pri kreiranju novog resursa klijentska aplikacija šalje zahtjev POST. Korisnik određene podatke može i izbrisati iz sustava. Tada se koristi zahtjev DELETE. U nastavku su prikazani konkretni slučajevi unutar klijentske i poslužiteljske aplikacije za manipulaciju podacima. U kodu (Kod 3.1) vidljiv je odsječak za dohvat podataka pri učitavanju stranice za prikaz svih klubova.


```

onMounted(async () => {
  try {
    const response = await api.get('/club/allClubs')
    if (response.data.success) {
      clubs.value = response.data.data
      dataReady.value = true
    } else {
      toast.error(response.data.error)
    }
  } catch (error) {
    toast.error(error.message);
  }
})

```

Kod 3.1 Zahtjev za podacima o svim klubovima

Sljedeći kod (Kod 3.2) prikazuje REST poziv za ažuriranje osobnih podataka korisnika. Podaci koji se šalju poslužiteljskoj aplikaciji za ažuriranje predaju se kroz tijelo zahtjeva.

```

async function updateMyData() {
  if (validateForm()) {
    let userData = {
      Name: nameInput.value,
      Surname: surnameInput.value,
      Email: emailInput.value,
      Password: passwordInput.value
    };
    try {
      const response = await
api.put('/user/updateProfile/' + store.getUserId, userData)
      if (response.data.success)
        toast.success("Uspješno ažurirani podaci")
    } catch (error) {
      toast.error(error.message);
    }
  } else {
    toast.warning("Unesite sve potrebne podatke")
  }
}

```

Kod 3.2 Zahtjev za ažuriranje osobnij podataka korisnika

U kodu (Kod 3.3) je vidljiv poziv kojim se želi obrisati određeni bazen u bazi podataka.

```
async function deletePool(poolId) {
    if (window.confirm("Jeste li sigurni da želite izbrisati
bazen?")) {
        try {
            const response = await
api.delete('/pool/deletePool/' + poolId)
            if (response.data.success) {
                pools.value = response.data.data
                poolNameInput.value = ""
                toast.success("Uspješno ste izbrisali
bazen")
            } else {
                toast.error(response.data.error)
            }
        } catch (error) {
            toast.error(error.message);
        }
    }
}
```

Kod 3.3 Zahtjev za brisanje podataka o bazenu

Obrada svih poziva odvija se u poslužiteljskoj aplikaciji, u kontrolerima. Svaki resurs određen je svojim jedinstvenim identifikatorom. U kodu ispod (Kod 3.4) pokazana je funkcionalnost obrade zahtjeva za brisanjem bazena. Definirana je uloga koja smije pristupiti određenom resursu i metodi, putanja do tog resursa i povratna vrijednost funkcije koja vraća rezultat strukturiran u JSON formatu, ovisno o ishodu metode. Metoda prima identifikator bazena kroz URI zahtjeva.

```

[Authorize(Roles = "admin")]
[HttpDelete("deletePool/{poolId}")]
public JsonResult DeletePool(int poolId)
{ try
    { var pool = context.Pools.Find(poolId);
      context.Pools.Remove(pool);
      context.SaveChanges();
      var pools = context.Pools.OrderBy(p =>
p.Name).ToList();
      return new JsonResult(new { success = true, data =
pools });
    }
    catch (Exception ex)
    {return new JsonResult(new { success = false, error = ex
});}}

```

Kod 3.4 Obrada zahtjeva u poslužiteljskoj aplikaciji

U kodu (Kod 3.5) je prikazan postupak ažuriranja osobnih podataka korisnika. Prvo poslužiteljska aplikacija pokušava dohvatiti korisnika za kojeg se trebaju ažurirati podaci. Ako ga baza podataka uspješno vrati podaci se ažuriraju, te se promjene spremaju natrag u bazu i povratna informacija o uspjehu se vraća korisniku. Ako korisnik nije pronađen u bazi podataka, poslužiteljska aplikacija obavještava korisnika o neuspješnom ažuriranju.

```

[HttpPut("updateProfile/{userId}")]
public JsonResult UpdateUserProfile([FromBody]
UpdateProfileDTO user, int userId)
{
    try
    {
        var DBuser = context.Users.Find(userId);
        if (DBuser == null)
        {
            return new JsonResult(new { success = false,
error = "Ne postoji korisnik s tim identifikatorom" });
        }
        else

```

```

        {
            DBuser.Name = user.Name;
            DBuser.Surname = user.Surname;
            DBuser.Email = user.Email;

            if (user.Password != "")
            {
                var passwordHasher = new
PasswordHasher<string>();
                var hashedPassword =
passwordHasher.HashPassword(DBuser.Username, user.Password);
                DBuser.Password = hashedPassword;
            }

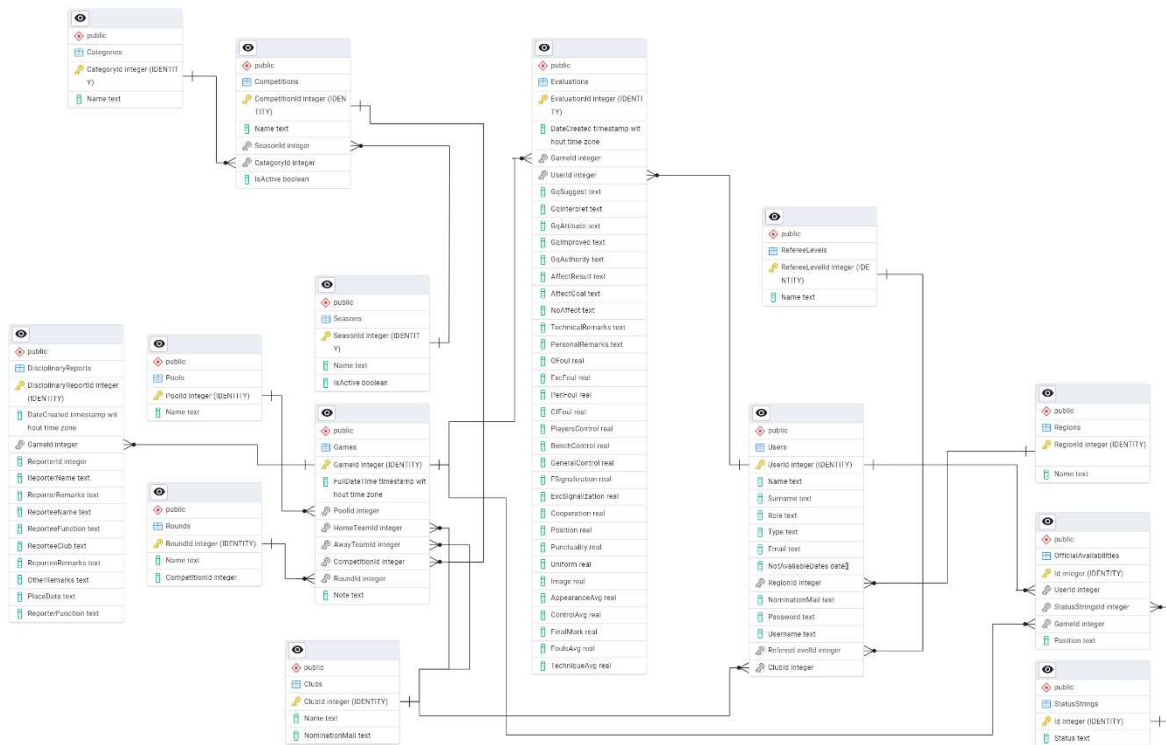
            context.SaveChanges();
            return new JsonResult(new { success = true, data
= "" });
        }
    }
    catch (Exception ex)
    {return new JsonResult(new { success = false, error = ex
});
    }
}

```

Kod 3.5 Ažuriranje korisnikovih podataka

3.3. Entiteti

Na slici (Slika 3.2) je prikaz svih entiteta sadržanih u bazi podataka i njihovih veza. Svaki entitet, sa sadržanim pripadajućim atributima, posebno je objašnjen u nastavku.



Slika 3.2 Dijagram entiteta i veza

3.3.1. Korisnik

Svaki korisnik sadrži ime, prezime, ulogu, tip, email, listu dana kada je nedostupan za delegiranje, jedinstveno korisničko ime i lozinku. Također sadrži strane ključeve za entitete regija, sudačka lista i klub. Uloga može biti korisnik ili administrator, dok u sustavu mogu biti tri tipa korisnika – sudac, delegat ili klupska osoba. Ako je tip korisnika sudac ili delegat, korisnik ima zabilježene podatke o sudačkoj listi, inače klupska osoba ima podatak kojem klubu pripada.

3.3.2. Utakmica

Pojedina utakmica se sastoji od vlastitih atributa datum odigravanja i bilješka, dok su ostali atributi strani ključevi – bazen, domaća ekipa, gostujuća ekipa, natjecanje i kolo.

3.3.3. Natjecanje

Svako natjecanje ima svoj naziv i zastavicu je li aktivno. Od stranih ključeva posjeduje identifikator sezone i kategorije.

3.3.4. Kategorija

Pojedina kategorija u bazi podataka sadrži samo svoj naziv.

3.3.5. Sezona

Svaka sezona ima svoj naziv te zastavicu je li aktivna.

3.3.6. Kolo

Svako kolo se u bazu podataka sprema s nazivom i identifikatorom natjecanja kojem pripada.

3.3.7. Klub

Klub posjeduje attribute naziv i email za delegiranja koji služi za slanje službenih obavijesti o delegiranju kroz sustav.

3.3.8. Bazen

Bazen je jednostavan entitet koji sadrži samo svoj naziv kao jedini atribut u tablici baze.

3.3.9. Regija

Svaka regija sadrži svoj naziv. Ona se pridjeljuje svim tipovima korisnika – sucu, delegatu i klupskoj osobi.

3.3.10. Sudačka lista

Sudačka lista je također još jedan jednostavan entitet koji sadrži samo svoj naziv.

3.3.11. Status službenih osoba

Entitet status korisnika je tablica u kojoj su sadržani svi mogući statusi u kojem se službena osoba može naći u odnosu na njezinu dostupnost prema delegiranju za utakmicu – dostupan, nedostupan, delegiran, potvrdio i otkazao.

3.3.12. Dostupnost službenih osoba

U entitetu dostupnost službenih osoba, koji ima vlastiti atribut pozicija, te strane ključeve na entitete korisnik, status službenih osoba i utakmica, sadržane su sve informacije o dostupnosti svakog korisnika za pojedinu utakmicu.

3.3.13. Evaluacija

Entitet evaluacija je najopsežniji entitet u bazi zbog količine podataka koji idu u jednu sudačku evaluaciju. Sastoji se od dva strana ključa – korisnik i utakmica, te 30 vlastitih atributa koji kroje konačnu ocjenu kroz evaluaciju za određenu utakmicu pojedinog korisnika.

3.3.14. Disciplinska prijava

Svaka disciplinska prijava posjeduje podatke o vremenu kad je kreirana, podatke i izjavu prijavitelja, te podatke o prijavljenoj osobi. Također može sadržavati i dodatne izjave ostalih uključenih dionika po potrebi.

3.4. Klijentska aplikacija

Klijentska aplikacija napisana je u programskom jeziku JavaScript koristeći Vue.js okvir za razvoj korisničkih sučelja. Glavne odlike Vue.js okvira su modularnost i reaktivnost kojima sam se služio kako bih stvorio komponente koje se mogu ponovno koristiti čime sam smanjio redundanciju koda i povećao organizaciju i preglednost koda aplikacije.

3.4.1. Pinia

Pinia je alat za upravljanjem stanja aplikacije. Omogućuje centralizirano upravljanje podacima aplikacije u spremištu (eng. store) što znači da se stanje može dijeliti između različitih komponenti aplikacije bez potrebe za direktnim prosljeđivanjem podataka kroz više razina komponenti [5]. U spremištu za korisnika korištene su akcije za postavljanje tokena pri prijavi u sustav te brisanje pri odjavi iz sustava (Kod 3.6), metode za provjeravanje autentificiranosti te tipa i uloge prijavljenog korisnika kako bi se unutar aplikacije mogle prikazivati različite funkcionalnosti ovisno o korisnikovom tipu i ulozi (Kod 3.7).

```
actions: {
  setToken(token, useCookies = true) {
    this.token = token;

    if (useCookies) {
      document.cookie = `authToken=${token}; path=/;
secure; samesite=strict`;
    } else {
      localStorage.setItem('authToken', token);
    }
  },
  logout() {
    window.location.href = "/login";
    this.token = null;
    document.cookie = 'authToken=; path=/; max-age=0;';
    localStorage.removeItem('authToken');
  },
}
```

Kod 3.6 Akcije u korisničkom spremniku


```
getters: {
  isAuthenticated: (state) => !!state.token,
  getUserId: (state) => {
    const decoded = state.token ? jwtDecode(state.token)
: null;
    return decoded?.userId;
  },
  isAdmin: (state) => {
    const decoded = state.token ? jwtDecode(state.token)
: null;
    return decoded?.role === 'admin';
  },
  isClubPerson: (state) => {
    const decoded = state.token ? jwtDecode(state.token)
: null;
    return decoded?.type === 'clubPerson';
  },
},
```

Kod 3.7 Dohvatne metode u korisničkom spremištu

3.4.2. Provjera pristupa

Metode definirane u korisničkom spremištu potom su korištene kroz aplikaciju za provjeru tipa i uloge korisnika i u ovisnosti o rezultatu različite funkcionalnosti i izgled aplikacije je prikazan korisniku. U dokumentu u kojem su definirane sve rute za navigaciju kroz aplikaciju, kreirane su metode za provjeru pristupa svakoj pojedinoj stranici (Kod 3.8).

```
const checkAdminAccess = (to, from, next) => {
  const store = useUserStore();
  if (!store.isAdmin) {
    toast.warning('Nemate pristup toj stranici');
    return next('/calendar');
  }
  next();
};

const checkAuthorizedAccess = (to, from, next) => {
  const store = useUserStore();
  if (!store.isAuthenticated) {
    toast.warning('Niste prijavljeni u sustav');
    return next('/login');
  }
  next();
};
```

Kod 3.8 Metode za provjeru pristupa

Zatim su te metode korištene pri definiranju pojedinih ruta unutar aplikacije za automatsko provjeravanje pristupa. Primjer određenih ruta vidljiv je u sljedećem kodu (Kod 3.9).

```
const routes = [
  { path: '/', component: Login },
  { path: '/calendar', component: CalendarContainer,
  beforeEnter: checkAuthorizedAccess },
  { path: '/officials', component: ListOfOfficials,
  beforeEnter: checkAdminAccess },
  { path: '/officialDetails/:userId', component:
  DetailOfficial, beforeEnter: checkAuthorizedAccess }
]
```

Kod 3.9 Definiranje ruta s metodama provjere pristupa

Još jedan primjer korištenja metoda iz spremnika je u komponenti za navigaciju gdje se ovisno o tipu i ulozi korisnika prijavljenog u sustav prikazuju tri različita tipa navigacijske linije za kretanje kroz aplikaciju (Kod 3.10).

```
<template>
  <AdminNavigationBar v-if="isAdmin"
:navName="props.navName" @logout="logout" />
  <ClubPersonNavigationBar v-if="isClubPerson"
:navName="props.navName" @logout="logout" />
  <UserNavigationBar v-if="!isAdmin && !isClubPerson"
:navName="props.navName" @logout="logout" />
</template>

<script setup>
  ...
  import AdminNavigationBar from './AdminNavigationBar.vue'
  import UserNavigationBar from './UserNavigationBar.vue'
  import ClubPersonNavigationBar from
'./ClubPersonNavigationBar.vue'
  import { useUserStore } from '../..../store.js'
  ...
  const router = useRouter()
  const store = useUserStore()
  const { isAuthenticated, isAdmin, isClubPerson } =
storeToRefs(store)

  const props = defineProps(['navName'])

  function logout() {
    store.logout()
    router.push("/login")
  }
</script>
```

Kod 3.10 Isječak navigacijske komponente s primjenom metoda korisničkog spremišta

3.4.3. Korisničke poruke

Za prikaz korisničkih poruka kada korisnik izvrši neku akciju ili pri pokušaju predaje nekih manjkavih podataka, koristio sam biblioteku Toast za prikaz obavijesti. Biblioteka koja je pisana za Vue.js aplikacije omogućava prikazivanje privremenih poruka korisnicima u određenom dijelu ekrana i zadanom vremenskom trajanju, te ostalim vizualnim modularnim elementima [6]. U nastavku (Kod 3.11) su prikazana tri primjera korištenja – pri uspjehu određene akcije na poslužiteljskoj strani, pogreške pri obradi zahtjeva ili upozorenju o pogrešnim ili manjkavim podacima u formi.

```
try {
  const response = await api.put('/user/updateUser/' +
  userId.value, userData)
  if (response.data.success) {
    toast.success("Uspješno ste ažurirali korisnika")
    resetData(response.data.data)
  } else {
    toast.error(response.data.error)
  }
} catch (error) {
  toast.error(error.message);
}
} else {
  toast.warning("Unesite sve tražene podatke")
}
```

Kod 3.11 Primjer korištenja biblioteke Toast s korisničkim porukama

3.4.4. API pozivi

Za komunikaciju, slanje i primanje podataka s poslužiteljske aplikacije, koristim Axios. To je JavaScript biblioteka koja nudi podršku za REST API, automatski rukuje JSON podacima, omogućuje prilagodbu zahtjeva i odgovora te lako integrira autentifikaciju putem JWT-a (eng. JSON Web Token) [7].

Pri slanju zahtjeva koristim Axiosovu mogućnost presretača, koja prije nego što klijentska aplikacija pošalje zahtjev na poslužitelj, provjerava token korisnika, te ako postoji dodaje ga zahtjevu za autentifikaciju korisnika na poslužitelju (Kod 3.12).

Također koristim presretač i pri dohvat u odgovora gdje, ako se dogodila pogreška na poslužitelju, u ovisnosti o tipu pogreške ispisujem poruku korisniku o grešci (Kod 3.13).

```
apiClient.interceptors.request.use(  
  (config) => {  
    const token = store.getToken  
    if (token) {  
      config.headers.Authorization = `Bearer ${token}`;  
    }  
    return config;  
  },  
  (error) => {  
    return Promise.reject(error);  
  });
```

Kod 3.12 Korištenje funkcionalnosti presretača Axios poziva i dodavanje tokena u zahtjev

```
apiClient.interceptors.response.use(  
  (response) => {  
    return response;  
  },  
  (error) => {  
    if (error.response?.status === 401) {  
      toast.error('Niste prijavljeni u sustav');  
      window.location.href = '/login';  
    }  
    if (error.response?.status === 403) {  
      toast.warning('Nemate pristup ovoj stranici');  
      window.location.href = '/calendar';  
      return Promise.reject(error);  
    }  
    const errorMessage =  
      error.response?.status === 404  
        ? 'Resurs nije pronađen'  
        : error.response?.status === 500  
          ? 'Pogreška pri dohvat u podataka.  
Pokušajte ponovno': error.message || 'Dogodila se neočekivana  
pogreška';  
    toast.error(errorMessage);  
    return Promise.reject(error);  
  });
```

Kod 3.13 Korištenje funkcionalnosti presretača Axios poziva pri dohvat u podataka s poslužitelja

3.4.5. JWT

JWT (eng. JSON Web Token) je internet standard za siguran prijenos podataka između dviju strana, obično klijentske i poslužiteljske aplikacije. Najčešće se koristi za autentifikaciju i autorizaciju korisnika [8]. Pri prijavi u sustav, klijentska aplikacija šalje korisničko ime i lozinku, poslužitelj provjerava podatke te ako su ispravni kreira token kojeg potom šalje natrag klijentu kojeg on sprema te koristi pri svakom sljedećem zahtjevu prema poslužitelju. Server pri obradi zahtjeva dekodira token i provjerava podatke na temelju kojih odlučuje ima li korisnik pravo na određenu radnju.

3.5. Poslužiteljska aplikacija

Poslužiteljska aplikacija razvijena je u jeziku C# koristeći ASP.NET Core okvir. Komuniciranje s PostgreSQL bazom podataka odvija se uz Entity Framework Core, ORM alat koji omogućuje objektno orijentirani pristup. Pri kreiranju baze podataka korišten je Code-First pristup. Modeli su definirani u poslužiteljskoj aplikaciji u C# jeziku, te su se potom putem migracija generirale tablice u bazi podataka.

Aplikacija koristi RESTful API arhitekturu – podaci su izloženi i pristupa im se putem kontrolera. Svaki kontroler obrađuje zahtjeve i posreduje u komunikaciji klijentske aplikacije i baze podataka.

Kroz poslužiteljsku aplikaciju također je implementirana i razina sigurnosti koristeći JWT autentifikaciju i autorizaciju, te dodatno definirajući DTO (eng. Data transfer Object), naziv za strukturu podataka koja služi za enkapsulaciju podataka, osiguravajući prijenos isključivo relevantnih i potrebnih podataka.

3.5.1. Modeli

Za kreiranje tablica u bazi podataka korišten je Code-First pristup putem Entity Framework Core. Ovaj pristup omogućuje definiranje strukture baze podataka putem C# klasa umjesto direktnog kreiranja tablica u bazi podataka. Nakon definiranja modela, s pomoću migracija generira se baza podataka i njezina struktura. U kodu (Kod 3.14) možemo vidjeti primjer modela utakmice. Postupak kreiranja migracije vidljiv je u sljedećem isječku (Kod 3.15).

```
public class Game
{
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int GameId { get; set; }
    public DateTime FullDateTime { get; set; }
    public int PoolId { get; set; }
    public virtual Pool? Pool { get; set; }
    public int HomeTeamId { get; set; }
    [ForeignKey("HomeTeamId")]
    public virtual Club? HomeTeam { get; set; }
    public int AwayTeamId { get; set; }
    [ForeignKey("AwayTeamId")]
    public virtual Club? AwayTeam { get; set; }
    public ICollection<OfficialAvailability> Officials { get;
set; } = null!;
    public int CompetitionId { get; set; }
    public virtual Competition? Competition { get; set; }
    public int? RoundId { get; set; }
    public virtual Round? Round { get; set; }
    public string? Note { get; set; }
}
```

Kod 3.14 Model utakmice koji predstavlja tablicu u bazi podataka

```
dotnet ef migrations add gameModel
dotnet ef database update
```

Kod 3.15 Primjer kreiranja migracije i ažuriranja baze podataka

3.5.2. Kontroleri

Kontroler je komponenta MVC (eng. Model-View-Controller) arhitekture koja upravlja zahtjevima i odgovorna je za komunikaciju između klijentske aplikacije i poslovne logike na poslužitelju. Obrađuje zahtjeve, dohvaća podatke iz baze i vraća odgovore. Na primjeru kontrolera za entitet bazen prikazat ćemo funkcionalnosti i značajke kontrolera.

Za svaki kontroler moguće je definirati osnovnu putanju kojom se pristupa resursima, kao i atribut koji osigurava autorizaciju korisnika pri pristupanju metodama kontrolera (Kod 3.16).

```
[Authorize]
[Route("api/[controller]")]
```

Kod 3.16 Definiranje autorizacije i osnovne putanje kontrolera

Potom se definiraju metode kojima se odrađuje poslovna logika, komunicira s bazom i obavljaju CRUD operacije (eng. Create, Read, Update, Delete). Svaka metoda posebno može imati atribut kojim se provjerava tip korisnika, pa tako putanji za brisanje određenog bazena može pristupiti samo korisnik koji ima tip administrator (Kod 3.17).

```
[Authorize(Roles = "admin")]
[HttpDelete("deletePool/{poolId}")]
public JsonResult DeletePool(int poolId)
{
    try
    {
        var pool = context.Pools.Find(poolId);
        context.Pools.Remove(pool);
        context.SaveChanges();
        var pools = context.Pools.OrderBy(p =>
p.Name).ToList();
        return new JsonResult(new { success = true, data =
pools });
    }
    catch (Exception ex)
    {
        return new JsonResult(new { success = false, error =
ex });
    }
}
```

Kod 3.17 Metoda DeletePool s administratorskim pristupom

3.5.3. DTO

DTO (eng. Data Transfer Object) je naziv za strukturu podataka koja služi za enkapsulaciju podataka, osiguravajući prijenos isključivo relevantnih i potrebnih podataka. Sakriva suvišne detalje modela baze podataka, sprječava izlaganje osjetljivih podataka te smanjuje količinu podataka koji se prenose. U primjeru u nastavku (Kod 3.18) prikazan je definicija jednog DTO-a koji služi za prijenos podataka o kolegama unutar jedne utakmice gdje se šalju samo relevantni podaci kao što su ID, ime i prezime, umjesto da se šalje cijeli entitet korisnika.

```
public class ColleagueDTO
{
    public int UserId { get; set; }
    public string Name { get; set; }
    public string Surname { get; set; }
}
```

Kod 3.18 Definicija ColleagueDTO objekta

Potom se taj DTO koristi pri zahtjevu za dohvat utakmica pojedinog korisnika i unutar odgovora se DTO koristi za atribut `Referees` i `Delegate` (Kod 3.19). također možemo vidjeti još jedan DTO – `AllGamesDTO` gdje također za utakmicu šaljemo samo određene, u tom trenutku, potrebne podatke.

```
var res = context.Games
    {...}
    .Select(x => new AllGamesDTO()
    {
        GameId = x.GameId,
        {...}
        Referees = x.Officials
        .Where(x => x.User.Type == "referee"
        && (x.StatusStringsId == 3 || x.StatusStringsId == 4))
        .Select(x => new ColleagueDTO()
        {
            UserId = x.UserId,
            Name = x.User.Name,
            Surname = x.User.Surname
        }).ToArray() {...}
```

Kod 3.19 Upotreba ColleagueDTO

3.5.4. Registracija korisnika

Pri postavljanju sustava inicijalno je kreiran korisnik s administratorskim ovlastima koji ima mogućnost stvaranja novih korisnika s ulogom „korisnik“. Samostalna registracija nije moguća. Pri stvaranju korisnika, administrator unosi osnovne podatke o korisniku, kao što su ime, prezime, tip, email, regija kojoj pripada itd. Također unosi i korisničko ime i lozinku koja se šalje na poslužitelj te se potom kriptografski sažima kako bi se spriječilo curenje podataka i povećala sigurnost. U kodu ispod (Kod 3.20) prikazana je metoda dodavanja korisnika. Ona prvo provjerava administratorovu ulogu (autorizacija), a potom kriptografski sažima lozinku i sprema novog korisnika u bazu podataka.

```
[Authorize(Roles = "admin")]
[HttpPost("addNewUser")]
public JsonResult PostAddNewUser([FromBody] User user)
{
    try
    {
        var passwordHasher = new
PasswordHasher<string>();
        var hashedPassword =
passwordHasher.HashPassword(user.Username, user.Password);
        user.Password = hashedPassword;

        context.Users.Add(user);
        context.SaveChanges();
    }
}
```

Kod 3.20 Spremanje novog korisnika u bazu podataka

3.5.5. Prijava korisnika

Nakon što je administrator kreirao novog korisnika, isti se može prijaviti u sustav. To radi tako što na početnoj stranici za prijavu unese korisničko ime i lozinku u formu za prijavu. Klijentska aplikacija potom šalje zahtjev za prijavu, a poslužiteljska aplikacija provjerava predane podatke. Ako su podaci ispravni, kreira se token koji se, skupa s korisnikovim podacima, vraća klijentskoj aplikaciji koja nastavlja s obradom podataka i usmjerava korisnika ovisno o njegovoj ulozi. U nastavku je prikazana metoda prijave u sustav (Kod 3.21).

```

[HttpPost("login")]
public IActionResult Login([FromBody] LoginDTO loginDto)
{
    try
    {
        var user =
        _userService.Authenticate(loginDto.Username,
        loginDto.Password);
        if (user == null)
            return Unauthorized(new { success = false, error
= "Neispravno korisničko ime ili lozinka" });

        var token = _jwtService.GenerateToken(user);
        returnUser = new
        {
            userId = user.UserId,
            name = user.Name,
            surname = user.Surname,
            type = user.Type,
            role = user.Role,
            username = user.Username
        };

        return Ok(new { success = true, data = new { token,
user = returnUser } });
    }
    catch (Exception)
    {
        return NotFound(new { success = false, error =
"Pogreška pri obradi podataka" });
    }
}

```

Kod 3.21 Prijava u sustav

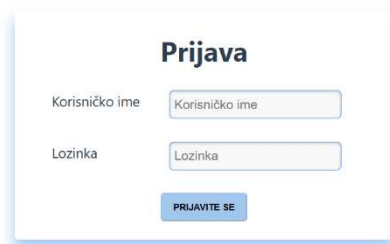
4. Korisničko sučelje

U nastavku su prikazani ekrani izgrađenog korisničkog sučelja klijentske aplikacije. Određene stranice u aplikaciji imaju različit prikaz ovisno o tipu i ulozi korisnika prijavljenog u sustav.

4.1. Prijava u sustav

Prijava u sustav se vrši unosom korisničkog imena i lozinke u formu za prijavu i pritiskom na gumb prijavite se. Potom klijentska aplikacija šalje zahtjev na poslužitelj koji provjerava podatke, te ako su ispravni vraća token i korisničke podatke te ovisno o tipu korisnika preusmjerava na početnu stranicu s dostupnim funkcionalnostima (Slika 4.1).

Sustav praćenja rada službenih osoba u vaterpolu



The image shows a login form with the title "Prijava". It features two input fields: "Korisničko ime" and "Lozinka". Below the fields is a blue button with the text "PRIJAVITE SE".

Slika 4.1 Prijava u sustav

4.2. Kalendar

Po uspješnoj prijavi u sustav korisnika se preusmjerava na početnu stranicu – kalendar. Ako je korisnik tipa administrator, navigacijska traka će biti prilagođena administratoru te će funkcionalnosti za pojedinu utakmicu (delegiranje, uređenje i brisanje) biti omogućene (Slika 4.2).

Kalendar Službene osobe Nova utakmica Popis utakmica Bazeni Osobe Sezone Klubovi Statistika Natjecanja Delegiranja Javna delegiranja Sudačke liste Moj profil Odjava

Veljača 2025 < Danas >

Pon	Uto	Sri	Čet	Pet	Sub	Ned
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

POPIS UTAKMICA DOSTUPNOST SLUŽBENIH OSOBA

1.B liga **SAKRIJ**

Medveščak - Jadran Kostrena 1. kolo Utrine, Zagreb 19.00 h
 Suci: Pero Bartičević ? Cvijeto Radović ? Delegat: Stijepo Čubela ?
 DELEGIRAJ AŽURIRAJ IZBRIŠI

Bellevue - Opatija 1. kolo Gruž, Dubrovnik 18.00 h
 Suci: Toni Paskojević ? Sara Kontek ? Delegat: Srđan Dragojević ?
 DELEGIRAJ AŽURIRAJ IZBRIŠI

Prvenstvo Hrvatske kadeti **SAKRIJ**

Jadran ST - OVK Split 4. kolo Poljud, Split 12.30 h
 Suci: Mateo Konsuo ? Ivan Matijević ? Delegat: Milan Grbin ?
 DELEGIRAJ AŽURIRAJ IZBRIŠI

Slika 4.2 Popis utakmica na odabrani datum u kalendaru

Pritiskom na gumb delegiraj otvara se skočni prozor za odabir sudaca i delegata za odabranu utakmicu (Slika 4.3).

Kalendar Službene osobe Nova utakmica Popis utakmica Bazeni Osobe Sezone Klubovi Statistika Natjecanja Delegiranja Javna delegiranja Sudačke liste Moj profil Odjava

Veljača 2025 < Danas >

Prvenstvo Hrvatske kadeti **SAKRIJ**

Jadran ST - OVK Split 4. kolo Poljud, Split 12.30 h

Sudac A
 Odaberite osobu

Sudac B
 Odaberite osobu

Delegat
 Odaberite osobu

DELEGIRAJ ODUSTANI

Slika 4.3 Delegiranje službenih osoba

Na istoj stranici administrator, uz prikaz utakmica sa službenim osobama, ima i mogućnost prikaza popisa svih službenih osoba i njihove dostupnosti za delegiranja na taj dan. Također može vidjeti osobe koje su već delegirane taj dan, imajući upozorenje u vidu uskličnika pored imena osobe (Slika 4.4).

Kalendar Službene osobe Nova utakmica Popis utakmica Bazeni Osobe Sezone Klubovi Statistika Natjecanja Delegiranja Javna delegiranja Sudačke liste Moj profil Odjava

Veljača 2025 < Danas >

Pon	Uto	Sri	Čet	Pet	Sub	Ned
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

POPIS UTAKMICA

Suci

[DUBROVNIK](#)
[SPLIT](#)
[ŠIBENIK](#)
[ZAGREB](#)
[RIJEKA](#)
[OSTALO](#)

Svi suci iz podružnice Šibenik su dostupni za delegiranje

DOSTUPNOST SLUŽBENIH OSOBA

Delegati

[DUBROVNIK](#)
[SPLIT](#)
[ŠIBENIK](#)
[ZAGREB](#)

Stijepo Čubela	!
Srđan Dragojević	!
Milan Grbin	!

Opres! Službena osoba je već delegirana na jednu od današnjih utakmica

Slika 4.4 Dostupnost službenih osoba na odabrani datum iz kalendara

4.3. Službene osobe

Na stranici službene osobe (Slika 4.5) korisnici su podijeljeni u dvije kategorije – suci i delegati. Pored svakog korisnika je gumb „detalji“ na kojeg pritiskom sustav vodi na novu stranicu s detaljnim prikazom statistike suđenja ili obavljanja dužnosti delegata (Slika 4.6).

Kalendar **Službene osobe** Nova utakmica Popis utakmica Bazeni Osobe Sezone Klubovi Statistika Natjecanja Delegiranja Javna delegiranja Sudačke liste Moj profil Odjava

Suci

[DUBROVNIK](#)
[SPLIT](#)
[ŠIBENIK](#)
[ZAGREB](#)
[RIJEKA](#)
[OSTALO](#)

Nema sudaca u podružnici Šibenik

Delegati

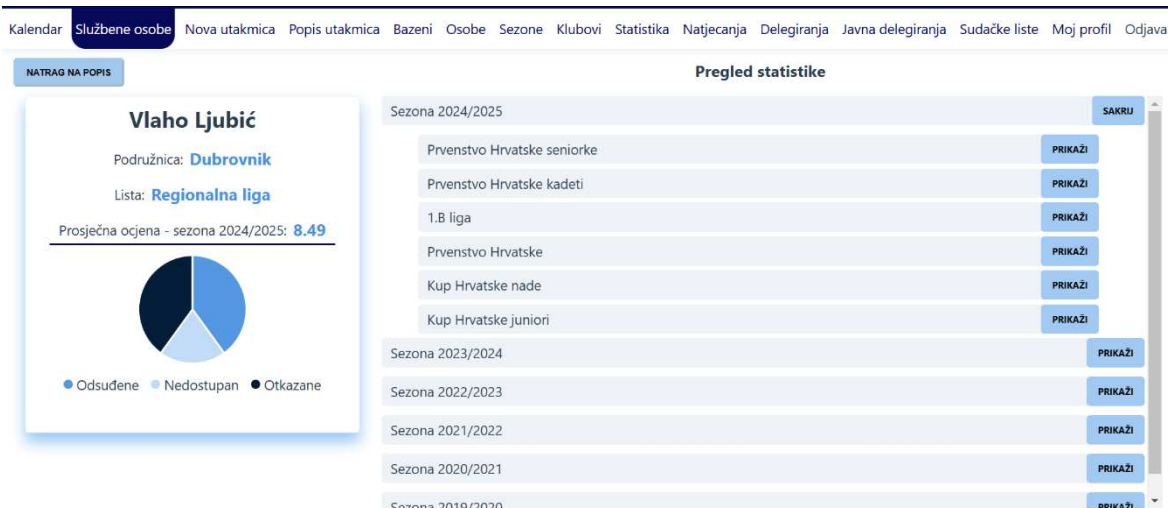
[DUBROVNIK](#)
[SPLIT](#)
[ŠIBENIK](#)
[ZAGREB](#)
[RIJEKA](#)
[OSTALO](#)

Luko Vuletić	DETALJI
Božo Vuletić	DETALJI
Stijepo Čubela	DETALJI
Srđan Dragojević	DETALJI
Nikola Curić	DETALJI

[PRETHODNA](#)
Stranica 1 od 2
[SLJEDEĆA](#)

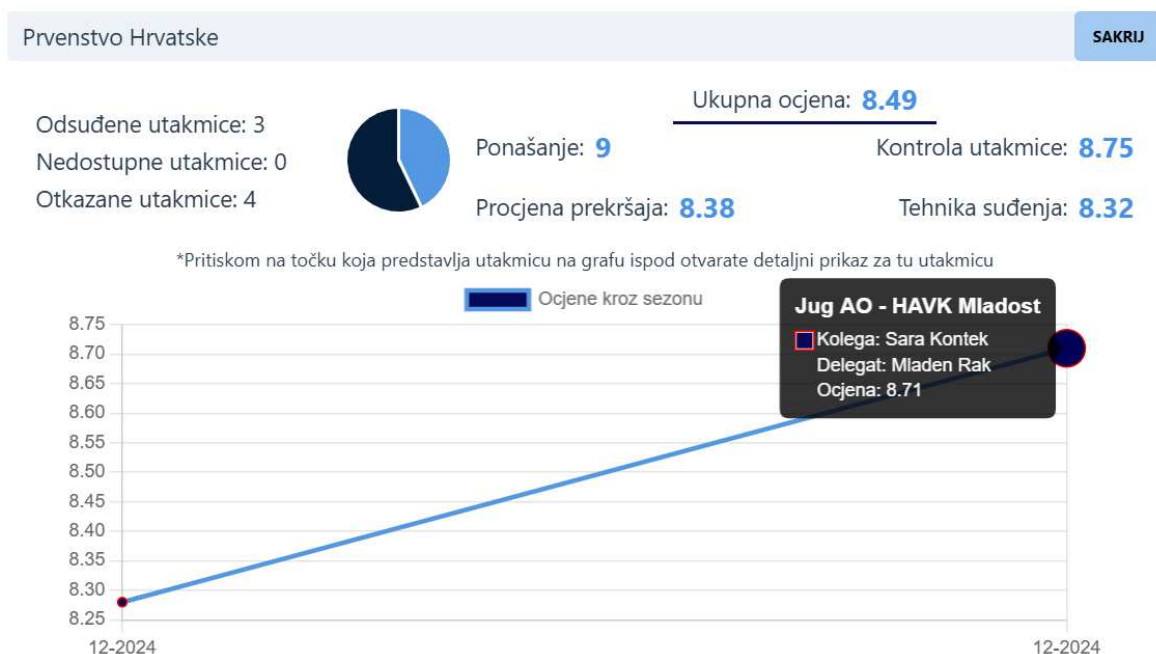
Slika 4.5 Popis službenih osoba

31



Slika 4.6 Pregled statistike suca u sezoni

Odabirom određenog natjecanja prikazuju se detalji za to natjecanje (Slika 4.7) – ukupan broj utakmica kroz natjecanje podijeljenih u kategorije (odsuđene, otkazane, nedostupne), ukupna prosječna ocjena za to natjecanje, te prosječne ocjene po kategorijama evaluacije, grafički prikaz kretanja ocjena po utakmicama kroz sezonu, te klikom na utakmicu prikazanu u grafičkom prikazu, pojavljuje se pojedina utakmica sa svojim prosječnim ocjenama (Slika 4.8).



Slika 4.7 Detalji pojedinog natjecanja

Jug AO - HAVK Mladost

Kolega: **Sara Kontek** Ukupna ocjena: **8.71** Delegat: **Mladen Rak**

Ponašanje: **9.33** Kontrola utakmice: **8.83** Procjena prekršaja: **8.38** Tehnika suđenja: **8.88**

Slika 4.8 Detalji pojedine utakmice

Ako u nekom natjecanju ili sezoni nema podataka za prikaz, prikazuje se prikladna poruka (Slika 4.9 i Slika 4.10).

Kup Hrvatske nade SAKRIJ

Nemate utakmica za prikaz u ovom natjecanju

Slika 4.9 Bez utakmica za prikaz u natjecanju

Sezona 2020/2021 SAKRIJ

Nemate natjecanja u ovoj sezoni

Slika 4.10 Bez natjecanja za prikaz u sezoni

4.4. Dodavanje ili ažuriranje utakmice

Za dodavanje nove utakmice u sustav ili ažuriranje podataka postojeće koristi se ista forma (Slika 4.11). Svaka forma unutar sustava ima vlastitu validacijsku metodu koja prije slanja zahtjeva na poslužitelj provjerava jesu li podaci uneseni.

Kalendar Službene osobe **Nova utakmica** Popis utakmica Bazeni Osobe Sezone Klubovi Statistika Natjecanja Delegiranja Javna delegiranja Sudačke liste Moj profil Odjava

Podaci o utakmici

Datum <input type="text"/>	Ekipa A <input type="text" value="Odaberite ekipu A"/>
Natjecanje <input type="text" value="Odaberite Natjecanje"/>	Ekipa B <input type="text" value="Odaberite ekipu B"/>
Kolo <input type="text" value="Odaberite kolo"/>	Napomena <input type="text"/>
Bazen <input type="text" value="Odaberite bazen"/>	

Slika 4.11 Dodavanje nove utakmice

4.5. Popis utakmica

Sljedeći ekran kojeg imaju svi korisnici je ekran za popis utakmica (Slika 4.12). Ako je korisnik s ulogom administratora, on vidi sve utakmice u sustavu, dok ostali korisnici na ovom popisu vide samo utakmice u kojima sudjeluju. Ako su to suci i delegati onda vide utakmice na kojima su sudili ili bili delegati, dok klupske osobe vide utakmice u kojima je igrao njihov klub.

Utakmice se mogu filtrirati po dva parametra – vremenskom i grupacijskom. Za vremenski se može odabrati prijašnje, buduće ili sve utakmice, dok ih se s druge strane može grupirati po natjecanju (sezoni i kolu) ili sortirati po datumu (Slika 4.13).

The screenshot shows the 'Popis utakmica' page with the following elements:

- Navigation menu: Kalendar, Službene osobe, Nova utakmica, **Popis utakmica**, Bazeni, Osobe, Sezone, Klubovi, Statistika, Natjecanja, Delegiranja, Javna delegiranja, Sudačke liste, Moj profil, Odjava
- Filter po vremenu: ODIGRANE UTAKMICE, **SVE UTAKMICE**, BUDUĆE UTAKMICE
- Filter po tipu: **PO DATUMU**, PO NATJECANJU
- Season: Sezona 2024/2025 (SAKRIJ)
- League: 1.B liga (SAKRIJ)
- Round: 1. kolo (SAKRIJ)
- Match 1: 3.02.2025 u 19.00 h, Utrine, Zagreb, Medveščak - Jadran Kostrena, Cvijeto Radović, Pero Bartičević, Stijepo Čubela (DETALJI)
- Match 2: 3.02.2025 u 18.00 h, Gruž, Dubrovnik, Bellevue - Opatija, Toni Paskojević, Sara Kontek, Srđan Dragojević (DETALJI)
- Round: 2. kolo (PRIKAŽI)
- Competition: Kup Hrvatske juniori (PRIKAŽI)
- Competition: Prvenstvo Hrvatske (SAKRIJ)

Slika 4.12 Popis vlastitih utakmica

The screenshot shows the 'Popis utakmica' page with the following elements:

- Navigation menu: Kalendar, Službene osobe, Nova utakmica, **Popis utakmica**, Bazeni, Osobe, Sezone, Klubovi, Statistika, Natjecanja, Delegiranja, Javna delegiranja, Sudačke liste, Moj profil, Odjava
- Filter po vremenu: ODIGRANE UTAKMICE, **SVE UTAKMICE**, **BUDUĆE UTAKMICE**
- Filter po tipu: **PO DATUMU**, PO NATJECANJU
- Match 1: 25.02.2025 u 10.30 h, Kantrida, Rijeka, Primorje EB - Pošk 1937, Ivan Matijević, Vlaho Ljubić, Luko Vuletić
- Match 2: 22.02.2025 u 19.30 h, Vukovar, Mornar - Primorje EB, Sara Kontek, Vlaho Ljubić, Nikola Curić
- Match 3: 12.02.2025 u 19.00 h, Poljud, Split, Bura - Marsonia, Vlaho Ljubić, Ivan Matijević, Dragan Štampalija
- Match 4: 8.02.2025 u 14.30 h, Zadar, Zadar - Osijek, Mato Miloslavić, Mateo Konsuo
- Match 5: 8.02.2025 u 13.15 h, Poljud, Split, Jadran ST - HAVK Mladost, Ivan Matijević, Mladen Rak

Slika 4.13 Filtrirane utakmice - buduće i grupirane po datumu

4.6. Bazeni

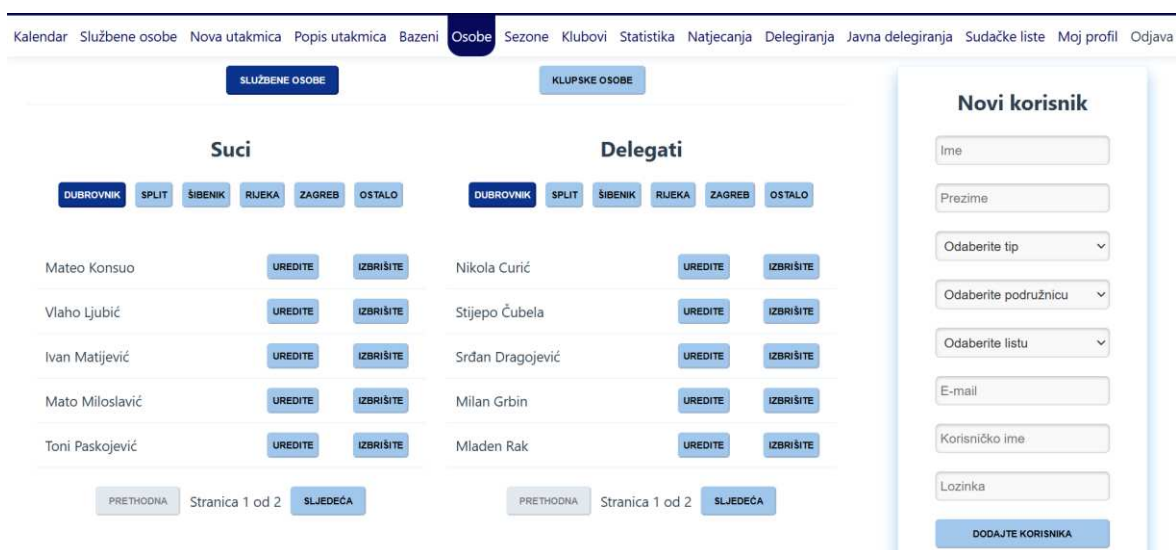
Administratorska stranica za pregled i uređenje bazena prikazana je na ispod (Slika 4.14).



Slika 4.14 Pregled i uređenje bazena

4.7. Korisnici

Administratorska stranica za pregled i uređenje korisnika sustava prikazana je na slici (Slika 4.15). Korisnici su podijeljeni na službene osobe i klupske osobe. Forma za dodavanje novih korisnika ili uređenje podataka postojećih dinamički se prilagođava tipu korisnika, pa tako suci i delegati nemaju svoj klub, dok klupske osobe nemaju listu kojoj pripadaju.



Slika 4.15 Pregled i uređenje korisnika

4.8. Sezone

Administratorska stranica za pregled i uređenje sezona prikazana je na slici (Slika 4.16). Uz naziv sezone, pri dodavanju ili uređenju podataka, sezona može biti aktivna ili neaktivna.



Slika 4.16 Pregled i uređenje sezona

4.9. Klubovi

Administratorska stranica za pregled i uređenje klubova prikazana je na slici (Slika 4.17). Uz naziv kluba, pri dodavanju ili uređenju podataka, klub mora imati i e-mail za mogućnost slanja delegiranja iz sustava.



Slika 4.17 Pregled i uređenje klubova

4.10. Klupska statistika sudjelovanja

Sustav također omogućuje pregled statistike svakog kluba u vidu njegovih nastupa u aktivnoj sezoni. Sljedeća stranica (Slika 4.18) prikazuje pregled statistike za određeni klub u aktivnoj natjecateljskoj sezoni.

The screenshot shows a web interface for club statistics. At the top, there is a navigation bar with links: Kalendar, Službene osobe, Nova utakmica, Popis utakmica, Bazeni, Osobe, Sezone, Klubovi, Statistika (highlighted), Natjecanja, Delegiranja, Javna delegiranja, Sudačke liste, Moj profil, and Odjava. Below the navigation bar, the main content is divided into two sections. On the left, under the heading "Popis klubova", there is a list of clubs: Primorje EB, Jug AO, HAVK Mladost, Jadran ST, and Cavtat. Each club name has a "DETALJI" button next to it. At the bottom of this list, there are navigation buttons: "PRETHODNA", "Stranica 1 od 4", and "SLJEDEĆA". On the right, there is a detailed view for the club "Jadran ST". It shows the current season "Sezona 2024/2025" with a "SAKRI" button. Below this, it states "Broj natjecanja u kojima je sudjelovao klub: 4". A list of competitions follows: "Kup Hrvatske juniori", "Prvenstvo Hrvatske", "Prvenstvo Hrvatske kadeti", and "1.B liga".

Slika 4.18 Klupska statistika sudjelovanja u natjecanjima

4.11. Sudačke liste

Administratorska stranica za pregled i uređenje sudačkih listi vidljiva je na slici (Slika 4.19).

The screenshot shows a web interface for managing referee lists. At the top, there is a navigation bar with links: Kalendar, Službene osobe, Nova utakmica, Popis utakmica, Bazeni, Osobe, Sezone, Klubovi, Statistika, Natjecanja, Delegiranja, Javna delegiranja, Sudačke liste (highlighted), Moj profil, and Odjava. Below the navigation bar, the main content is divided into two sections. On the left, under the heading "Popis lista službenih osoba", there is a list of categories: "European aquatics", "Mlađe kategorije", "Nepoznato", "Prvenstvo Hrvatske", and "Regionalna liga". Each category has two buttons: "UREDITE" and "IZBRIŠITE". At the bottom of this list, there are navigation buttons: "PRETHODNA", "Stranica 1 od 2", and "SLJEDEĆA". On the right, there is a form for adding a new list, titled "Nova lista". It contains a text input field labeled "Naziv liste" and a "DODAJTE LISTU" button.

Slika 4.19 Pregled i uređenje sudačkih listi

4.12. Natjecanja

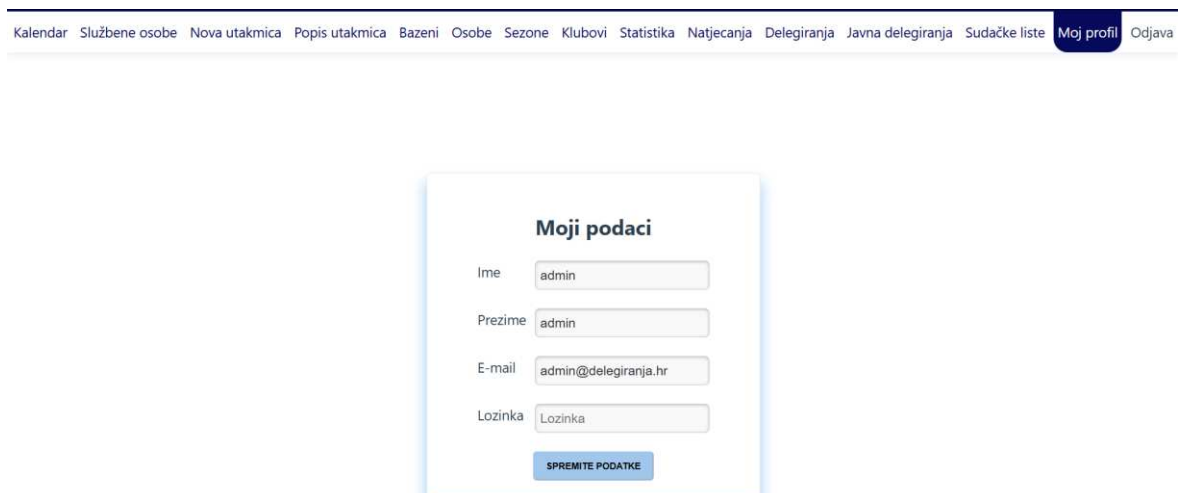
Administratorska stranica za pregled i uređenje natjecanja prikazana je na slici (Slika 4.20). Kako bi se zadržala preglednost s obzirom na veći broj mogućih natjecanja po sezoni, uvedena je paginacija. U formi za dodavanje ili uređenje podataka omogućena je forma glava-stavka gdje se dinamički pri dodavanju novog natjecanja mogu dodati i kola koja pripadaju tom natjecanju.



Slika 4.20 Pregled i uređenje natjecanja

4.13. Profil

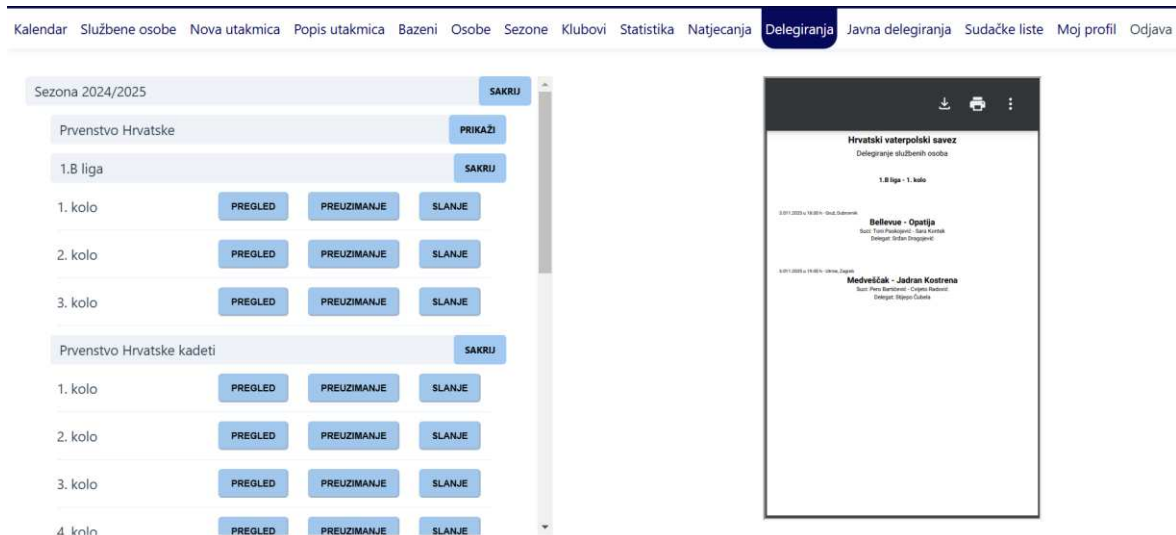
Administratorska stranica za pregled i uređenje profila prikazana je na slici (Slika 4.21).



Slika 4.21 Pregled i uređenje osobnih podataka

4.14. Delegiranja

Komponenta za prikaz delegiranja koristi se na dvije stranice – unutar sustava s administratorskim pristupom (Slika 4.22) te stranica za javno dostupno delegiranje (Slika 4.23) kojoj je moguće pristupiti bez prijave u sustav. Administrator ima dodatnu mogućnost slanja delegiranja pritiskom na gumb slanje.



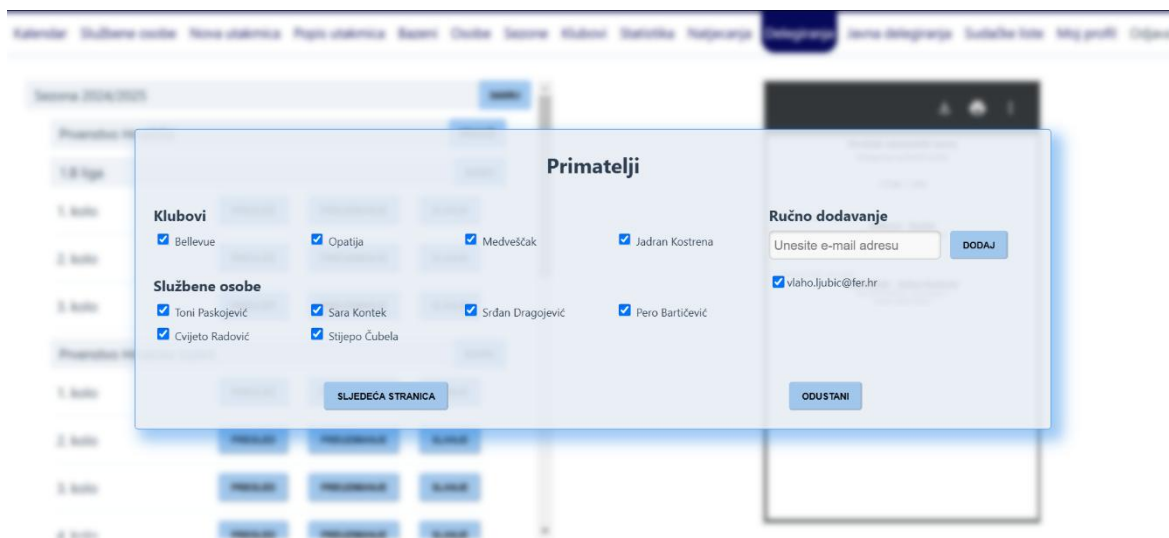
Slika 4.22 Administratorski prikaz stranice za delegiranja



Slika 4.23 Javno dostupna stranica za pregled delegiranja

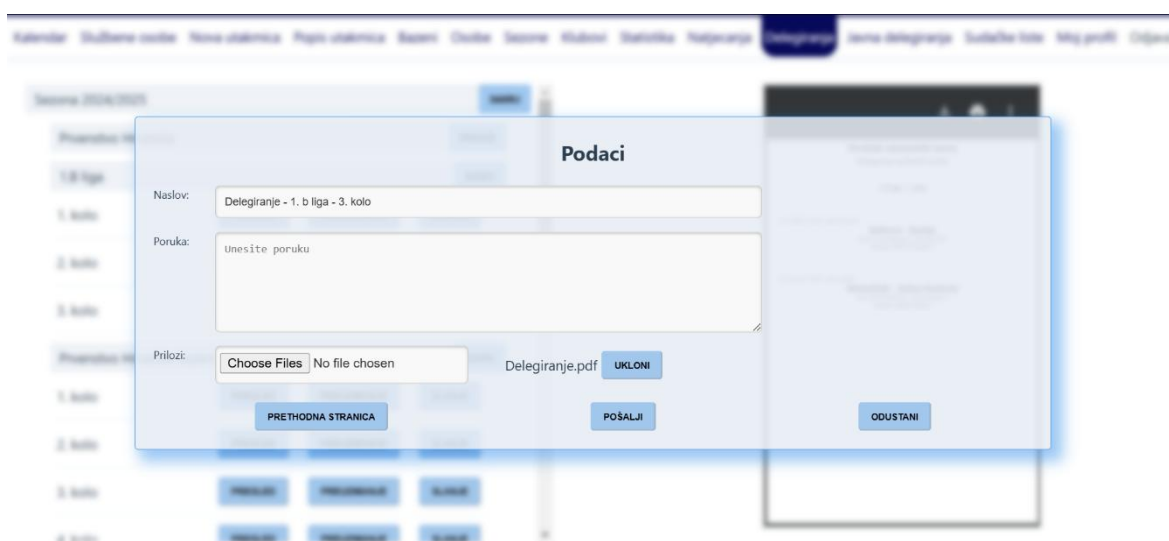
4.15. Slanje delegiranja

Na administratorskoj stranici delegiranja postoji mogućnost slanja dokumenta delegiranja za svako pojedino kolo određenog natjecanja. Pritiskom gumba slanje otvara se skočni prozor s dvije stranice. Na prvoj stranici (Slika 4.24) prikazane su službene osobe i klubovi sadržani u tom kolu za koje se delegiranje želi poslati. Također postoji i opcija ručnog dodavanja mail adrese ako se delegiranje želi poslati dodatnoj osobi.



Slika 4.24 Odabiranje osoba za slanje delegiranja mailom

Druga stranica sadrži elemente za unos naslova i teksta maila, kao i automatski pridodan PDF dokument pripremljenog delegiranja, uz mogućnost dodavanja dodatnih dokumenata (Slika 4.25).



Slika 4.25 Naslov, tekst i prilozi maila

4.16. Detalji o utakmici

Svi korisnici preko popisa utakmica u kojima sudjeluju mogu pregledati detalje o utakmici – evaluacije za suce, eventualne disciplinske prijave i osnovne podatke (Slika 4.26).

Kalendar Moja dostupnost Moja statistika **Moje utakmice** Moj profil Delegiranja Odjava

[POVRATAK NA UTAKMICE](#)

Medveščak vs Jadran Kostrena

1.B liga | 1. kolo | Utrine, Zagreb | 3.2.2025, 19.0h

Ocjenjivački listovi

Pero Bartičević	PREGLEDAJTE
Cvijeto Radović	ISPUNITE

Radnje

[KREIRAJTE DISCIPLINSKU PRIJAVU](#)

Dokumenti

Prijavitelj: Cvijeto Radović [PREGLEDAJTE PRIJAVU](#)

Prijavljeni: Maroje Ljubić

Slika 4.26 Pregled podataka o utakmici

Jedino korisnik s tipom delegat može ispuniti ocjenjivački list, svi ostali korisnici imaju samo mogućnost pregleda ako je ocjenjivački list ispunjen.

4.17. Evaluacije

Pritiskom na gumb sa stranice za pregled detalja o utakmici, ocjenjivački se listovi mogu ispuniti (delegat) ili pregledati (ostali korisnici). Za obje funkcionalnosti korištena je ista forma, s razlikama u mogućnošću uređivanja i predaje podataka (Slika 4.27 i Slika 4.28).

Kalendar Moja dostupnost Moja statistika **Moje utakmice** Moj profil Delegiranja Odjava

POVRATAK NA UTAKMICU

Medveščak - Jadran Kostrena

Pero Bartičević

Opća pitanja

Je li sudio/la autoritativno?

Je li se popravio/la u odnosu na prošli put?

Dobar stav i involviranost prije i poslije utakmice?

Bio/la je u stanju ispratiti utakmicu do kraja?

Predlažem davanje težih utakmica

Primjena prekršaja

Obični prekršaji

Isključenja

Peterci

Kontra prekršaji

Sudačka tehnika

Signalizacija običnih prekršaja

Signalizacija teških prekršaja

Suradnja s kolegom

Kretnja tijekom utakmice

Kontrola utakmice

Igrači u bazenu

Klupa i treneri

Kompletna kontrola utakmice

Protokol i ponašanje

Točnost

Izgled

Stav

Tehničke pogreške

Odluka utjecala na ishod utakmice

Odluka utjecala na postizanje pogotka

Odluka nije utjecala na postizanje pogotka

Bilješke i prijedlozi

Tehničke pogreške

Komentar i prijedlozi

Slika 4.27 Ispunjavanje evaluacije

Kalendar Moja dostupnost Moja statistika **Moje utakmice** Moj profil Delegiranja Odjava

POVRATAK NA UTAKMICU

Medveščak - Jadran Kostrena

Pero Bartičević

Opća pitanja

Je li sudio/la autoritativno?

Je li se popravio/la u odnosu na prošli put?

Dobar stav i involviranost prije i poslije utakmice?

Bio/la je u stanju ispratiti utakmicu do kraja?

Predlažem davanje težih utakmica

Primjena prekršaja

Obični prekršaji

Isključenja

Peterci

Kontra prekršaji

Sudačka tehnika

Signalizacija običnih prekršaja

Signalizacija teških prekršaja

Suradnja s kolegom

Kretnja tijekom utakmice

Kontrola utakmice

Igrači u bazenu

Klupa i treneri

Kompletna kontrola utakmice

Protokol i ponašanje

Točnost

Izgled

Stav

Tehničke pogreške

Odluka utjecala na ishod utakmice

Odluka utjecala na postizanje pogotka

Odluka nije utjecala na postizanje pogotka

Bilješke i prijedlozi

Tehničke pogreške

Komentar i prijedlozi

Ocjene po kategorijama

Ukupna ocjena: **7.98**

Ponašanje: **8.67** Kontrola utakmice: **8.33**

Procjena prekršaja: **8** Tehnika sudjenja: **7.5**

Slika 4.28 Pregled evaluacije

4.18. Disciplinske prijave

Za razliku od evaluacije, disciplinsku prijavu mogu ispuniti svi korisnici koji su sudjelovali u utakmici. Prijavu mogu podnijeti i osobe koji nisu aktivni korisnici sustava, odabirom opcije „Ostali“ kod prijavitelja. Slika (Slika 4.29) prikazuje ispunjenu disciplinsku prijavu.

The screenshot shows a web interface for reporting a disciplinary incident. The navigation bar includes 'Kalendar', 'Moja dostupnost', 'Moja statistika', 'Moje utakmice' (selected), 'Moj profil', 'Delegiranja', and 'Odjava'. A button 'POVRATAK NA UTAKMICU' is visible. The main heading is 'Disciplinska prijava'. The form is organized into four main sections: 'Opće informacije' (General information) with fields for 'Utakmica' (Medveščak - Jadran Kostrena), 'Natjecanje' (1.B liga), 'Kolo' (1. kolo), 'Mjesto' (Utrine, Zagreb), and 'Datum' (3.2.2025. 19.0h); 'Prijavljeni' (Reported) with fields for 'Ime i prezime' (Maroje Ljubić), 'Funkcija' (igrač), 'Klub' (Medveščak), and 'Izjava' (Ispriko, bilo je u afektu); 'Prijavitelj' (Reporter) with fields for 'Ime i prezime' (Cvijeto Radović) and 'Razlog prijave' (Nesportsko ponašanje); and 'Ostale izjave' (Other statements) with a field for 'Izjava' (Unesite izjavu). A date field at the bottom indicates 'Zagreb, 3.2.2025'.

Slika 4.29 Ispunjena disciplinska prijava

4.19. Javljanje nedostupnosti

Jedna od glavnih funkcionalnosti sustava je javljanje nedostupnosti za delegiranje za određene datume (Slika 4.30). Mijenjanje za datume u prošlosti je onemogućeno.

The screenshot shows a web interface for managing availability. The navigation bar includes 'Kalendar', 'Moja dostupnost' (selected), 'Moja statistika', 'Moje utakmice', 'Moj profil', 'Delegiranja', and 'Odjava'. The main heading is 'Moja dostupnost'. The current date is 'Na datum 04.02.2025.'. The user's availability for this date is 'Vaša dostupnost je dostupan'. A button 'PROMIENITE DOSTUPNOST' is visible. The calendar shows dates from 27 to 28 of the previous month, and 1 to 23 of the current month. Dates 14, 15, and 16 are highlighted in red, indicating unavailability.

Slika 4.30 Javljanje nedostupnosti

4.20. Klupska funkcionalnost

Osobama koje predstavljaju klub također je omogućena prijava u sustav. Te osobe, pored osnovnih funkcionalnosti koje su zajedničke svima – pregled utakmica kroz kalendar, promjena osobnih podataka, pregled javno dostupnog delegiranja – imaju mogućnost prijave satnice za svoje domaće utakmice. Popis utakmica prikazan je na slici (Slika 4.31), dok se pritiskom na gumb satnica otvara sljedeći prikaz (Slika 4.32).

Filter po vremenu		Filter po tipu		
ODIGRANE UTAKMICE	SVE UTAKMICE	BUDUĆE UTAKMICE	PO DATUMU	PO NATJECANJU
Sezona 2024/2025				
1.B liga				
16.01.2025 u 16.55 h	Gruž, Dubrovnik	Jug AO - Jadran ST	SATNICA	
Prvenstvo Hrvatske				
29.01.2025 u 13.30 h	Gruž, Dubrovnik	Jug AO - Cavtat	SATNICA	
16.01.2025 u 17.31 h	Gruž, Dubrovnik	Jug AO - Jadran ST	Vlaho Ljubić	Sara Kontek Dragan Štampalija

Slika 4.31 Popis domaćih klupskih utakmica

Prijava satnice

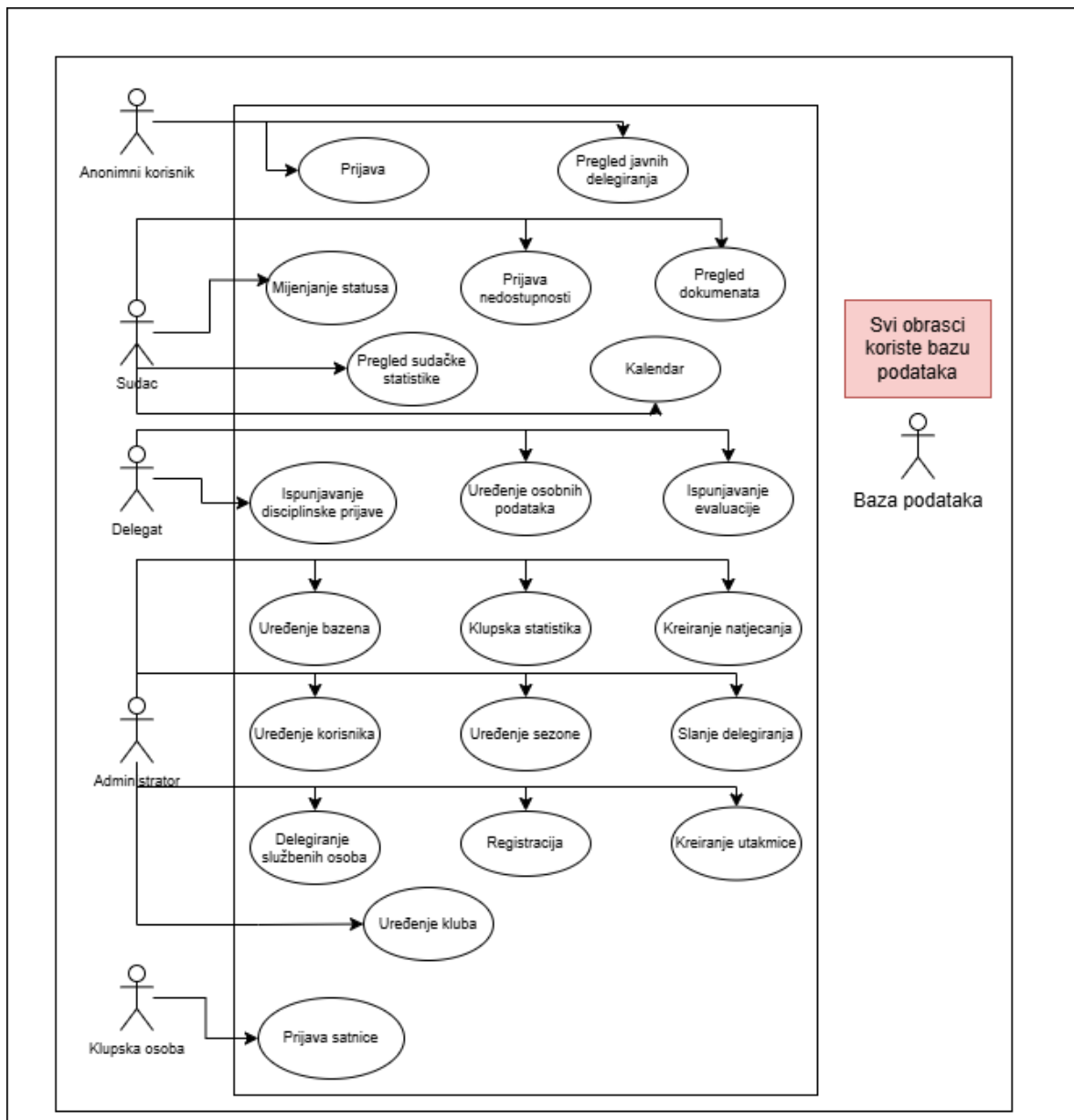
Jug AO - Jadran ST

📅 16-01-2025 16:55

Slika 4.32 Prijava satnice utakmice

5. Ponašanje sustava

Ponašanje sustava prema interakciji korisnika i zahtjevi koje sustav ispunjava prikazani su UML dijagramom obrazaca uporabe koji je prikazan na slici (Slika 5.1). Dijagram prikazuje glavne interakcije koje se kroz sustav mogu izvesti.



Slika 5.1 UML dijagram sustava

Dionici sustava su korisnik, administrator te anonimni korisnik koji ima samo jednu mogućnost – pregled javnih delegiranja. Korisnik može biti jedan od tri tipa – sudac, delegat i klupska osoba.

Da bi mogli koristiti sustav, osim već spomenutog pregledavanja javnog delegiranja, korisnici moraju biti prijavljeni u sustav. Kreiranje novog korisnika moguće je samo administratoru te nema procesa samostalne registracije. Nakon uspješne prijave u sustav korisnika se preusmjerava na početnu stranicu – kalendar. U nastavku će svaki obrazac uporabe biti zasebno objašnjen. Za one obrasce koji se istovjetno primjenjuju na više korisnika, predstaviti ću samo jedan i naglasiti za koje sve korisnike vrijedi.

5.1. Obrasci uporabe

- **OU1-Registracija**

- **Cilj:** Kreiranje novog korisnika
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Korisnik nije registriran
- **Rezultat:** Uspješno kreiran korisnik
- **Osnovni tijek**
 - Administrator upisuje potrebne podatke za registraciju o korisniku i odabire gumb za dodavanje novog korisnika
 - Sustav vrši provjeru unesenih podataka
 - Novi korisnik se sprema u bazu podataka i ispisuje se potvrdna poruka o kreiranju novog korisnika
- **Moguća odstupanja**
 - Administrator je unio neispravne ili nepotpune podatke

Prijava u sustav istovjetna je za svakog korisnika. Pri otvaranju početne stranice korisnik u formu za prijavu upisuje svoje korisničko ime i lozinku. Pritiskom na gumb sustav provjerava unesene podatke i ovisno o ishodu preusmjerava korisnika na stranicu kalendara ili ispisuje poruku o neuspjehu prijave.

- **OU2-Prijava**

- **Cilj:** Prijava korisnika u sustav
- **Glavni aktori:** Anonimni korisnik, baza podataka

- **Preduvjeti:** Registrirani korisnik
- **Rezultat:** Korisnik uspješno prijavljen u sustav
- **Osnovni tijek**
 - Korisnik upisuje korisničko ime i lozinku i odabire gumb za prijavu
 - Sustav vrši provjeru unesenih podataka
 - Korisnik se preusmjerava na početnu stranicu
- **Moguća odstupanja**
 - Korisnik je unio neispravne ili nepotpune podatke; sustav ispisuje poruku

Nakon uspješne prijave svaki se korisnik preusmjerava na stranicu kalendara. Na toj se stranici pregledavaju utakmice s osnovnim podacima – službenim osobama, vremenom i mjestom odigravanja, statusom potvrde službenih osoba. Odabirom drugog dana na kalendaru prikazuju se utakmice na taj dan ili poruka o nepostojanju utakmica na taj dan. Administrator, za razliku od drugih korisnika, ima dodatne opcije za delegiranje službenih osoba, pregled njihove dostupnosti za taj dan, ažuriranje podataka o utakmici i brisanju utakmice iz sustava.

- **OU3-Kalendar**

- **Cilj:** Pregled utakmica i dostupnosti službenih osoba
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Prijavljeni administrator na stranici kalendar
- **Rezultat:** Uspješno pregledavanje podataka
- **Osnovni tijek**
 - Administrator pregledava popis utakmica po danima
 - Administrator pregledava dostupnost službenih osoba
- **Moguća odstupanja**
 - Nepostojanje utakmica na taj dan

Na stranici *kalendar* za svaku utakmicu moguće je delegirati službene osobe – dva suca i delegata. Odabirom gumba *delegiraj* otvara se skočni prozor unutar kojeg administrator

odabire imena s padajuće liste za odabir sudaca i delegata. Nakon odabira imena, administrator pritiskom na gumb *delegiraj* sprema podatke o delegiranim službenim osobama za tu utakmicu.

- **OU4-Delegiranje službenih osoba**
 - **Cilj:** Delegiranje službenih osoba za utakmicu
 - **Glavni aktori:** Administrator, baza podataka
 - **Preduvjeti:** Prijavljeni administrator na stranici kalendar
 - **Rezultat:** Uspješno delegiranje službenih osoba
 - **Osnovni tijek**
 - Administrator pregledava dostupnost službenih osoba
 - Pritiskom na gumb otvara se prozor za delegiranje
 - Odabire suce i delegata s popisa
 - Potvrđuje i sprema odabir u bazu podataka
 - **Moguća odstupanja**
 - Nepostojanje utakmica na taj dan

Pregled statistike sudaca jedna je od važnijih mogućnosti ovog sustava. Odabirom određenog suca administratoru se otvara prikaz u kojem može pregledati osnovne podatke o sucu (regija, lista prosječna ocjena u sezoni) te detaljne podatke unutar sezone, natjecanja ili pojedine utakmice. Sezone i natjecanja su prikazana po njihovim atributima aktivnosti, a pojedine utakmice su na grafovima prikazane ako je za njih delegat ispunio evaluacije. Pored administratora, svoju statistiku mogu vidjeti i sami suci, ali i delegati no u smanjenom obimu jer se za delegate ne vode ocjene u sustavu.

- **OU5-Pregled sudačke statistike**
 - **Cilj:** Pregled osnovnih podataka i detaljne statistike o suđenju
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni sudac
 - **Rezultat:** Uspješno pregledavanje detaljnih podataka o suđenju
 - **Osnovni tijek**

- Sudac pregledava osnovne podatke i detaljne statistike o prosječnim ocjenama po sezoni, natjecanju ili pojedinačnoj utakmici
- **Moguća odstupanja**
 - Neodsudene utakmice u pojedinoj sezoni ili natjecanju

Administrator ima mogućnost kreiranja novih utakmica u sustavu. Odabirom gumba iz navigacijske trake odlazi na stranicu s formom za kreiranje utakmice. Tamo unosi sve potrebne podatke te odabirom gumba sprema podatke i preusmjerava se na stranicu *kalendar*.

- **OU6-Kreiranje utakmice**

- **Cilj:** Kreiranje nove utakmice
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Prijavljeni administrator
- **Rezultat:** Uspješno kreiranje nove utakmice
- **Osnovni tijek**
 - Administrator unosi potrebne podatke za utakmicu u formu
 - Sustav vrši provjeru unesenih podataka
 - Administrator se preusmjerava na stranicu *kalendar*
- **Moguća odstupanja**
 - Nepotpune informacije u formi

Administrator ima mogućnost pregleda podataka o svim bazenima i njihovom uređivanju. Na lijevoj strani ekrana nalazi se popis svih bazena te gumbi *uredite* i *izbrišite*. Ako unutar popisa ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutnu stranicu te gumbe *prethodna* i *sljedeća*. S desne strane nalazi se forma za dodavanje novog bazena ili uređenje postojećeg. Klikom gumba *uredite* forma se popunjava podacima odabranog bazena. Pri dodavanju novog ili uređivanju podataka postojećeg bazena, sustav provjerava podatke te ispisuje poruku o uspjehu ili neuspjehu, te ako je bazen uspješno dodan popis s bazenima se automatski ažurira novim vrijednostima.

- **OU7-Uređenje bazena**
 - **Cilj:** Uređenje podataka postojećih bazena
 - **Glavni aktori:** Administrator, baza podataka
 - **Preduvjeti:** Prijavljeni administrator
 - **Rezultat:** Uspješno uređeni podaci o izabranom bazenu
 - **Osnovni tijek**
 - Administrator odabire bazen čije podatke želi urediti
 - Administrator unosi potrebne podatke za bazen u formu
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu
 - **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Administrator ima mogućnost pregleda podataka o svim sezonama i njihovom uređivanju. Na lijevoj strani ekrana nalazi se popis svih sezona te gumbi *uredite* i *izbrišite*. Ako unutar popisa ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutnu stranicu te gumbe *prethodna* i *sljedeća*. S desne strane nalazi se forma za dodavanje nove sezone ili uređenje postojeće. Klikom gumba *uredite* forma se popunjava podacima odabrane sezone. Pri dodavanju nove ili uređivanju podataka postojeće sezone, sustav provjerava podatke te ispisuje poruku o uspjehu ili neuspjehu, te ako je sezona uspješno dodana popis sa sezonama se automatski ažurira novim vrijednostima.

- **OU8-Uređenje sezone**
 - **Cilj:** Uređenje podataka postojećih sezona
 - **Glavni aktori:** Administrator, baza podataka
 - **Preduvjeti:** Prijavljeni administrator
 - **Rezultat:** Uspješno uređeni podaci o izabranoj sezoni

- **Osnovni tijek**
 - Administrator odabire sezonu čije podatke želi urediti
 - Administrator unosi potrebne podatke za sezonu u formu
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu
- **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Administrator ima mogućnost pregleda podataka o svim klubovima i njihovom uređivanju. Na lijevoj strani ekrana nalazi se popis svih klubova te gumbi *uredite* i *izbrišite*. Ako unutar popisa ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutnu stranicu te gumbe *prethodna* i *sljedeća*. S desne strane nalazi se forma za dodavanje novog kluba ili uređenje postojećeg. Klikom gumba *uredite* forma se popunjava podacima odabranog kluba. Pri dodavanju novog ili uređivanju podataka postojećeg kluba, sustav provjerava podatke te ispisuje poruku o uspjehu ili neuspjehu, te ako je klub uspješno dodan popis s klubovima se automatski ažurira novim vrijednostima.

- **OU9-Uređenje kluba**

- **Cilj:** Uređenje podataka postojećih klubova
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Prijavljeni administrator
- **Rezultat:** Uspješno uređeni podaci o izabranom klubu
- **Osnovni tijek**
 - Administrator odabire klub čije podatke želi urediti
 - Administrator unosi potrebne podatke za klub u formu
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu

- **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Administrator ima mogućnost pregleda podataka o svim korisnicima i njihovom uređivanju. Na vrhu ekrana nalazi se navigacijska traka kojom administrator filtrira podatke između službenih osoba i klupskih osoba. Ako su odabrane službene osobe, prikazana su dva popisa. Jedan za suce, a drugi za delegate. Svaki element ima gumbе *uredite* i *izbrišite*. Ako unutar popisa ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutachnu stranicu te gumbе *prethodna* i *sljedeća*. S desne strane nalazi se forma za dodavanje novog korisnika ili uređenje postojećeg. Klikom gumba *uredite* forma se popunjava podacima odabranog korisnika. Ako je riječ o sucima ili delegatima, forma se prilagođava njihovim podacima i izostavlja atribute vezane za klupske osobe, i obratno. Pri dodavanju novog ili uređivanju podataka postojećeg korisnika, sustav provjerava podatke te ispisuje poruku o uspjehu ili neuspjehu, te ako je korisnik uspješno dodan popis s korisnicima se automatski ažurira novim vrijednostima.

- **OU10-Uređenje korisnika**

- **Cilj:** Uređenje podataka postojećih korisnika
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Prijavljeni administrator
- **Rezultat:** Uspješno uređeni podaci o izabranom korisniku
- **Osnovni tijek**
 - Administrator odabire korisnika čije podatke želi urediti
 - Administrator unosi potrebne podatke za korisnika u formu
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu
- **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Omogućen je i pregled klupske statistike koja sadrži pregled natjecanja unutar kojih se klub natjecao u svakoj sezoni. S lijeve strane se nalazi popis svih klubova i gumb *detalji*. Ako unutar popisa ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutnu stranicu te gumb *prethodna* i *sljedeća*. Odabirom određenog kluba, s desne strane ekrana se prikazuje element s podacima o sezonama u kojima se klub natjecao te popisom natjecanja unutar te sezone.

- **OU11-Klupska statistika**

- **Cilj:** Pregled klupske statistike
- **Glavni aktori:** Administrator, baza podataka
- **Preduvjeti:** Prijavljeni administrator
- **Rezultat:** Uspješno pregledani podaci o klupskoj statistici
- **Osnovni tijek**
 - Administrator odabire klub čije podatke želi pregledati
- **Moguća odstupanja**
 - Nepostojanje podataka za taj klub

Administrator može pregledati podatke o svim natjecanjima i urediti ih. Na lijevoj strani ekrana nalazi se popis svih natjecanja grupiranih po sezonama te gumbi *uredite* i *izbrišite*. Ako unutar popisa po sezoni ima više od 5 elemenata, na dnu popisa se pojavljuje navigacija koja prikazuje 5 odabranih elemenata. Navigacija prikazuje trenutnu stranicu te gumb *prethodna* i *sljedeća*. S desne strane nalazi se forma za dodavanje novog natjecanja ili uređenje postojećeg. Klikom gumba *uredite* forma se popunjava podacima odabranog natjecanja. Pri dodavanju novog ili uređivanju podataka postojećeg natjecanja, sustav provjerava podatke te ispisuje poruku o uspjehu ili neuspjehu, te ako je natjecanje uspješno dodano popis s natjecanjima se automatski ažurira novim vrijednostima. Unutar forme za dodavanje samog natjecanja, moguće je dinamički dodavati i kola za to natjecanje.

- **OU12-Kreiranje natjecanja**

- **Cilj:** Kreiranje novog natjecanja s kolima
- **Glavni aktori:** Administrator, baza podataka

- **Preduvjeti:** Prijavljeni administrator
- **Rezultat:** Uspješno kreirano novo natjecanje
- **Osnovni tijek**
 - Administrator unosi potrebne podatke za natjecanje u formu
 - Administrator unosi podatke za nova kola za natjecanje
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu
- **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Svaki korisnik može pregledati i urediti svoje osobne podatke. Odabirom gumba *Moj profil* na navigacijskoj traci otvara se ekran s popunjenom formom korisnikovih podataka. Korisnik ih može izmijeniti i spremi u bazu podataka.

- **OU13-Uređivanje osobnih podataka**
 - **Cilj:** Uređivanje osobnih podataka
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni korisnik
 - **Rezultat:** Uspješno uređeni osobni podaci
 - **Osnovni tijek**
 - Korisnik mijenja potrebne podatke
 - Sustav vrši provjeru unesenih podataka
 - Ispisuje se poruka o uspjehu
 - **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Nakon što je administrator delegirao službene osobe za sve utakmice u pojedinom kolu, administrator može to isto delegiranje poslati svim klubovima i službenim osobama dionicima određenog kola unutar natjecanja. Odlaskom na stranicu *Delegiranja*, administratoru se prikazuje popis svih kola grupiranih po natjecanjima i sezonama. Pored svakog kola nalaze se gumbi *pregled*, *preuzimanje*, *slanje*. Administrator može prije slanja pregledati dokument u prozoru na desnoj strani ekrana, te ako je sve u redu odabrati opciju *slanje*. Potom mu se otvara prozor unutar kojeg uređuje primatelje maila s delegiranjem. Tu su već unaprijed definirani svi klubovi koji igraju utakmice tog kola, službene osobe koje su delegirane, te odjeljak unutar kojeg se može ručno dodati adresa dodatnog primatelja. Na sljedećoj stranici administrator uređuje podatke o mailu – naslov, tijelo i privitke. Ako su svi podaci ispravno uneseni ispisuje se poruka o uspjehu slanja maila.

- **OU14-Slanje delegiranja**
 - **Cilj:** Slanje delegiranja za kolo svim dionicima
 - **Glavni aktori:** Administrator, baza podataka
 - **Preduvjeti:** Prijavljeni administrator
 - **Rezultat:** Uspješno poslan mail s informacijama o delegiranju
 - **Osnovni tijek**
 - Administrator unosi potrebne podatke o primateljima delegiranja
 - Administrator unosi podatke o mailu
 - Ispisuje se poruka o uspjehu
 - **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Suci i delegat koji su bili dionici utakmice te administrator mogu pregledati podatke o utakmici, disciplinske prijave vezane za tu utakmicu i evaluacije koje im je delegat dodijelio. Nakon što je utakmica završila, u popisu utakmica korisnicima se pojavljuje opcija *detalji* koja vodi na stranicu s detaljima o utakmici. Tamo korisnici imaju mogućnost pregledati evaluacije i disciplinske prijave, ili ih kreirati. Samo delegat može ispuniti evaluaciju za suce, dok ih ostali korisnici s pristupom mogu pregledavati.

- **OU15-Pregled dokumenata**
 - **Cilj:** Pregled evaluacija i disciplinskih prijava
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni sudac
 - **Rezultat:** Uspješno pregledane evaluacije i eventualne disciplinske prijave
 - **Osnovni tijek**
 - Sudac pregledava dokumente
 - **Moguća odstupanja**
 - Nepostojanje dokumenata o utakmici

Svi suci i delegati mogu za pojedini dan unutar kalendara javiti nedostupnost za delegiranje. Odabirom gumba *moja dostupnost* prikazuje se stranica s kalendarom i dijelom gdje se može mijenjati dostupnost za taj određeni dan iz kalendara. Za dane koji su u prošlosti, nije moguće naknadno mijenjati dostupnost.

- **OU16-Prijava nedostupnosti**
 - **Cilj:** Prijava nedostupnosti za određeni dan
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni sudac
 - **Rezultat:** Uspješno prijavljena nedostupnost za delegiranje
 - **Osnovni tijek**
 - Sudac pregledava kalendar
 - Odabire dan za kojeg želi prijaviti nedostupnost
 - Pritiskom gumba *promijenite dostupnost* postavlja dostupnost za taj dan
 - **Moguća odstupanja**
 - -

Svi suci i delegati mogu za pojedinu utakmicu potvrditi ili otkazati delegiranje. Odabirom gumba *potvrdi* ili *otkaži* mijenja se status za tu utakmicu. Nakon promjene, ona je vidljiva na popisu utakmica i administratoru i ostalim korisnicima. Jednom otkazana utakmica se više ne može potvrditi.

- **OU17-Mijenjanje statusa**
 - **Cilj:** Potvrđivanje ili otkazivanje utakmice
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni sudac
 - **Rezultat:** Uspješno potvrđena ili otkazana utakmica
 - **Osnovni tijek**
 - Sudac pregledava popis svojih utakmica
 - Odabire jednu od opcija za otkaz ili potvrdu utakmice
 - **Moguća odstupanja**
 - -

Evaluaciju može ispuniti samo delegat, dok je pregledati mogu svi prijavljeni korisnici koji su bili dionici utakmice i administrator. Evaluacija sadrži podatke o utakmici koje nije moguće mijenjati te atribute o suđenju podijeljene u sedam kategorija. Ocjenjuje se opće ponašanje, primjena prekršaja, sudačka tehnika, kontrola utakmice, protokol i ponašanje, tehničke pogreške te slobodne bilješke o dojmu i pogreškama te prijedlozi za buduća delegiranja. Da bi se evaluacija spremila moraju biti uneseni svi podaci.

- **OU18-Ispunjavanje evaluacije**
 - **Cilj:** Evaluirati suca za određenu utakmicu
 - **Glavni aktori:** Delegat, baza podataka
 - **Preduvjeti:** Prijavljeni delegat koji je dionik utakmice
 - **Rezultat:** Uspješno evaluiran sudac
 - **Osnovni tijek**
 - Delegat odabire za kojeg suca želi ispuniti evaluaciju

- Delegat unosi sve potrebne podatke u formu
- Ispisuje se poruka o uspjehu
- **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Disciplinsku prijavu mogu ispuniti svi dionici utakmice i administrator. Disciplinska prijava sadrži podatke o utakmici koje nije moguće mijenjati te podatke o prijavitelju, prijavljenom te opcionalno dodatne izjave. Da bi se disciplinska prijava spremila moraju biti uneseni svi podaci.

- **OU19-Ispunjavanje disciplinske prijave**
 - **Cilj:** Ispuniti disciplinsku prijavu
 - **Glavni aktori:** Sudac, baza podataka
 - **Preduvjeti:** Prijavljeni korisnik koji je dionik utakmice
 - **Rezultat:** Uspješno ispunjena disciplinska prijava
 - **Osnovni tijek**
 - Sudac unosi sve potrebne podatke u formu
 - Prijavljena osoba unosi svoju izjavu
 - Opcionalno se unose dodatne izjave i podaci o osobama
 - Ispisuje se poruka o uspjehu
 - **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Klupske osobe, pored već navedenih i prije objašnjenih zajedničkih funkcionalnosti za sve korisnike, imaju mogućnost prijave satnice već kreiranih utakmica za svoj klub koji igra domaće utakmice. Opciji za prijavu satnice pristupaju odabirom gumba *satnica* na popisu utakmica gdje su prikazane samo njihove domaće utakmice.

- **OU20-Prijava satnice**
 - **Cilj:** Prijaviti satnicu domaće utakmice
 - **Glavni aktori:** Klupska osoba, baza podataka
 - **Preduvjeti:** Prijavljena klupska osoba
 - **Rezultat:** Uspješno prijavljena satnica domaće utakmice
 - **Osnovni tijek**
 - Klupska osoba odabire popis domaćih utakmica
 - Klupska osoba odabire za koju utakmicu želi prijaviti satnicu
 - Unosi satnicu i predaje podatke na spremanje
 - Ispisuje se poruka o uspjehu
 - **Moguća odstupanja**
 - Nepotpune informacije u formi
 - Ispisuje se poruka o neuspjehu

Zaključak

Cilj ovog diplomskog rada bio je izrada web-aplikacije koja služi kao administracijski sustav koji omogućuje efikasno vođenje vaterpolskog natjecanja i praćenja rada službenih osoba. Sustav omogućuje digitalizaciju i automatizaciju ključnih procesa za efikasno provođenje natjecanja, poput javljanja dostupnosti za natjecanja, delegiranja službenih osoba i kreiranje dokumenata za utakmicu. Osim toga omogućuje praćenje angažmana i ocjena sudaca, što doprinosi objektivnijoj evaluaciji njihovog rada. Nakon stvaranja dokumenta delegiranja, istog je moguće poslati svih službenim osobama i klubovima kroz jednostavno korisničko sučelje, čime se postiže optimizacija provođenja sustava delegiranja i skraćivanje vremena potrebnog za obavljanje ručnog posla.

Za nastavak razvoja aplikacije, u postojeći obim funkcionalnosti, može se dodati dodatna razina administratora koja bi bila zadužena samo za pojedino natjecanje. Postojeće statističke podatke može se prilagoditi izvozu u različite oblike, kako bi se ukupna statistika objedinila na jednom mjestu i omogućilo stvaranje poretka po kvaliteti. Također je moguće uvođenje dodatnih obavijesti slanjem maila pri promjenama u delegiranjima.

Literatura

- [1] Vue.js: The progressive JavaScript Framework, vuejs.org, Poveznica: <https://vuejs.org>; pristupljeno 1. veljače 2025.
- [2] ASP.NET Core: Open-source web framework for .NET, dotnet.microsoft.com, Poveznica: <https://dotnet.microsoft.com/en-us/apps/aspnet>; pristupljeno 1. veljače 2025.
- [3] Entity Framework Core: Overview of Entity Framework Core – EF Core, learn.microsoft.com, Poveznica: <https://learn.microsoft.com/en-us/ef/core/>; pristupljeno 1. veljače 2025.
- [4] PostgreSQL: The world's most advanced open source database, postgresql.org, Poveznica: <https://www.postgresql.org/>; pristupljeno 1. veljače 2025.
- [5] Pinia: The intuitive store for Vue.js, pinia.vuejs.org, Poveznica: <https://pinia.vuejs.org>; pristupljeno 1. veljače 2025.
- [6] Vue-toastification: Vue notifications made easy!, github.com, <https://github.com/Maronato/vue-toastification/tree/next>; pristupljeno 1. veljače 2025.
- [7] Axios, axios-http.com, Poveznica: <https://axios-http.com/>; pristupljeno 1. veljače 2025.
- [8] JWT: JSON Web Tokens, jwt.io, Poveznica: <https://jwt.io/>; pristupljeno 1. veljače 2025.
- [9] Client-server model, researchgate.net, Poveznica: https://www.researchgate.net/publication/271295146_Client-Server_Model; pristupljeno 14. veljače 2025.
- [10] REST API, researchgate.net, Poveznica: https://www.researchgate.net/publication/303515127_REST_APIs_A_Large-Scale_Analysis_of_Compliance_with_Principles_and_Best_Practices/references; pristupljeno 14. veljače 2025.

Sažetak

Sustav praćenja rada službenih osoba u vaterpolu

Tema ovog diplomskog rada je izrada web-aplikacije koja služi kao administracijski sustav koji omogućuje efikasno vođenje vaterpolskog natjecanja i praćenja rada službenih osoba. Uvodni dio opisuje probleme koji će se riješiti ovim sustavom. Zatim se navode i pojašnjavaju korištene tehnologije. U drugom poglavlju objašnjava se arhitektura aplikacije – baza podataka, klijentska i poslužiteljska aplikacija. Treće poglavlje bavi se korisničkim sučeljem i prikazom različitih mogućih funkcionalnosti u ovisnosti o tipu korisnika koji koristi sustav.

Ključne riječi: web-aplikacija, Vue.js, ASP.NET Core, PostgreSQL, vaterpolo

Summary

System for monitoring the work of officials in water polo

The topic of this thesis is the development of a web application that serves as an administrative system, enabling efficient management of water polo competitions and monitoring the work of officials. The introductory section describes the problems that this system aims to solve. Then, the technologies used are listed and explained. The second chapter covers the application architecture, including the database, client and server applications. The third chapter focuses on the user interface and presents various functionalities depending on the type of user utilizing the system.

Keywords: web application, Vue.js, ASP.NET Core, PostgreSQL, water polo