

# Programsko inženjerstvo

Ak. god. 2020./2021.

## FringillaSport

Dokumentacija, Rev. 2

Grupa: *Fringilla*

Voditelj: *Lovro Nuić*

Datum predaje: 14. 01. 2021.

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Postojeća slična rješenja . . . . .	7
2.2	Moguće nadogradnje projektnog zadatka . . . . .	8
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>10</b>
3.1	Funkcionalni zahtjevi . . . . .	10
3.1.1	Obrasci uporabe . . . . .	13
3.1.2	Sekvencijski dijagrami . . . . .	24
3.2	Ostali zahtjevi . . . . .	28
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>29</b>
4.1	Baza podataka . . . . .	31
4.1.1	Opis tablica . . . . .	31
4.2	Dijagram baze podataka . . . . .	38
4.3	Dijagram razreda . . . . .	39
4.4	Dijagram stanja . . . . .	47
4.5	Dijagram aktivnosti . . . . .	48
4.6	Dijagram komponenti . . . . .	49
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>51</b>
5.1	Korištene tehnologije i alati . . . . .	51
5.2	Ispitivanje programskog rješenja . . . . .	52
5.2.1	Ispitivanje komponenti . . . . .	52
5.2.2	Ispitivanje sustava . . . . .	54
5.3	Dijagram razmještaja . . . . .	65
5.4	Upute za puštanje u pogon . . . . .	66
5.4.1	Instalacija postgresql baze podataka . . . . .	66
5.4.2	Instalacija drugih potrebnih paketa . . . . .	66
5.4.3	Pokretanje backenda . . . . .	66

5.4.4	Pokretanje frontenda . . . . .	67
5.4.5	Pokretanje servera za predlaganje sporta . . . . .	67
<b>6</b>	<b>Zaključak i budući rad</b>	<b>68</b>
	<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>70</b>

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Nuić	15.10.2020.
0.2	Dodani dionici i funkcionalni zahtjevi trenera.	Marfat	16.10.2020.
0.3	Dodani funkcionalni zahtjevi neprijavljenog korisnika i sportaša i dodan dio obrazaca uporabe.	Rašić	16.10.2020.
0.4	Uređeni funkcionalni zahtjevi trenera zbog konzistentnosti, dodani funkcionalni zahtjevi baze podataka i dio obrazaca uporabe.	Marfat	17.10.2020.
0.5	Dodani funkcionalni zahtjevi i obrasci uporabe za iznajmljivača i administratora.	Ilić	17.10.2020.
0.6	Dodani ostali zahtjevi i glavni sudionik u UC20 promijenjen u "Trener". Dodan sekvencijski dijagram za UC6 i postavljen defaultni direktorij za slike u folder "slike". Dodani ostali sekvencijski dijagrami.	Marfat	31.10.2020.
0.7	Promijenjeno "Korisnik" u "Sportaš" u obrascima uporabe nakon prijave	Crnogorac	31.10.2020.
0.8	U opisu projekta dodana postojeća slična rješenja i moguće nadogradnje projektnog zadatka	Rašić	31.10.2020.
0.9	Dodani UML dijagrami u PNG formatu u direktorij slike te u LaTeX dokument	Crnogorac	31.10.2020.
0.10	Dodan opis projektnog zadatka	Paradžik	1.11.2020.

Rev.	Opis promjene/dodatka	Autori	Datum
0.11	Dodan opis baze podataka i relacija	Srdarev	2.11.2020.
0.12	Dodan dijagram baze podataka	Srdarev	4.11.2020.
0.13	Promijenjen UC10	Marfat	5.11.2020.
0.14	Ažuriran dnevnik sastajanja	Rašić	8.11.2020
0.15	Restrukturirana baza podataka	Srdarev	8.11.2020.
0.16	Ispravak gramatičkih i pravopisnih grešaka	Paradžik	8.11.2020
0.17	Ažuriran dnevnik sastajanja	Rašić	13.11.2020
0.18	Uređen stil tablica baze podataka	Nuić	13.11.2020
0.19	Dijagrami razreda	Nuić	13.11.2020
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Nuić	13.11.2020.
1.1	Dijagram stanja	Paradžik	10.01.2021.
1.2	Dijagram aktivnosti	Paradžik	11.01.2021.
1.3	Ispravak dokumentacije iz 1. ciklusa	Nuić	13.01.2021.
1.4	Osvježavanje dijagrama baze podataka	Nuić	14.01.2021.
1.5	Zaključak i budući rad	Paradžik	14.01.2021.
1.6	Dodan dijagram komponenti i dijagram razmještaja	Crnogorac	14.01.2020.
1.7	Dodani dijagrami razreda	Nuić	14.01.2021.
1.8	Dodane upute za pokretanje	Nuić	14.01.2021.
2.0	Završna datoteka za predaju	Nuić	14.01.2021.

## 2. Opis projektnog zadatka

Cilj ovog projekta je timskim radom i korištenjem našeg znanja oblikovanja programske potpore razviti web aplikaciju "FringillaSport". Cilj aplikacije "FringillaSport" je omogućiti jednostavnije povezivanje sportaša, trenera i iznajmljivača sportskih prostora te taj proces pojednostavniti i ubrzati.

Aplikaciju mogu koristiti i registrirani i neregistrirani korisnici. Registrirani korisnici mogu biti:

- Administrator
- Iznajmljivač
- Trener
- Sportaš

Pri registriranju korisnik bira ulogu Sportaša, Trenera ili Iznajmljivača. Za registraciju Sportaša, Trenera i Iznajmljivača su potrebni sljedeći podaci:

- ime
- prezime
- email
- korisničko ime
- zaporku

Za registraciju Trenera potrebno je još priložiti i službenu dokumentaciju koja potvrđuje da je ta osoba službeni trener za neke sportove (takvu prijavu odobrava administrator koji provjerava vjerodostojnost informacija).

Mogućnosti korisnika:

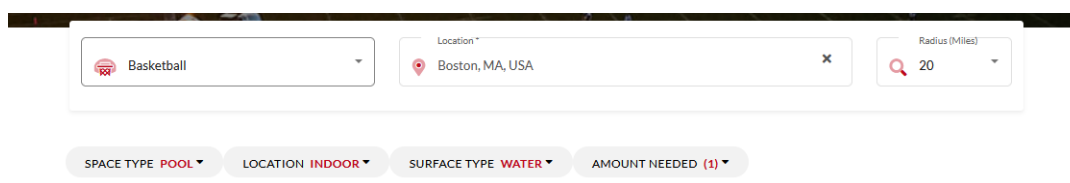
- Neregistrirani korisnik:
  - Pristup stranici te pristup tražilici sportskih okupljanja/treninga. (Korisnik ima samo pregled, ne može se prijaviti na sportsko okupljanje/-trening.)
- Sportaš:
  - Pristup tražilici sportskih okupljanja/treninga.

- Prijava za pojedino sportsko okupljanje/trening unutar tražilice sportskih okupljanja/treninga.
- Vidljivost svih prijavljenih događaja u vlastitom kalendaru sportskih okupljanja/treninga.
- Dobivanje preporuke za odabir sporta u kojem bi imao najveće mogućnosti ostvariti izvrsne rezultate na natjecanjima. Preporuka se temelji na visini, težini, broju godina i ostalih osobnih podataka.
- Može zatražiti ulogu trenera ako dobije službeni certifikat.
- Iznajmljivač:
  - Može imati odvojeni račun trenera/sportaša.
  - Može u uređivaču sportskih objekata dodati svoj sportski objekt navodeći lokaciju, tip sportova koji se mogu u objektu održavati, cijenu po satu, termine dostupne za rezervaciju i dokumentaciju koja potvrđuje da je on vlasnik. (Takvu prijavu administrator provjerava i potvrđuje ako je sve ispravno.)
  - U svom kalendaru može vidjeti trenutno odobrene termine za iznajmljivanje i nove termine za koje je netko od korisnika zatražio iznajmljivanje te takve termine odobriti ili odbaciti.
- Trener:
  - Može naknadno prijaviti dodatnu službenu dokumentaciju. (Za neki drugi sport ili slično.)
  - Mogućnost organiziranja i naplaćivanja treninga u sportovima za koje ima potvrđenu dokumentaciju. Trening se može organizirati u uređivaču treninga po istom principu kao što se organizira sportsko okupljanje.
  - Nasljeđuje prava sportaša te može organizirati i sportska okupljanja za ostale sportove.
  - Uz kalendar sportskih okupljanja na kojima on prisustvuje izvan profesionalnih treninga, ima i pristup kalendaru svih svojih termina treninga te podatke o njima. ime
- Administrator:
  - Upravlja sustavom.
  - Odobrava i provjerava istinitost podataka iznajmljivača sportskih objekata i trenera koji se žele registrirati u aplikaciju.
  - Može brisati ili dodavati druge korisnike te im mijenjati uloge. (Bri-

sanjem određenog korisnika, brišu se sve njegove prijave na sportska okupljanja i/ili svi termini treninga kreirani od tog korisnika ili/i svi sportski objekti kreirani.)

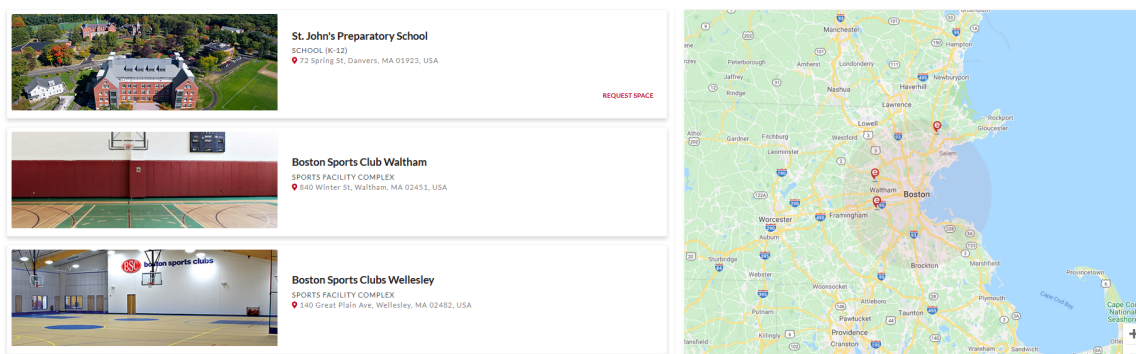
## 2.1 Postojeća slična rješenja

Aplikacija slična aplikaciji koju mi razvijamo je playeasy<sup>1</sup>. Playeasy je aplikacija dizajnirana za pretraživanje i iznajmljivanje sportskih objekata ili površina. Pretraživanje je puno detaljnije za razliku od aplikacije koju mi razvijamo s mogućnostima sužavanja rezultata na temelju sporta, tipa prostora (dvorana, auditorij, teretana...), tipa površine (beton, drvo, zemlja...) i drugih.



Slika 2.1: playeasy - Pretraživanje

Nakon pretraživanja se otvara popis svih dostupnih objekata i površina, te karta s njihovim lokacijama.



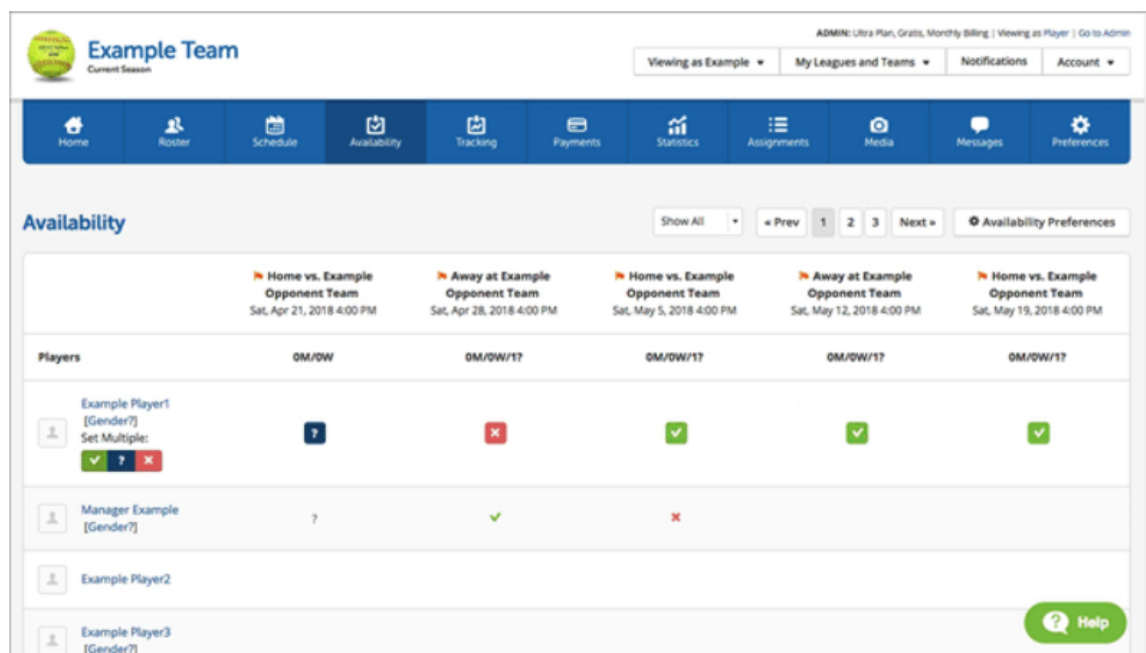
Slika 2.2: playeasy - Prikaz dostupnih sportskih objekata i površina

Najveća razlika između naše aplikacije i playeasy je što playeasy povezuje samo organizatore sportskih aktivnosti i iznajmljivače, a naša aplikacija povezuje organizatore s iznajmljivačima i sportašima.

<sup>1</sup><https://www.playeasy.com/>



Druga slična aplikacija je teamsnap<sup>2</sup>. Ova aplikacija se fokusira na povezivanje trenera i igrača, te olakšavanje komunikacije između njih. Nema mogućnosti iznajmljivanja sportskih objekata ili površina. Aplikacija sadrži mogućnosti potvrđivanja može li netko doći na trening, grupni chat između članova sportskih ekipa, plaćanje treninga i slične mogućnosti.



Slika 2.3: teamsnap - Dostupnost igrača za pojedine sportske događaje

Ostale aplikacije slične našoj su uniquevenues<sup>3</sup> i SchoolHire<sup>4</sup>. Obje ove aplikacije se fokusiraju na iznajmljivanje sportskih površina ili objekata poput aplikacije playeasy, te nemaju mogućnost organiziranja sportskih okupljanja ili treninga.

## 2.2 Moguće nadogradnje projektnog zadatka

Moguća nadogradnja ovog projekta bi bila mogućnost stvaranja sportskih timova. Članovi jednog tima bi mogli međusobno komunicirati, organizirati se i prijavljivati na sportske aktivnosti, te slati upute drugim timovima žele li s njima igrati.

Druga mogućnost bi bila organizacija natjecanja, na koja bi se mogli prijavljivati pojedinci ili timovi ovisno o sportu. Za natjecanje bi se mogao definirati raspored

<sup>2</sup><https://www.teamsnap.com/>

<sup>3</sup><https://www.uniquevenues.com/>

<sup>4</sup><https://schoolhire.co.uk/>

koji bi svi mogli pregledavati, a organizator uređivati.

Još jedna nadogradnja bi mogla biti mogućnost ocjenjivanja trenera, sportskih objekata i površina. Svatko tko je prisustvovao treningu kod nekog trenera ili sportskoj aktivnosti u nekom sportskom objektu ili površini bi mogao ostaviti kratki komentar i ocjenu o tom treneru, sportskom objektu ili površini.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Sportaš
2. Trener
3. Iznajmljivač
4. Administrator
5. Razvojni tim

**Aktori i njihovi funkcionalni zahtjevi:**

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
  - (a) koristiti tražilicu sportskih okupljanja/treninga
  - (b) pregledati na karti lokacije sportskih okupljanja/treninga
  - (c) odabrati sportsko okupljanje/trening i dobiti prikaz informacija o tom sportskom okupljanju/treningu (datum, vrijeme, mjesto, tip sporta)
  - (d) se registrirati u sustav kao sportaš, trener ili iznajmljivač, za što mu je potrebno ime, prezime, e-mail adresa i lozinka
2. Sportaš (inicijator) može:
  - (a) se prijaviti u aplikaciju
  - (b) pregledavati i mijenjati osobne podatke
  - (c) se prijaviti za pojedino sportsko okupljanje/trening
  - (d) pristupiti kalendaru svih sportskih okupljanja/treninga na koje se prijavio ili koje je organizirao
  - (e) organizirati sportsko okupljanje za što su mu potrebni datum, vrijeme i mjesto okupljanja, tip sporta te maksimalan broj sudionika
  - (f) slati upit iznajmljivačima za iznajmljivanje sportskih površina/dvorana
  - (g) pregledavati sportska okupljanja/treninge predložene od strane aplikacije na temelju sportaševih osobnih podataka

- (h) zatražiti da mu se dodijeli uloga trenera

3. Trener (inicijator) može:

- (a) se prijaviti u aplikaciju
- (b) priložiti službenu dokumentaciju kojom potvrđuje da je trener za određene sportove
- (c) pregledavati i mijenjati osobne podatke
- (d) se prijaviti za pojedino sportsko okupljanje/trening
- (e) slati upit iznajmljivačima za iznajmljivanje sportskih površina/dvorana
- (f) organizirati sportska okupljanja za sve sportove i profesionalne (plaćene) treninge za one sportove za koje je potvrđen kao trener za što su mu potrebni datum, vrijeme, mjesto okupljanja, tip sporta te maksimalan broj sudionika
- (g) pristupiti kalendaru u kojemu se nalaze sva sportska okupljanja kojima trener prisustvuje i svi njegovi termini treninga i podaci o tim treninzima
- (h) koristiti tražilicu sportskih okupljanja/treninga

4. Iznajmljivač (inicijator) može:

- (a) se prijaviti u aplikaciju
- (b) pristupiti uređivaču sportskih objekata u kojemu može mijenjati podatke i termine za već dodane sportske objekte ili dodati novi objekt navodeći lokaciju, tip sportova koji se tamo mogu održavati, cijenu po satu, termine dostupne za rezervaciju i dokumentaciju kojom potvrđuje vlasništvo
- (c) pristupiti kalendaru gdje su mu vidljivi svi odobreni termini za iznajmljivanje i novi termini koje su zatražili korisnici koje može odobriti ili odbaciti

5. Administrator (inicijator) može:

- (a) se prijaviti u aplikaciju
- (b) vidjeti sve registrirane korisnike i njihove podatke
- (c) odobriti ili odbiti registracije trenera i iznajmljivača na temelju dokumentacije koju prilože
- (d) korisnike dodavati, brisati ili im mijenjati razinu pristupa aplikaciji(sportaš, trener, iznajmljivač)

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o sportskim okupljanjima i profesionalnim treninzima
- (c) pohranjuje sve podatke o sportskim objektima koji su prijavljeni u aplikaciji

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 -Pregled sportskih aktivnosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati dostupna sportska okupljanja i treninge
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Karta i tražilica su prikazani prilikom učitavanja aplikacije
  2. Korisnik na karti odabire sportsko okupljanje ili trening
  3. Prikazuju se informacije o odabranom sportskom okupljanju/treningu

##### UC2 - Pretraživanje sportskih aktivnosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Pretraživati dostupna sportska okupljanja i treninge
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Karta i tražilica su prikazani prilikom učitavanja aplikacije
  2. Korisnik u tražilicu upisuje parametre (mjesto, vrijeme, sport)
  3. Karta se ažurira na temelju unesenih parametara

##### UC3 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi tražene podatke
  3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzete e-mail adrese, upis podataka u nedozvoljenom formatu, upis neispravne e-mail adrese

1. Sustav obavještava korisnika o neuspjeloj registraciji i vraća ga na stranicu za registraciju
2. Korisnik mijenja neispravno upisane podatke ili odustaje od registracije

#### UC4 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu
  2. Korisnik unosi e-mail adresu i lozinku
  3. Potvrda o ispravnosti unesenih podataka
  4. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Neispravna e-mail adresa ili lozinka
    1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga na stranicu za prijavu
    2. Korisnik mijenja neispravno unesene podatke ili odustaje od prijave

#### UC5 -Pregled osobnih podataka

- **Glavni sudionik:** Sportaš
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Osobni podatci"
  2. Prikazuju se osobni podatci korisnika

#### UC6 -Promjena osobnih podataka

- **Glavni sudionik:** Sportaš
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i odabrao je opciju "Osobni podatci"
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju za promjenu osobnih podataka
2. Korisnik mijenja osobne podatke
3. Korisnik sprema promjene

#### **UC7 -Prijava na sportsku aktivnost**

- **Glavni sudionik:** Sportaš
- **Cilj:** Prijaviti se na sportsko okupljanje ili trening
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire sportsku aktivnost na koju se želi prijaviti
  2. Korisnik odabire opciju "Prijavi se"
  3. Korisnik potvrđi da se želi prijaviti za odabranu sportsku aktivnost

#### **UC8 -Pregled prijavljenih sportskih aktivnosti**

- **Glavni sudionik:** Sportaš
- **Cilj:** Pregledati sportska okupljanja ili treninge na koje se korisnik prijavio ili koje je organizirao
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moje prijave"
  2. Prikazuje se kalendar s upisanim terminima svih prijavljenih ili organiziranih sportskih aktivnosti

#### **UC9 -Organiziranje sportskog okupljanja**

- **Glavni sudionik:** Sportaš
- **Cilj:** Organizirati sportsko okupljanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Organiziraj sportsko okupljanje"
  2. Korisnik upisuje tražene podatke
  3. Korisnik potvrđuje unesene podatke
  4. Korisnika se preusmjerava na stranicu za pregled prijavljenih sportskih aktivnosti



- **Opis mogućih odstupanja:**

- 2.a Upisano mjesto sportskog okupljanja nije javno i korisnik ga nije iznajmio

- 1. Sustav obavještava korisnika da nije iznajmio sportsku površinu/dvoranu te mu daje opciju za iznajmljivanje
    - 2. Korisnik može promijeniti tražene podatke, odabrati opciju za iznajmljivanje ili odustati od organizacije

### UC10 -Iznajmljivanje sportskih površina/dvorana

- **Glavni sudionik:** Sportaš
- **Cilj:** Iznajmljivanje sportske površine ili dvorane
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i odabrao je opciju "Organiziraj sportsko okupljanje"
- **Opis osnovnog tijeka:**
  - 1. Korisnik odabire opciju "Iznajmljivanje sportske površine ili dvorane"
  - 2. Korisnik odabire sportsku površinu ili dvoranu koju želi iznajmiti
  - 3. Korisnik odabire datum kada želi iznajmiti odabranu sportsku površinu ili dvoranu
  - 4. Korisniku se prikazuju slobodni termini za odabrani datum
  - 5. Korisnik odabire željeno vrijeme
  - 6. Korisnik odabire opciju "Potvrdi rezervaciju"
- **Opis mogućih odstupanja:**
  - 3.a Za odabrani datum nema slobodnih termina
    - 1. Sustav javlja korisniku da za odabrani datum nema slobodnih termina i daje mu opciju promjene datuma

### UC11 -Pregled predloženih aktivnosti

- **Glavni sudionik:** Sportaš
- **Cilj:** Pregled sportskih aktivnosti predloženih od strane sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  - 1. Korisnik odabire opciju "Predloži za mene"
  - 2. Sustav prikazuje predložena sportska okupljanja i treninge na temelju osobnih podataka korisnika

**UC12 - Dodavanje sportskog objekta**

- **Glavni sudionik:** Iznajmljivač
- **Cilj:** Dodati novi sportski objekt
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Korisnik je prijavljen s ovlastima iznajmljivača
- **Opis osnovnog tijeka:**
  1. Iznajmljivač odabire opciju "Dodaj novi sportski objekt"
  2. Iznajmljivač ispunjava podatke za objekt i prilaže dokumentaciju o vlasništvu
  3. Administrator pregledava i potvrđuje dokumentaciju
  4. Iznajmljivaču se prikazuje poruka o uspješnom dodavanju objekta
- **Opis mogućih odstupanja:**
  - 3.a Administrator odbija dodavanje objekta zbog nevažeće dokumentacije
    1. Sustav obavještava iznajmljivača o neuspješnom dodavanju objekta
    2. Iznajmljivač može odustati ili ponovno dodati objekt s važećom dokumentacijom

**UC13 - Pregled sportskog objekta**

- **Glavni sudionik:** Iznajmljivač
- **Cilj:** Pregledati podatke i slobodne termine za sportski objekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima iznajmljivača, administrator mu je odobrio prijavu tog sportskog objekta
- **Opis osnovnog tijeka:**
  1. Iznajmljivač odabire objekt za koji želi pregledati podatke
  2. Prikazuju se podaci i slobodni termini za taj sportski objekt

**UC14 - Uređivanje sportskog objekta**

- **Glavni sudionik:** Iznajmljivač
- **Cilj:** Urediti podatke za sportski objekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima iznajmljivača te pregledava podatke i slobodne termine za odabrani sportski objekt.
- **Opis osnovnog tijeka:**
  1. Iznajmljivač odabire "Uredi"
  2. Iznajmljivač mijenja podatke ili slobodne termine za taj objekt

3. Iznajmljivač odabire opciju "Spremi"
4. Promjene se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
  - 2.a Iznajmljivač promjeni podatke, ali ne odabere opciju "Spremi"
    1. Sustav obavještava iznajmljivača da nije spremio podatke prije izlaska iz prozora

#### UC15 - Pregled rezervacija sportskog objekta

- **Glavni sudionik:** Iznajmljivač
- **Cilj:** Odobriti nove termine koje su korisnici zatražili i vidjeti već odobrene termine
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima iznajmljivača
- **Opis osnovnog tijeka:**
  1. Iznajmljivač odabire kalendar
  2. Iznajmljivač vidi datume s već odobrenim terminima te listu zahtjeva za iznajmljivanjem
  3. Za svaki zahtjev može odabrati "Potvrdi" ili "Odbaci"

#### UC16 - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati registrirane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju pregledavanja registriranih korisnika
  2. Sustav prikazuje listu registriranih korisnika

#### UC17 - Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima administratora i odabrao je opciju pregleda registriranih korisnika
- **Opis osnovnog tijeka:**
  1. Administrator iz liste bira korisnika kojeg želi izbrisati

2. Odabirom opcije "Obriši" briše se korisnik te sve njegove prijave na sportske aktivnosti, njegovi sportski objekti te svi treninzi koje je organizirao

#### **UC18 - Promjena prava pristupa**

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti prava pristupa registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima administratora i odabrao je opciju pregleda registriranih korisnika
- **Opis osnovnog tijeka:**
  1. Administrator iz liste bira korisnika kojemu želi promijeniti ovlasti
  2. Administrator bira koju razinu ovlasti daje odabranom korisniku

#### **UC19 - Pregled zahtjeva**

- **Glavni sudionik:** Administrator
- **Cilj:** Odobriti ili odbaciti zahtjeve za registraciju trenera i zahtjeve iznajmljivača za dodavanje sportskog objekta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s ovlastima administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju pregledavanja zahtjeva
  2. Administrator uz svaki zahtjev ima opciju dohvaćanja priložene dokumentacije
  3. Administrator nakon pregleda dokumentacije zahtjeva bira "Potvrdi" ili "Odbaci"

#### **UC20 - Organiziranje profesionalnog treninga**

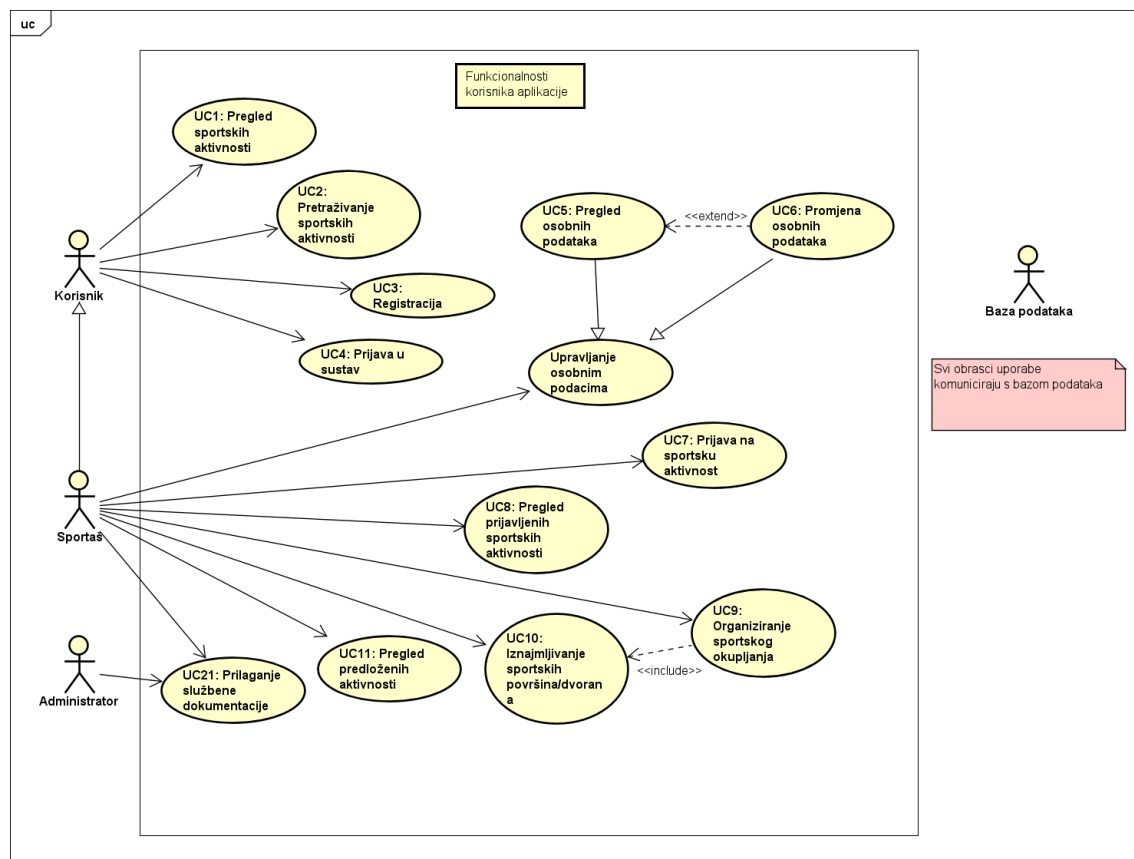
- **Glavni sudionik:** Trener
- **Cilj:** Organizirati profesionalni trening
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i odobrena mu je licenca trenera za određeni sport
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Organiziraj profesionalni trening"
  2. Korisnik upisuje tražene podatke

3. Korisnik potvrđuje upisane podatke
  4. Korisnika se preusmjerava na stranicu za pregled prijavljenih sportskih aktivnosti
- **Opis mogućih odstupanja:**
    - 2.a Upisano mjesto profesionalnog treninga nije javno i korisnik ga nije iznajmio
      1. Sustav obavještava korisnika da nije iznajmio sportsku površinu/dvoranu te mu daje opciju za iznajmljivanje
      2. Korisnik može promijeniti tražene podatke, odabrati opciju za iznajmljivanje ili odustati od organizacije

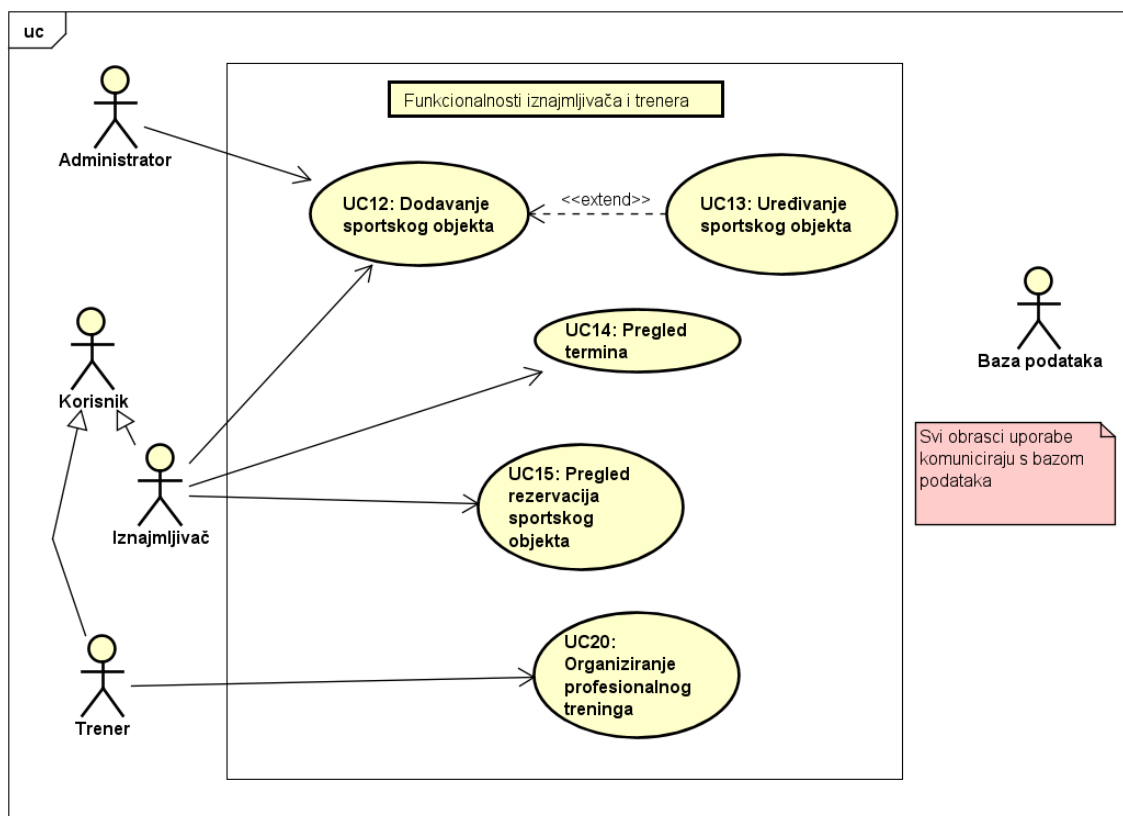
### **UC21 -Prilaganje službene dokumentacije**

- **Glavni sudionik:** Sportaš
- **Cilj:** Dobiti odgovarajuću ulogu i odgovarajuće ovlasti
- **Sudionici:** Baza podataka, Administrator
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Dodaj službenu dokumentaciju"
  2. Korisnik upisuje tražene podatke
  3. Korisnik potvrđuje upisane podatke
  4. Administrator zaprima prijavu
  5. Korisnika se preusmjerava na stranicu kojom se korisniku daje do znanja da je njegova prijava zaprimljena te da će biti obaviješten kada ona bude obrađena

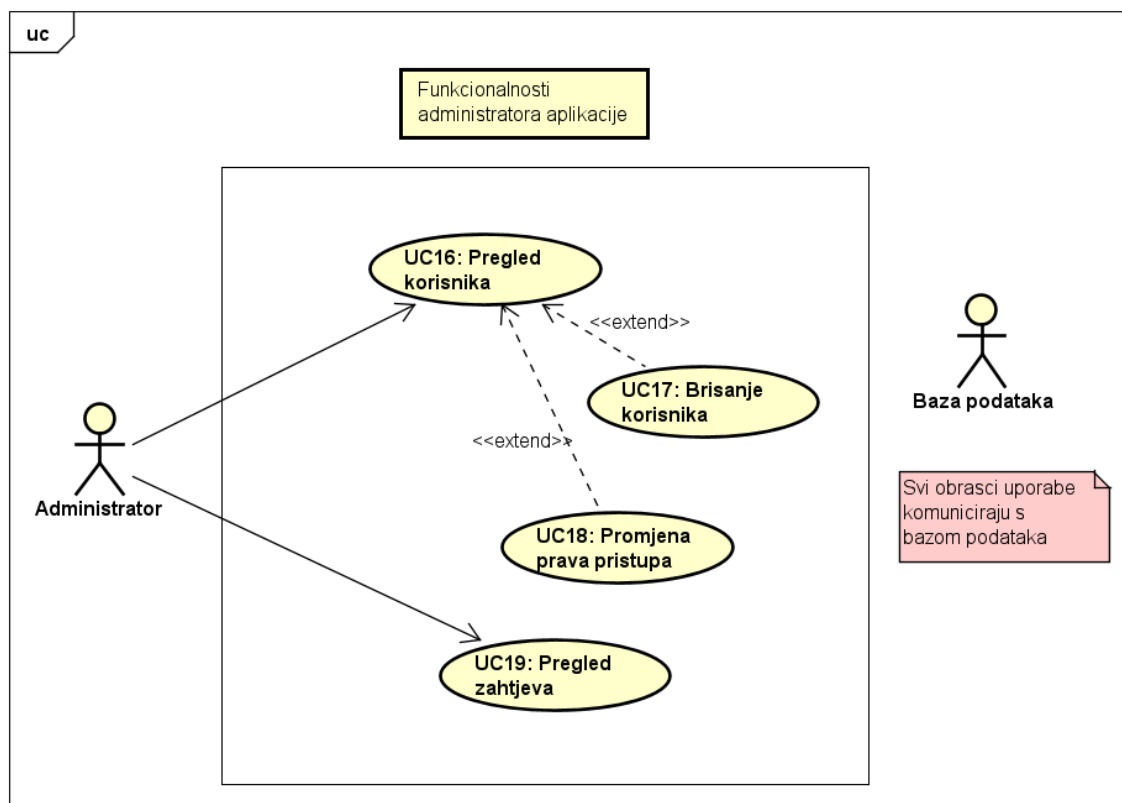
## Dijagrami obrazaca uporabe



Slika 3.1: Funkcionalnosti korisnika aplikacije



Slika 3.2: Funkcionalnosti iznajmljivača i trenera



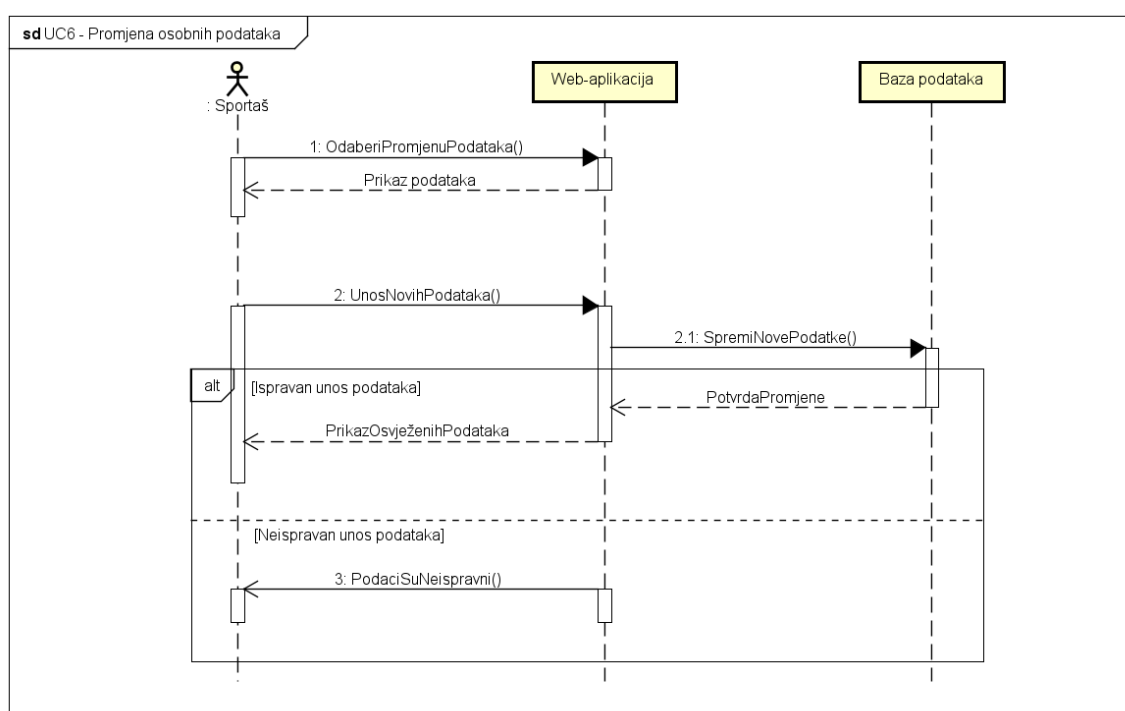
Slika 3.3: Funkcionalnosti administratora aplikacije



### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC6 - Promjena osobnih podataka

Prijavljeni korisnik šalje zahtjev za promjenu podataka. Poslužitelj zatim prikazuje korisnikove trenutne podatke i omogućuje mu izmjenu. Zatim korisnik unosi nove podatke te potvrđuje promjenu. Ako je korisnik ispravno unio podatke, poslužitelj sprema osvježene podatke u bazu podataka. Nakon što poslužitelj dobije potvrdu od baze da su podaci spremljeni, šalje korisniku obavijest da su novi podaci spremljeni u sustav. Ako je korisnik neispravno unio podatke, poslužitelj ga odmah obavještava o neispravnosti podataka.

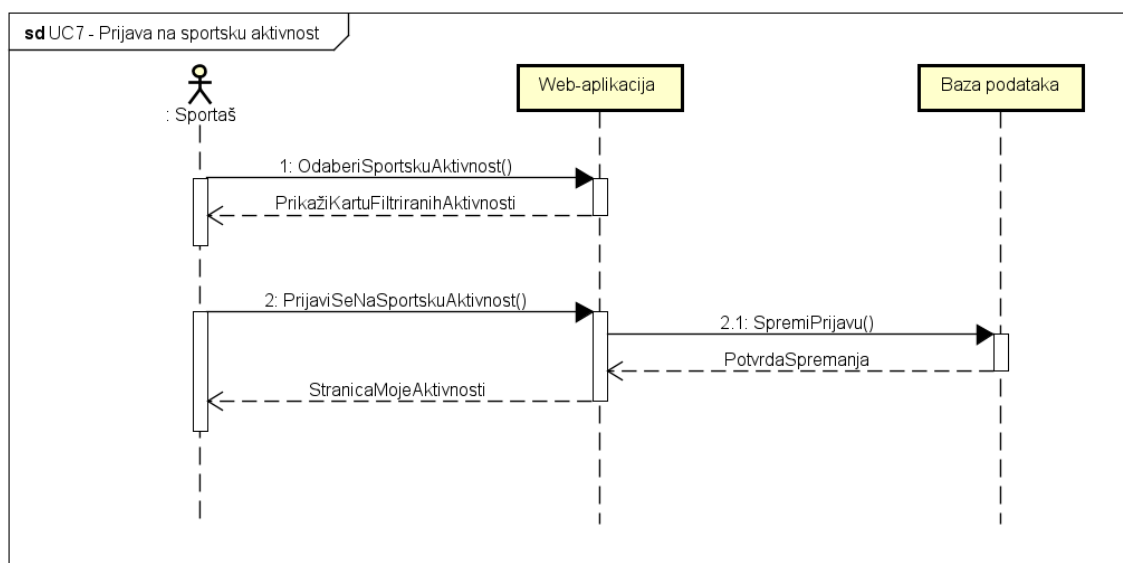


Slika 3.4: Sekvencijski dijagram za UC6

#### Obrazac uporabe UC7 - Prijava na sportsku aktivnost

Prijavljeni korisnik šalje zahtjev za odabir željene sportske aktivnosti. Poslužitelj zatim prikazuje kartu na kojoj se nalaze sve lokacije na kojima se održava sportska aktivnost koju je korisnik odabrao i koje imaju slobodnih mjesta. Korisnik zatim odabire željenu lokaciju i prijavljuje se za aktivnost. Poslužitelj zatim sprema prijavu u bazu podataka. Nakon što poslužitelj dobije potvrdu od baze da su podaci

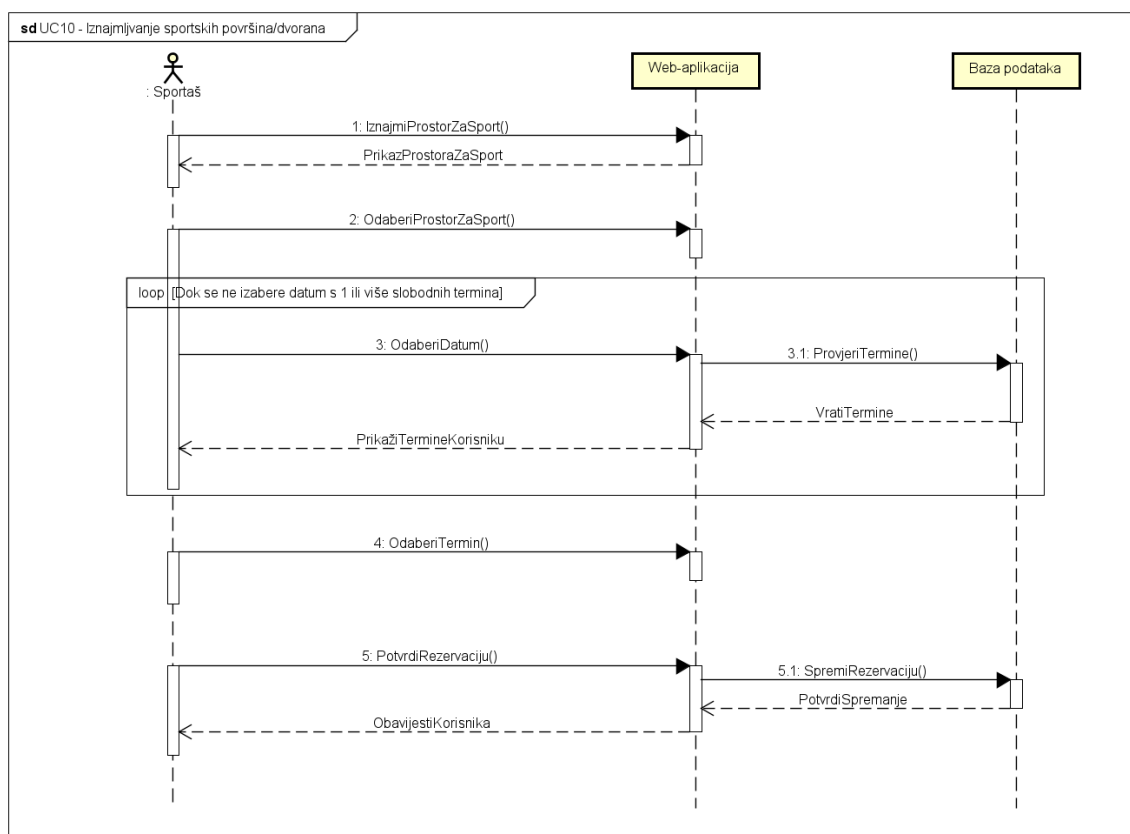
spremljeni, preusmjerava korisnika na stranicu gdje se nalaze sve prijavljene korisnikove aktivnosti.



Slika 3.5: Sekvencijski dijagram za UC7

### Obrazac uporabe UC10 - Iznajmljivanje sportskih površina/dvorana

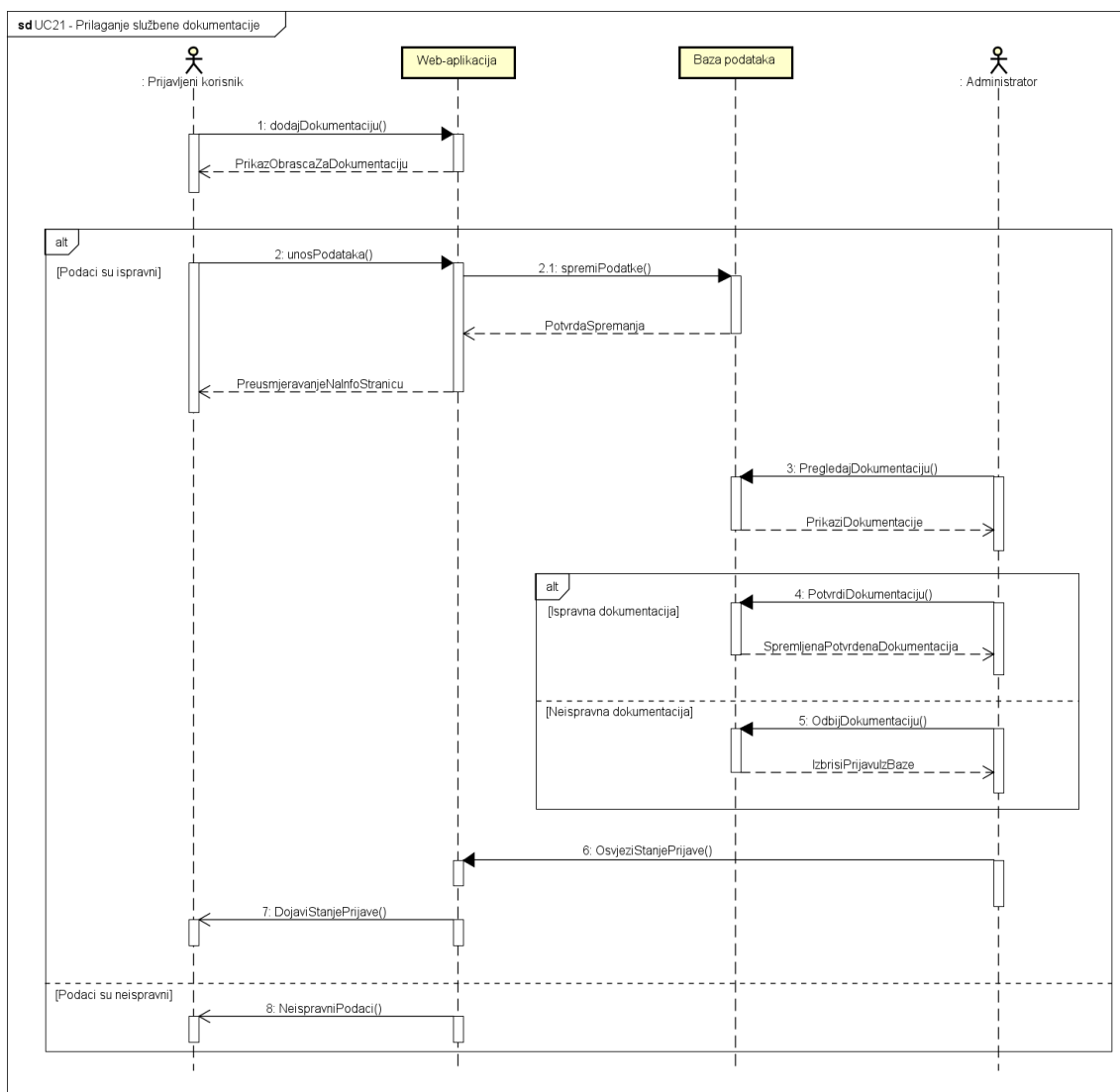
Prilikom organiziranja sportske aktivnosti prijavljeni korisnik šalje zahtjev za iznajmljivanje sportske površine/dvorane. Poslužitelj zatim prikazuje dostupne sportske površine/dvorane koje se mogu iznajmiti. Prijavljeni korisnik zatim odabire željenu sportsku površinu/dvoranu. Nakon toga prijavljeni korisnik odabire željeni datum. Poslužitelj zatim provjerava u bazi postoje li slobodni termini na datum koji je korisnik izabrao. Ako oni ne postoje, obavještava se korisnika da odabere neki drugi datum. Ako postoje slobodni termini na odabrani datum poslužitelj ih prikazuje prijavljenom korisniku. Korisnik zatim odabire željeni termin i nakon toga šalje zahtjev za potvrdu rezervacije. Poslužitelj sprema rezervaciju u bazu podataka i nakon što dobije potvrdu da je rezervacija spremljena o tome obavještava korisnika.



Slika 3.6: Sekvencijski dijagram za UC10

### Obrazac uporabe UC21 - Prilaganje službene dokumentacije

Prijavljeni korisnik šalje zahtjev za dodavanje službene dokumentacije. Poslužitelj prikazuje obrazac za prijavu dokumentacije. Prijavljeni korisnik zatim unosi svoje podatke, te ako su oni ispravni poslužitelj ih sprema u bazu podataka. Nakon što baza podataka spremi podatke, vraća potvrdu poslužitelju koji zatim preusmjerava prijavljenog korisnika na stranicu kojom se korisniku daje do znanja da je njegova prijava zaprimljena te da će biti obaviješten kada ona bude obrađena. Administrator zatim može zatražiti pregled svih dokumentacija iz baze podataka i pregledati nove prijave. Ako su svi podaci ispravni administrator potvrđuje dokumentaciju i potvrđenu dokumentaciju sprema u bazu podataka. Ako su podaci neispravni prijava se briše iz baze podataka. Zatim administrator osvježiti stanje prijave na poslužitelju, a nakon toga poslužitelj obaviještava korisnika o stanju njegove prijave. Ako su podaci bili neispravni kod samog upisa poslužitelj odmah javlja grešku korisniku.



Slika 3.7: Sekvencijski dijagram za UC21

## 3.2 Ostali zahtjevi

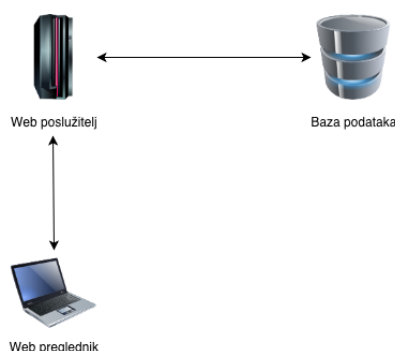
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Pristup bazi podataka mora biti zaštićen i ne smije trajati duže od nekoliko sekundi
- Sustav mora podržavati hrvatsku abecedu uključujući dijakritičke znakove
- Sustavu istovremeno mogu pristupiti više različitih vrsta korisnika
- Korisničko sučelje mora biti jednostavno, responzivno i intuitivno za korištenje te se mora ispravno prikazivati na stolnim računalima, laptopima i mobilnim uređajima
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnosti sustava
- Nadogradnja sustava ne smije narušiti funkcionalnosti sustava
- Sustav mora prijavljenim korisnicima jasno prikazati ovlasti koje im pripadaju (sportaš, trener, iznajmljivač)
- Sustav kao valutu za prikaz cijena i transakcija koristi HRK
- Pristup sustavu se mora odvijati pomoću protokola HTTPS

## 4. Arhitektura i dizajn sustava

Oblikovanje arhitekture i dizajna sustava jedno je od temeljnih i najvažnijih faza razvoja programske potpore. Uključuje glavna pravila i strukture sustava te se sve dalje izgrađuje na osnovi te arhitekture. Upravo zbog toga je ključan faktor u uspješnosti projekta, cijeni i težini izrade te održavanja.

Za našu aplikaciju odabrali smo objektno orijentiranu arhitekturu jer se pokazala kao najprikladnija za kompleksnu više-korisničku aplikaciju. Kako bi cilj omogućavanja što lakšeg i šireg korištenja naše aplikacije bio ispunjen, odabrali smo izradu web aplikacije kojoj će korisnici moći pristupiti s bilo kojeg uređaja s web preglednikom i spojenim na mrežu.

Na slici su vidljivi podsustavi klijent-poslužitelj arhitekture sustava. Komunikacija između web preglednika i web poslužitelja odvija se putem HTTP protokola.



Slika 4.1: Arhitektura sustava

Web preglednik je program putem kojega korisnik pristupa web-aplikaciji, šalje zahtjeve web poslužitelju i prikazuje vizualnu interpretaciju odgovora.

Web poslužitelj prima zahtjeve od korisnika i prosljeđuje ih web aplikaciji koja je pokrenuta na njemu, a odgovore na taj zahtjev, koje dobije od web aplikacije, prosljeđuje natrag web pregledniku korisnika.

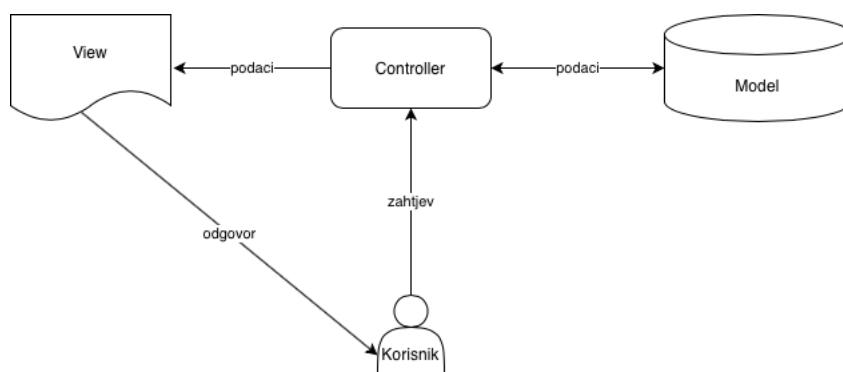
Web aplikacija na osnovu logike sustava, zahtjeva korisnika i podataka iz baze podataka generira odgovor na zahtjev korisnika.

Objektno orijentirana paradigma je način modeliranja sustava u kojem su osnova izgradnje objekti i modularnost, a osnovni elementi su:

- **Objekt** - primjerak razreda, odnosno smislene cjeline, sadrži vlastite atribute i metode
- **Nasljeđivanje** - mogućnost preuzimanja metoda od razreda-roditelja ili nadjačavanje istih vlastitim proširenjima
- **Polimorfizam** - moguće je postojanje metoda istog imena, a različitih atributa i funkcionalnosti

Oblikovanje navedene arhitekture pratit će MVC(Model-View-Controller) obrazac koji se sastoji od:

- **Model** - struktura za spremanje i upravljanje objektima i podacima
- **View** - komponenta za vizualni prikaz podataka korisniku
- **Controller** - sadrži logiku upravljanja korisničkim zahtjevima i na osnovu nje prilagođava podatke iz modela i prosljeđuje ili preusmjerava View-u za prikaz



Slika 4.2: Arhitektura sustava

Praćenje MVC obrasca omogućuje razdvajanje ovisnosti i odgovornosti što rezultira u olakšanju paralelnog razvoja više dijelova aplikacije, testiranju i naknadnom dodavanju funkcionalnosti u sustav.

Arhitektura sustava iz pogleda aktora sadrži sljedeće podsustave:

- Aplikacija za sportaše i trenere
- Aplikacija za iznajmljivače
- Aplikacija za administratora

Implementacijski je arhitektura podijeljena na backend i frontend. Frontend obuhvaća komponente vezane za vizualni prikaz kranjem korisniku i baziran je

na React-u<sup>1</sup>, open-source Javascript biblioteci. Backend sloj izravno prati MVC razvojni obrazac i za njega je korišten programski jezik Java, točnije Spring okvir, kojemu je jedna od glavnih značajki inverzija ovisnosti. Osim toga nudi i gotove implementacije mnogobrojnih protokola i automatsko namještanje detaljnih postavki sustava. Za sigurnost je korišten JWT (JSON Web Token) koji je popularni otvoreni standard tokena za potpisivanje i enkripciju prilikom slanja podataka u JSON formatu.

## 4.1 Baza podataka

Kako bi se korisniku omogućila brza i efikasna izmjena i pregled podataka web aplikacija je u interakciji s bazom podataka. Baza zadovoljava treću normalnu formu te se brine o konzistentnosti podataka, primarnim i stranim ključevima te integritetskim ograničenjima. Baza se brine o istodobnom pristupu podacima, dakle transakcije zadovoljavaju svojstva ACID(Atomicity, Consistency, Isolation, Durability). Za upravljanje bazom podataka koristimo sustav PostgreSQL zbog toga što je to besplatan sustav te je otvorenog koda i dostupan je za više operacijskih sustava.

### 4.1.1 Opis tablica

U nastavku navedeni su entiteti i atributi koje sadrži baza podataka te njihov tip podatka i opis. Atributi koji su u tablici označeni svijetlozelenom bojom predstavljaju primarni ključ, a atributi označeni svijetloplavom su strani ključevi.

t_user		
id	INT	Jedinstveni identifikator svakog korisnika registriranog u sustavu.
username	VARCHAR	Korisničko ime s kojime se korisnik prijavljuje u sustav.
passwd_hash	VARCHAR	Zaporka u obliku sažetka(hasha).

U relaciji t\_user nalaze se korisničko ime, zaporka te id koji služi kao jedinstveni identifikator svakog korisnika prijavljenog u sustav.

---

<sup>1</sup><https://reactjs.org/>



<b>t_administrator</b>		
id	INT	Identifikator Jedinstveni identifikator svakog korisnika registriranog u sustav.

Relacija t\_administrator sadrži identifikatore svih registriranih korisnika koji rade kao administratori sustava i imaju posebne ovlasti. Administrator odobrava i provjerava istinitost podataka i dokumenata koje prilažu iznajmljivači i treneri. Id je strani ključ koji se poziva na relaciju t\_users.

<b>t_person</b>		
id	INT	Jedinstveni identifikator svakog korisnika registriranog u sustavu.
email	VARCHAR	Email adresa sa kojom je povezan korisnički račun.
name	VARCHAR	Korisnikovo ime.
surname	VARCHAR	Korisnikovo prezime.

Relacija t\_person odnosi se na sve korisnike koji su registrirani kao sportaši, iznajmljivači ili treneri. Za njih bilježimo email ime i prezime. U relaciji također postoji id kao strani ključ koji se poziva na relaciju t\_user.

<b>t_athlete</b>		
id	INT	Jedinstveni identifikator svakog korisnika registriranog u sustavu.
birthday_date	DATE	Datum rođendna sportaša.
gender	VARCHAR	Spol sportaša.
height	DOUBLE	Sportaševa visina izražena u cm.
weight	DOUBLE	Sportaševa težina izražena u kg.

Relacija t\_athlete sadrži sve bitne podatke o korisnicima koji se u sustav prijavljuju kao sportaši. Sportaši su korisnici koji se mogu prijaviti ili organizirati sportska okupljanja odnosno događaje koji se ne naplaćuju i nisu pod profesionalnim vodstvom. U relaciji također postoji id kao strani ključ koji se poziva na relaciju t\_users.

<b>t_renter</b>		
id	INT	Identifikator Jedinstveni identifikator svakog korisnika registriranog u sustav.

Relacija t\_renter sadrži identifikatore svih registriranih korisnika koji su se u sustav prijavili kao iznajmljivači sportskih objekata. Oni u sustav mogu dodati svoj sportski objekt navodeći njegove specifikacije poput lokacije, tipa sportova koji se u tom objektu mogu igrati, cijene po satu i termine dostupne za rezervaciju. Id je strani ključ koji se poziva na relaciju t\_user.

<b>t_coach</b>		
id	INT	Identifikator Jedinstveni identifikator svakog korisnika registriranog u sustav.

Relacija t\_coach sadrži identifikatore svih trenera. Za razliku od sportaša treneri mogu organizirati i treninge za koje mogu postaviti određenu cijenu.

<b>t_location</b>		
id	INT	Jedinstveni identifikator svake lokacije dostupne u sustavu.
name	VARCHAR	Naziv lokacije.
location_type	VARCHAR	Javna površina ili plaćeni sportski objekt.
gpxcoordinates	VARCHAR	Geografske kordinate lokacije.
person_id	INT	Identifikator korisnika koji je sportsku lokaciju dodao u sustav.

U relaciji t\_location nalaze sve sve lokacije unesene u sustav. Ovisno o tipu lokacije razlikujemo plaćene i javno dostupne lokacije. Za lokacije bilježi se id kreatora lokacije odnosno korisnika koji je unjeo u sustav lokaciju i potrebne podatke. Plaćene lokacije dopušteno je kreirati samo korisnicima kojima je odobren status iznajmljivača. Za plaćene lokacije evidentira se id iznajmljivača koji posjeduju taj objekt. Također u relaciji se spremaju kordinate lokacije koje mogu poslužiti kao marker na karti u slučaju jedne kordinate ili za iscrtavanje cijelog poligona u slučaju više zabilježenih koordinata.

<b>t_opening</b>		
id	INT	Jedinstveni identifikator radnog vremena.
cost	DOUBLE	Cijena u slučaju plaćene lokacije.
start_time	TIME	Vrijeme otvaranja.
end_time	TIME	Vrijeme zatvaranja.
from_date	TIMESTAMP	Datum početka rasporeda.
to_date	TIMESTAMP	Datum kraja rasporeda.
weekday	INT	Dan u tjednu na kojega se odnosi radno vrijeme.

Relacija t\_opening evidentiraju se radna vremena koja možemo pridružiti nekim lokacijama. Za radno vrijeme evidentiraju se početak i kraj te u kojemu danu te datumi odkada dokada se takav raspored proteže.

<b>t_location_openings</b>		
paid_location-id	INT	Identifikator lokacije za iznajmljivanje.
openings_id	INT	Identifikator radnog vremena iz relacije t_opening.

Relacija t\_location\_open\_hours služi kao veza između relacija t\_opening i t\_location. U njoj su evidentirani radna vremena za sve plaćene lokacije.

<b>t_sport</b>		
id	INT	Jedinstveni identifikator pojedinog sporta.
name	VARCHAR	Naziv sporta.
iconbwuri	VARCHAR	URI crno bijele ikone.
icon_color_uri	VARCHAR	URI ikone s bojom.
type	VARCHAR	Tip sporta (indoor/outdoor).

U relaciji t\_sport nalazi se popis svih sportova evidentiranih u sustavu. Svaki sport ima id, ime i tip u ovisnosti o tome igra li se sport u zatvorenom prostoru ili ne.

<b>t_sportevent</b>		
id	INT	Jedinstveni identifikator svakog sportskog eventa.
location_id	INT	Jedinstveni identifikator lokacije na kojoj se održava sportski event.
organizer_id	INT	Jedinstveni identifikator organizatora sportskog eventa.
sport_id	INT	Jedinstveni identifikator sporta koji se igra na event.
sportevent_type	VARCHAR	Plaćeni ili neplaćeni event (amateur/professional).
max_number_of_participants	INT	Broj koliko ljudi maksimalno može sudjelovati na sportskom event.
start_date_time	TIMESTAMP	Datum i sat početka eventa.
end_date_time	TIMESTAMP	Datum i sat kraja eventa.
cost	DOUBLE	Cijena prisustvovanja u slučaju plaćenih eventova.

U relaciji t\_sportevent vidi se popis svih organiziranih sportskih eventova. Razlikujemo amaterske i profesionalne eventove po tome plaćaju li se ili ne odnosno tko ih je organizirao trener ili sportaš. Za svaki event unesen u sustav poznat nam je datum i vrijeme početka i kraja eventa, maksimalan broj sudionika, cijena, koji sport se može igrati, tko je organizirao event i naravno na kojoj lokaciji će se event održavati.

<b>t_sportevent_participants</b>		
events_participant_id	INT	Identifikator sportskog eventa u kojemu korisnik prisustvuje.
participants_id	INT	Identifikator korisnika koji prisustvuje u event.

U relaciji t\_sportevent\_participants evidentiramo id svakog korisnika koji prisustvuje u nekom eventu te id eventa u kojem prisustvuje.

<b>t_documentation</b>		
id	INT	Jedinstveni identifikator dokumenta.

<b>t_documentation</b>		
document-intern_uri	VARCHAR	Uri dokumenta.
type	INT	Vrsta dokumenta.
approved_by_id	INT	Identifikator administratora koji je potvrdio dokument.

U relaciji t\_documentation nalazi se popis svih dokumenata koje prilažu sportaši koji žele postati treneri i iznajmljivači da bi im se priznao status iznajmljivača. Također evidentira se koji administrator potvrđuje ili odbacuje dokumentaciju.

<b>t_coach_coach_certificate</b>		
coach_id	INT	Jedinstveni identifikator trenera.
coach-certificate_id	INT	Id dokumenta (certifikata trenera).

Relacija t\_coach\_coach\_certificate sadrži parove identifikatora trenera i identifikatora dokumenta koje prilažu da bi potvrdili svoj status trenera.

<b>t_location_ownership_certificate</b>		
paid_location-id	INT	Jedinstveni identifikator lokacije.
ownership-certificate_id	INT	Identifikator dokumenta.

Relacija t\_location\_ownership\_certificate sadrži parove identifikatora iznajmljivača i identifikatora dokumenta kojeg prilažu da bi dokazali da posjeduju sportski objekt kojeg žele iznajmljivati.

<b>t_documentation_sports</b>		
documentation-id	INT	Jedinstveni identifikator dokumenta.
sports_id	INT	Identifikator sporta.

Relacija t\_documentation\_sports sadrži parove identifikatora dokumenta koje trener prilaže da mu se potvrdi status trenera i identifikatora sporta na koji se taj

dokument odnosi.

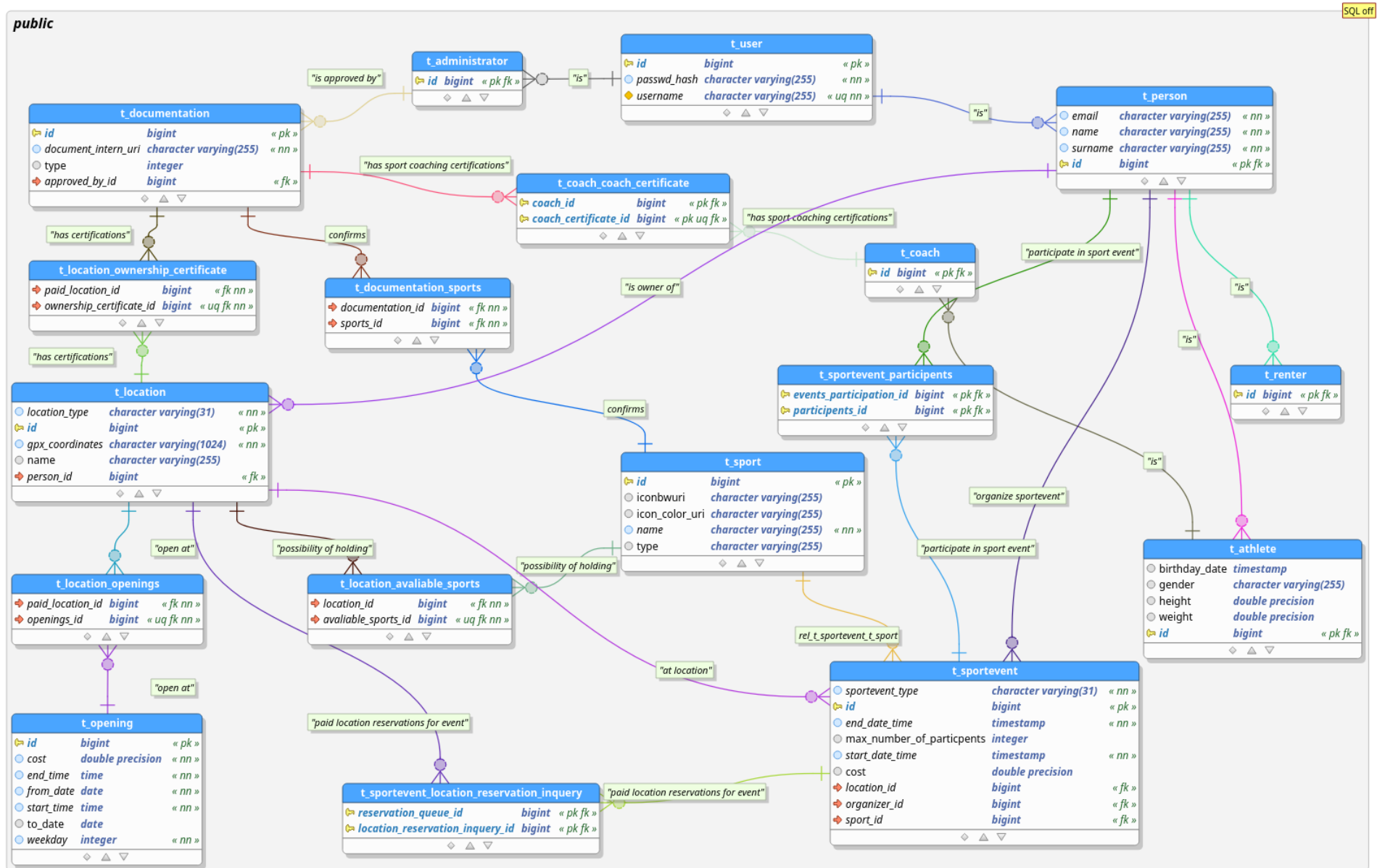
t_location_available_sports		
location_id	INT	Jedinstveni identifikator lokacije.
available-sports_id	INT	Identifikator sporta.

Relacija t\_location\_available\_sports sadrži parove identifikatora lokacije i sportova koji se mogu igrati na toj lokaciji.

t_sportevent_location_reservation_inquiry		
reservation-queue_id	INT	Jedinstveni identifikator liste za čekanje.
location-reservation-inquiry_id	INT	Identifikator rezervacije.

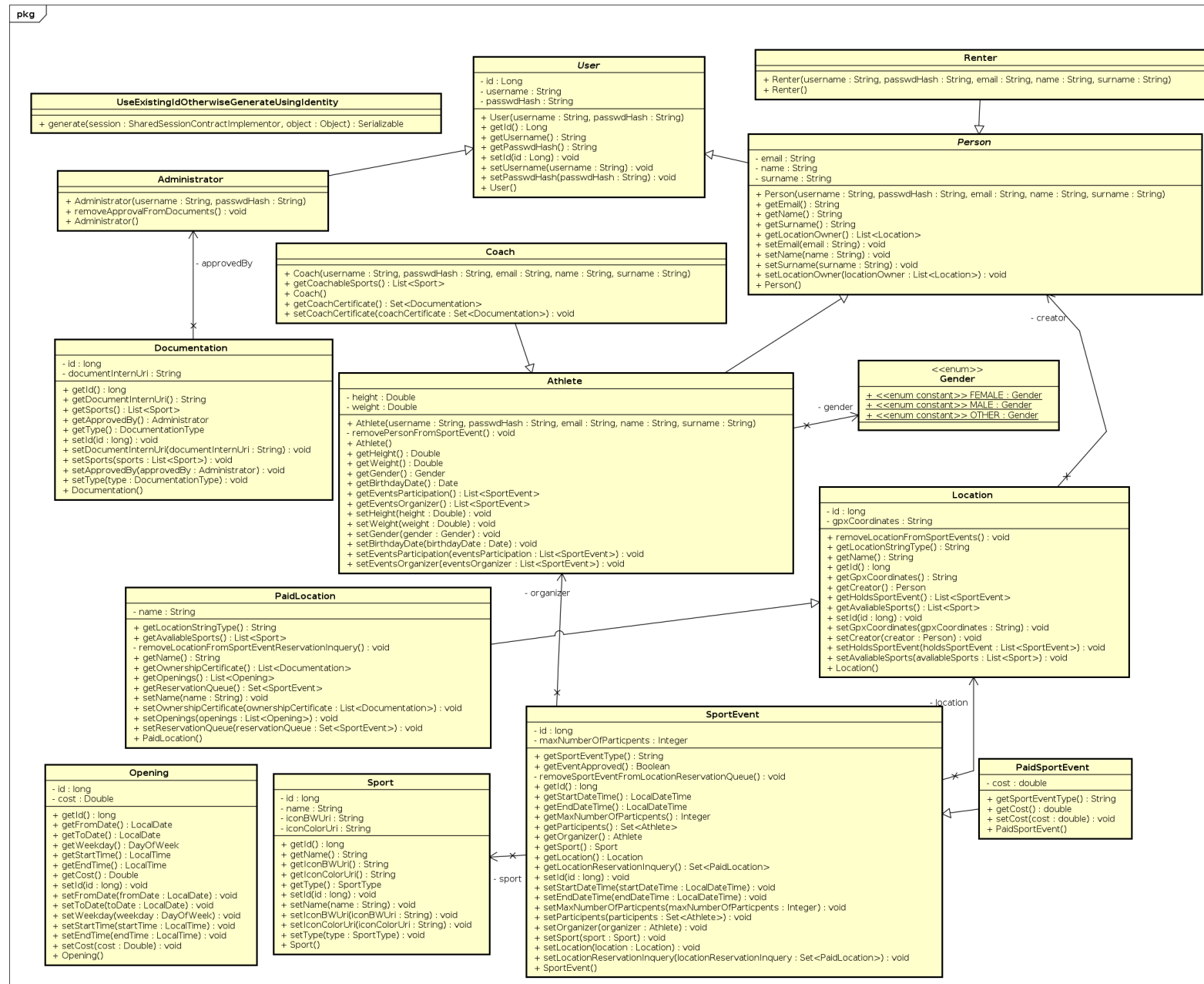
Relacija t\_sportevent\_location\_reservation\_inquiry Sadrži podatke o rezervaciji lokacije za sportska događanja.

## 4.2 Dijagram baze podataka



Slika 4.3: Dijagram baze podataka

## 4.3 Dijagram razreda



Slika 4.4: Dijagram razreda



Razred **User** predstavlja registriranog korisnika koji sadrži atribut `username` i `password`. Različite vrste korisnika reprezentiramo razredima **Administrator**, **Person**, **Athlete**, **Coach**, **Renter**.

Razred **Person** nasljeđuje **User** i predstavlja fizičku osobu te sadrži osobne podatke. **Athlete** nasljeđuje razred **Person** i predstavlja sportaša, dok **Coach** nasljeđuje razred **Athlete**.

Razred **Coach** sadrži i referencu na svoju dokumentaciju koja potvrđuje da je službeni trener za određene sportove.

Razred **Renter** predstavlja iznajmljivača i nasljeđuje razred **Person**. Razred **Administrator** nasljeđuje razred **User** te sadrži referencu na sve dokumente (**Documentation**) koje je potvrdio.

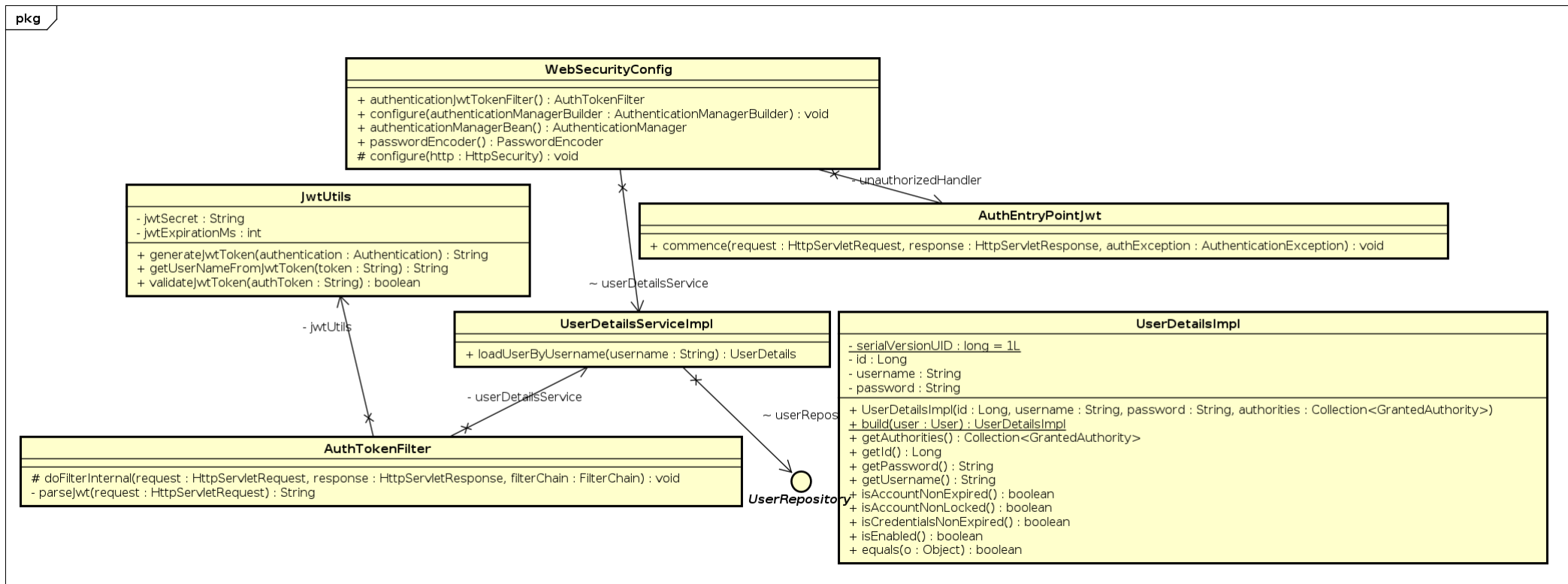
Razred **Location** predstavlja lokaciju nekog sportskog okupljanja kojeg je kreirao sportaš, te sadrži atribut `GPXCoordinates` u kojem su zapisane GPX koordinate objekta. Lokacija sadrži referencu na osobu (**Person**) koja ju je napravila.

Razred **PaidLocation** predstavlja lokaciju koju je iznajmljuje iznajmljivač (**Renter**). Plaćena lokacija sadrži popis radnih sati (**Opening**) i popis sportova (**Sport**) koji se mogu igrati na navedenoj lokaciji. Također lokacija koja se iznajmljuje mora imati dokumentaciju o objektu (**Documentation**) koju potvrđuje **Administrator**.

Razred **Opening** predstavlja jedan segment radnog vremena. Segment je definiran danom u tjednu (**weekday**) i početnim (**start\_time**) i završnim satom (**end\_time**). Takav segment se ponavlja sve dok je unutar područja aktivnosti definiranog s početnim datumom (**from\_date**) i završnim datumom (**to\_date**).

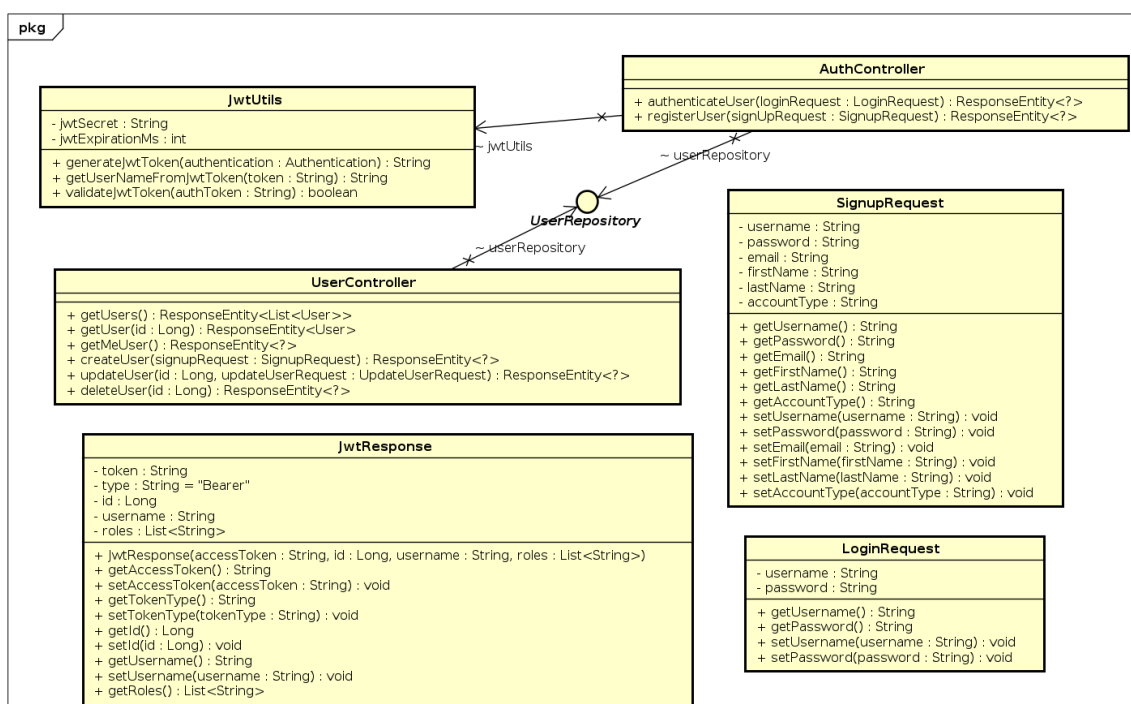
Razred **SportEvent** predstavlja događaj koje ovisno o tipu može biti sportsko okupljanje ili profesionalni trening. **SportEvent** sadrži referencu na lokaciju događaja (**Location**) i na osobu koja je napravila događaj (**Athlete**). Ako događaj iznajmljuje objekt (**PaidLocation**) tada iznajmljivač putem atributa `location_approved` potvrđuje rezervaciju objekta.

Razred **Documentation** predstavlja dokumentaciju te sadrži interni url PDF datoteke koja je prenesena. Također sadrži i referencu na sportove koji se odnose na dokumentaciju. Dokumentaciju potvrđuje administrator. Razred **Sport** sadrži informacije vezane uz jedan sport.



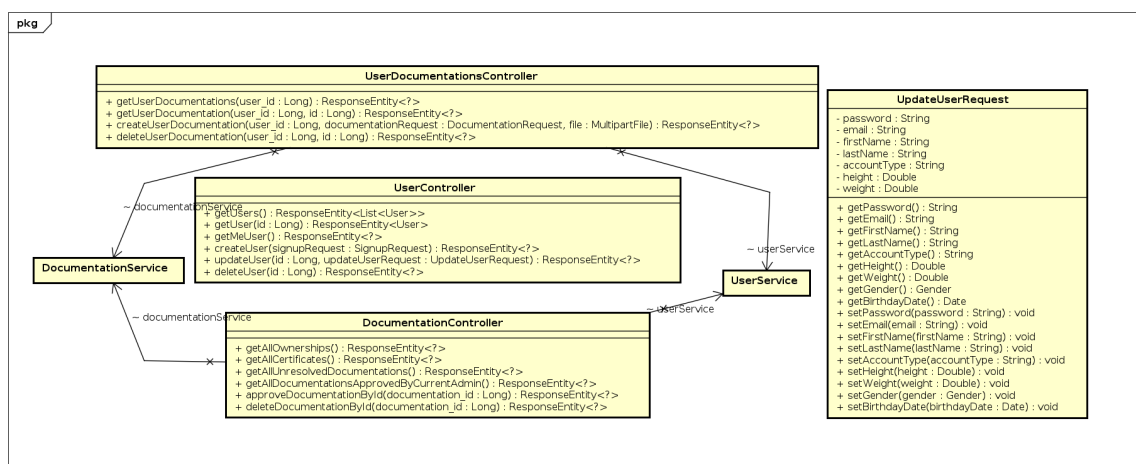
Slika 4.5: Dijagram razreda JWT autentifikacije

U razredu **WebSecurityConfig** koji implementira **WebSecurityConfigurerAdapter** sučelje postavljamo opcije vezane uz Spring Security. Razred **AuthEntryPointJwt** implementira **AuthenticationEntryPoint** te se metoda **commence** pokreće svaki put kada neregistrirani korisnik pristupi zaštićenom sadržaju, te se vraća HTTP 401 UNAUTHORIZED. Kako bi Spring Security omogućili da učita korisnika implementiramo sučelje **UserDetailsService** u razredu **UserDetailsService-Impl**, a podatke samog korisnika spremamo u razredu **UserDetailsImpl**. Razred **JwtUtils** pruža metode za parsiranje, generiranje, i validaciju JWT.



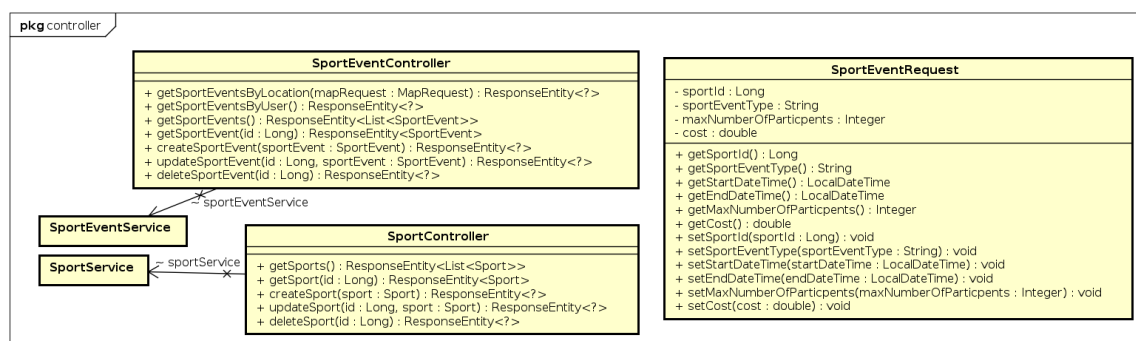
Slika 4.6: Dijagram razreda prijave u sustav

Razred **AuthController** posluhuje zahtjeve za prijavu i registraciju koje prima pomoću razreda **LoginRequest** i **SignupRequest**. Dok razred **ProfileController** posluhuje osnovne informacije o prijavljenom korisniku.



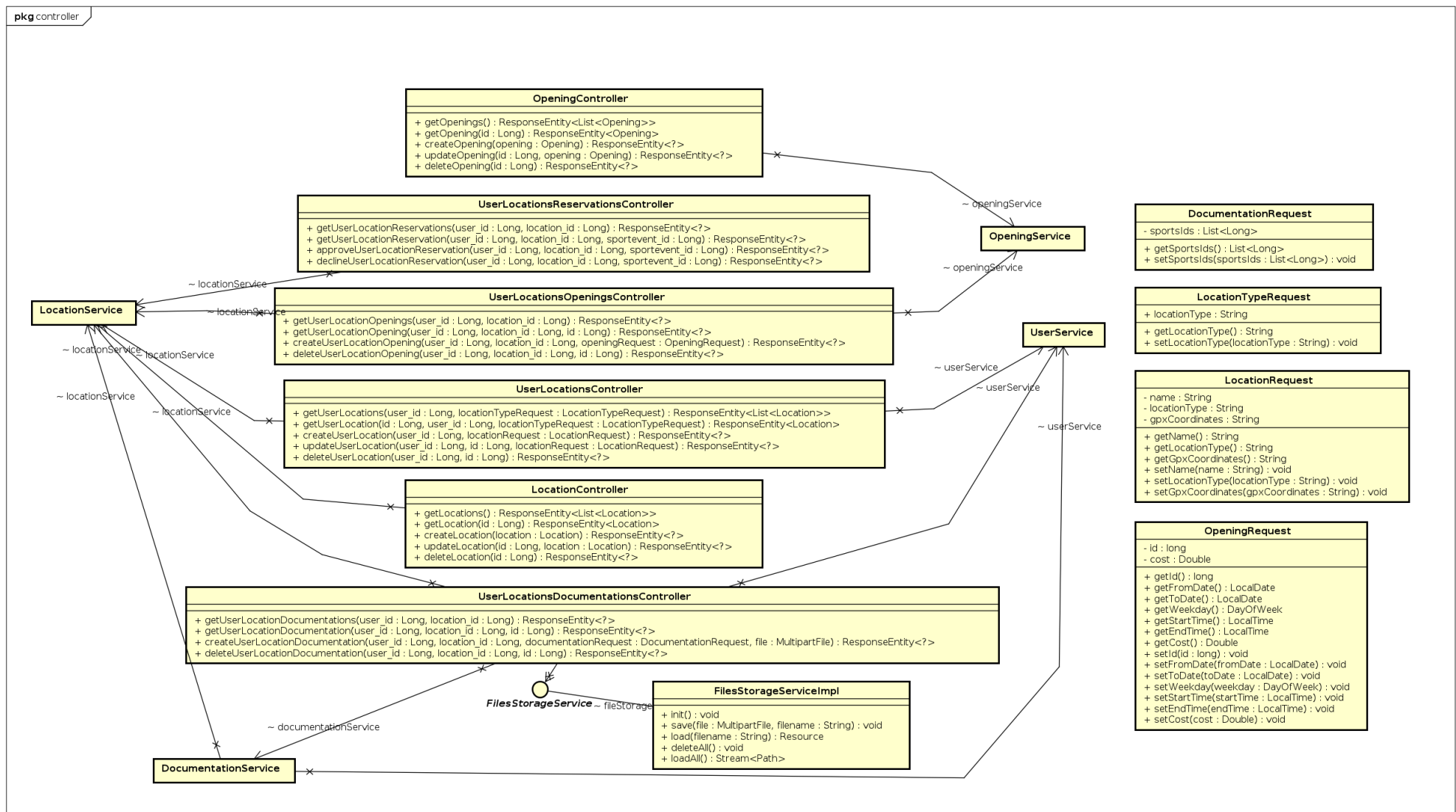
Slika 4.7: Dijagram razreda osnovnih korisničkih funkcionalnosti

Razred **UserController** omogućuje pregled osnovnih informacija o korisniku. Također pomoću podataka koje prima preko razreda **UpdateUserRequest** može vršiti osvježavanje podataka korisnika. Razred **UserDocumentationController** poslužuje zahtjeve za predaju dokumentacija trenera. Razred **DocumentationController** služi za pregled dokumentacija trenera i lokacija od strane administratora te potvrđivanje ili uklanjanje.

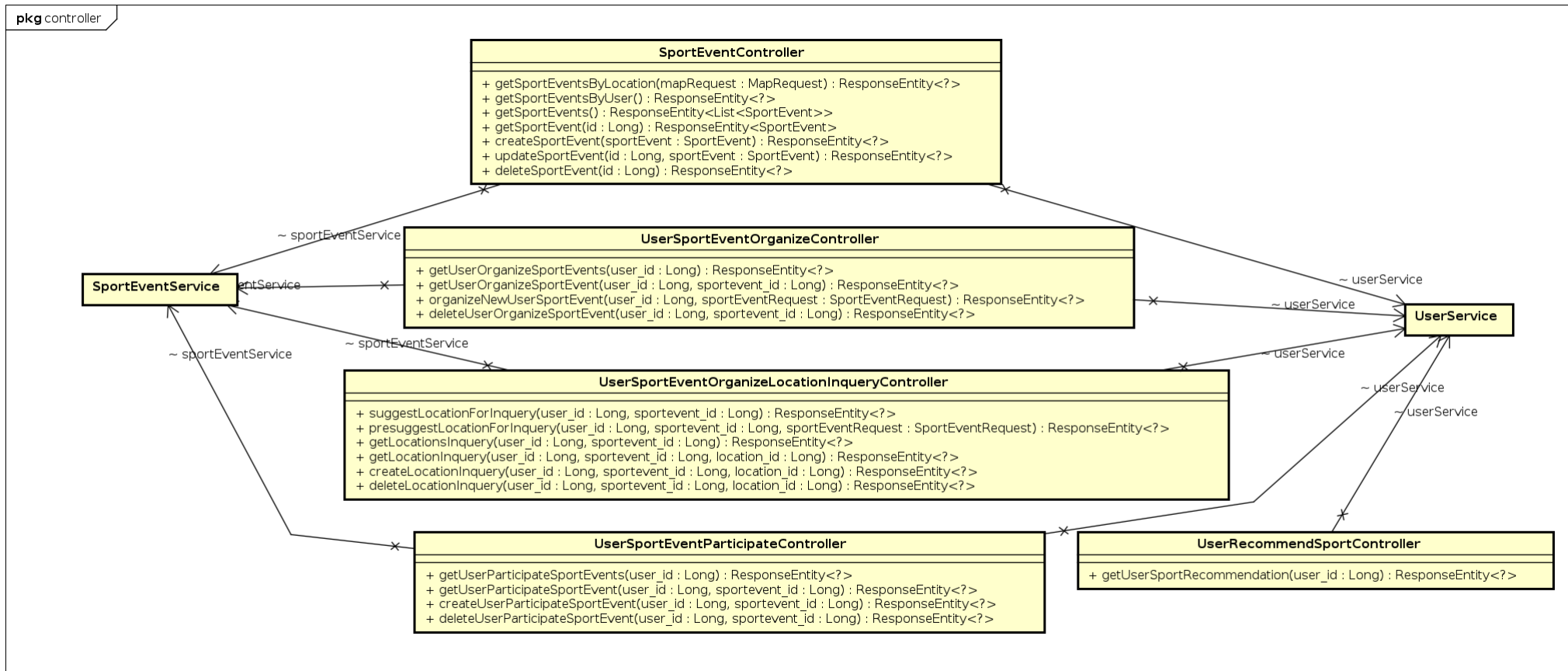


Slika 4.8: Dijagram razreda funkcionalnosti sportskih okupljanja

Razred **SportEventController** poslužuje osnovne informacije o sportskim okupljanjima. Također omogućuje dobivanje podataka o sportskim okupljanjima na nekoj lokaciji. Razred **SportController** služi za dohvaćanje podataka o sportovima koji su dostupni unutar aplikacije.



Slika 4.9: Dijagram razreda funkcionalnost lokacija



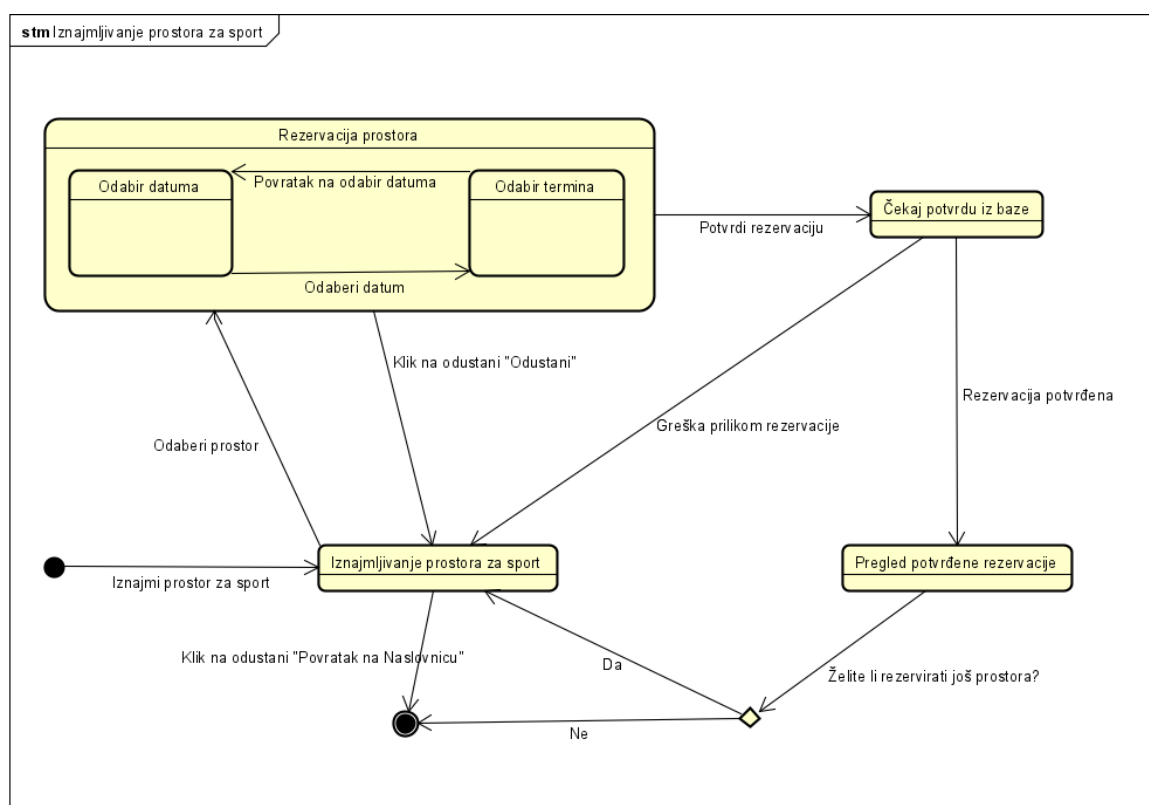
Slika 4.10: Dijagram razreda veze lokacija i sportskih okupljanja

Razred **OpeningController** poslužuje informacije o radnom vremenu lokacije. Razred **LocationController** omogućuje pregled lokacija. Razred **UserLocationController** služi za upravljanje korisnikovih lokacija. **UserLocationsOpenings** omogućuje postavljanje radnog vremena korisnikovoj lokaciji koja se plaća. Razred **UserLocationsReservationsController** omogućuje potvrđivanje rezervacije nekog sportskog okupljanja na pojedinoj lokaciji. Razred **UserLocationsDocumentationController** omogućuje iznajmljivačima prijenos dokumentacija na poslužitelj.

Razred **UserSportEventOrganizeController** omogućuje sportašu slanje zahtjeva za organiziranje sportskih okupljanja. Svako sportsko okupljanje može zatražiti rezervaciju plaćene lokacije preko razreda **UserSportEventOrganizeLocationInquiryController**. Sportaš preko razreda **UserSportEventParticipateController** se može prijaviti na sportsko okupljanje.

## 4.4 Dijagram stanja

Na slici 4.7 je prikazan dijagram stanja za iznajmljivanje prostora za sport. Korisnik s početne stranice "Naslovnica" klikom na "Iznajmi prostor za sport" dolazi na stranicu za iznajmljivanje prostora za sport. Na toj stranici korisnik može pregledati dostupne prostore za sport, nakon odabira prostora, korisnik odabire dostupan datum i vrijeme termina. Prije klika na "Potvrdi rezervaciju" korisnik može odustati od rezervacije klikom na "Odustani" i vratiti se nazad na stranicu za pregled dostupnih prostora za sport. Nakon odabira prostora, datuma i termina korisnik završava rezervaciju klikom na "Potvrdi rezervaciju" te čeka potvrdu rezervacije. Nakon što sustav potvrdi rezervaciju korisnik bira hoće li napraviti još rezervacija ili ne, ako odabere da želi napraviti još rezervacija sustav ga vrati na stranicu za iznajmljivanje prostora za sport, inače ga vrati na "Naslovnicu". Ako dođe do greške prilikom rezervacije sustav obavijesti korisnika o grešci te ga vrati na početnu stranicu za iznajmljivanje. Korisnik se može s početne stranice za iznajmljivanje vratiti na "Naslovnicu" klikom na "Povratak na Naslovnicu".

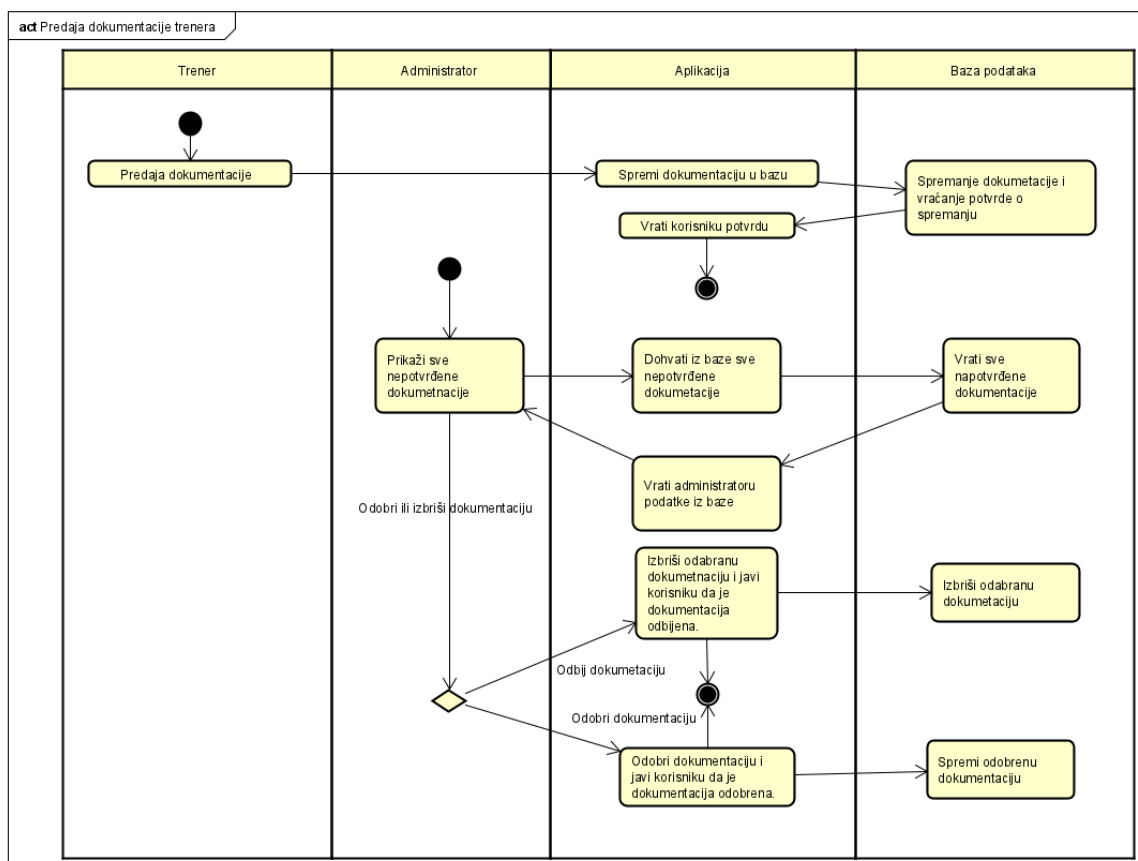


Slika 4.11: Dijagram stanja iznajmljivanja prostora za sport



## 4.5 Dijagram aktivnosti

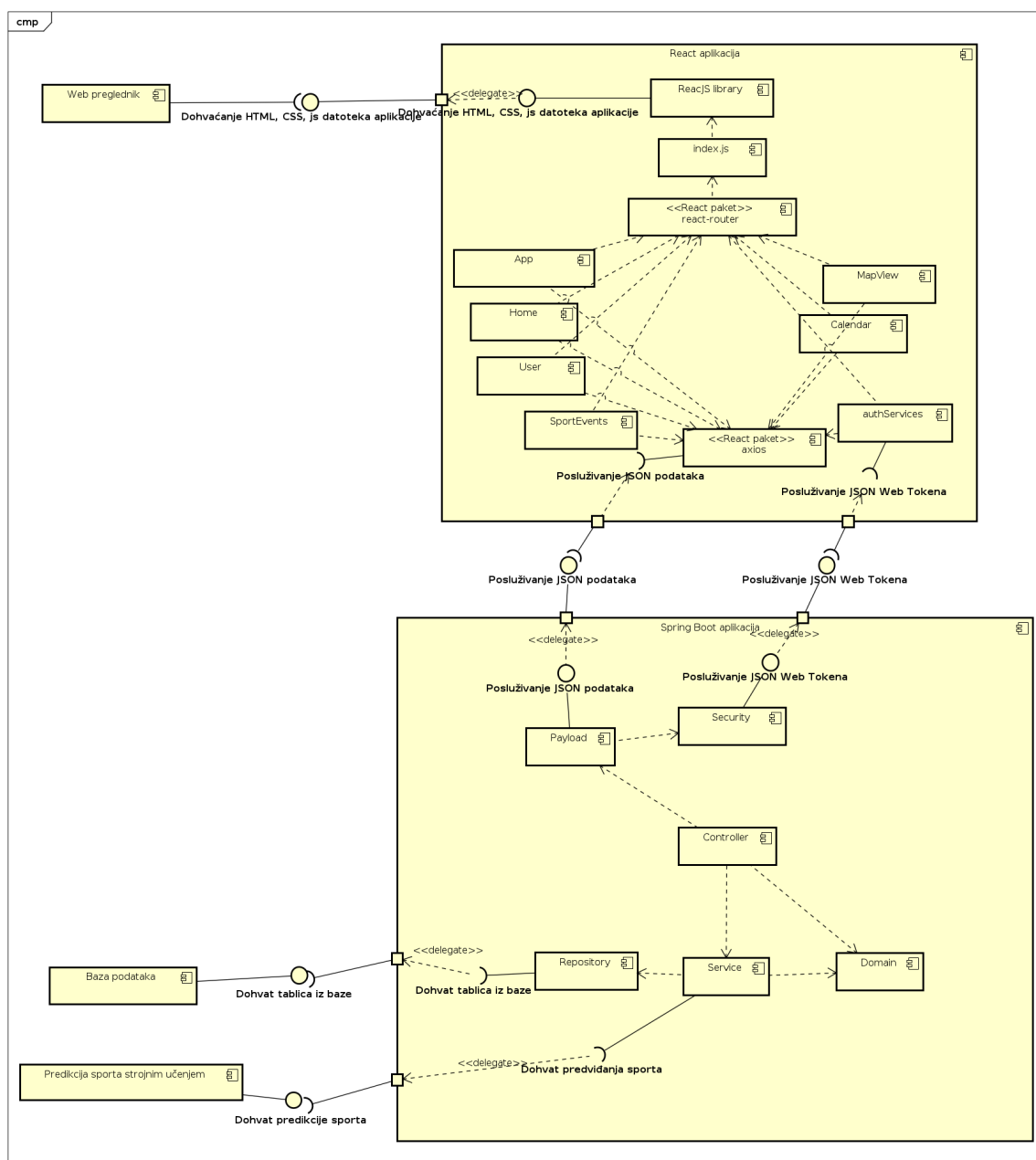
Na slici 4.8 prikazan je dijagram aktivnosti predaje dokumentacije trenera i prikaz aktivnosti potvrđivanja (ili odbijanja) predane dokumentacije. Trener preda dokumentaciju aplikaciji te aplikacija tu dokumentaciju spremi u bazu. Prilikom spremanja dokumentacije u bazu se zapisuje i podatak da je ta dokumentacija nepotvrđena. Administrator može zatražiti od aplikacije popis svih nepotvrđenih dokumentacija, aplikacija ih uzme iz baze te prikaže administratoru. Administrator pregledava dokumentacije te odlučuje hoće li ih odbaciti (izbrisati) ili potvrditi. Nakon što administrator donese odluku o dokumentaciji trener dobiva obavijest o odluci administratora.



Slika 4.12: Dijagram aktivnosti predaje dokumentacije za trenera

## 4.6 Dijagram komponenti

Dijagram komponenti prikazan na slici opisuje organizaciju i međuovisnost komponenti aplikacije. Aplikaciji se pristupa s web preglednika kojem React biblioteka poslužuje HTML, CSS i Javascript datoteke generirane iz Javascript datoteka predanih biblioteci. React aplikacija odnosno "frontend" dio web aplikacije s "backend" dijelom komunicira slanjem podataka u JSON formatu pomoću programskog paketa Axios. Za osjetljive podatke poput korisničkih informacija podaci se šalju u obliku JWT, odnosno JSON Web Token. "Backend" dio web aplikacije s bazom podataka komunicira preko komponente Repository, a podaci dobiveni iz baze podataka koriste se dalje u "backend" dijelu web aplikacije. "Backend" dio web aplikacije također komunicira s predikcijom sporta strojnim učenjem.



Slika 4.13: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

Komunikacija članova tima realizirana je aplikacijom Discord<sup>1</sup>. Dokumentacija je pisana u alatu TEXstudio<sup>2</sup>, a UML dijagrami crtani su alatom Astah<sup>3</sup>. Sustav za upravljanje izvornim kodom je Git<sup>4</sup>, a udaljeni repozitorij projekta dostupan je na web platformi GitLab<sup>5</sup>. Kao razvojno okruženje za frontend koristio se Visual Studio Code<sup>6</sup> dok su se za backend koristili su se IntelliJ IDEA<sup>7</sup> i Eclipse<sup>8</sup>. Za izradu aplikacije korištene su tehnologije Spring Boot<sup>9</sup> za backend te React<sup>10</sup> za frontend. Tehnologija Spring Boot zahtjeva programiranje u Javi<sup>11</sup>, a tehnologija React programiranja u JavaScriptu<sup>12</sup>. Sustav baze podataka ove aplikacije je PostgreSQL<sup>13</sup>. Još treba napisati u čemu smo testirali aplikaciju, uključujući korištene biblioteke.

---

<sup>1</sup><https://discord.com/>

<sup>2</sup><https://www.texstudio.org/>

<sup>3</sup><https://astah.net/>

<sup>4</sup><https://git-scm.com/>

<sup>5</sup><https://gitlab.com/>

<sup>6</sup><https://code.visualstudio.com/>

<sup>7</sup><https://www.jetbrains.com/idea/>

<sup>8</sup><https://www.eclipse.org/>

<sup>9</sup><https://spring.io/projects/spring-boot>

<sup>10</sup><https://reactjs.org/>

<sup>11</sup><https://www.java.com/en/>

<sup>12</sup><https://www.javascript.com/>

<sup>13</sup><https://www.postgresql.org/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Kako bi se uvjerali u ispravnost pojedinih klasa i metoda proveli smo ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Testove smo napisali tako da ispituju redovne slučajeve, ali i rubne slučajeve. Koristili smo javin open source framework za testiranje JUnit.

#### Ispitni slučaj 1: traženje korisnika po korisničkom imenu

Sljedeća dva testa provjeravaju funkcionalnosti klase userService. Klasa userService pruža metodu za dohvat korisnika na temelju njegovog korisničkog imena. Tu metodu smo testirali sa validnim korisničkim imenom "athlete" u metodi testFindGoodUsername te sa imenom nepostojećeg korisnika u metodi testFindBadUsername.

```
1 @Test
2 void testFindGoodUsername() {
3     Optional < User > user = userService.findByUsername("athlete");
4     assertThat(user.isPresent());
5 }
6
7 @Test
8 void testFindBadUsername() {
9     Optional < User > user = userService.findByUsername("wrongUsername");
10    assertThat(user.isEmpty());
11 }
```

Izvorni kod 5.1: findByUsername

Rezultat izvođenja ispitnog slučaja: Ispit je uspješan.

#### Ispitni slučaj 2: Dodavanje i spremanje korisnika.

Slijedećim testom provjerili smo funkcionalnosti stvaranja i dodavanja novog korisnika sportaša. Testirane metode također su dio klase userService.

```
1 @Test
2 void testSaveAndFindUser() {
3     User user = new Athlete("newUsername", "newPassword", "e@mail.com",
```

```
        "name", "surname");
4    userService.save(user);
5    Optional < User > user1 = userService.findByUsername("newUsername");
6    assertThat(user1.isPresent());
7 }
```

Izvorni kod 5.2: testSaveAndFindUser

Izvođenje ispitanog slučaja: Ispit je uspješan.

### Ispitni slučaj 3: Testiranje kontrolera za sportske događaje.

Naredna dva testa provjerava kontroler sportskih događaja, odnosno klasu `sportEventController`. U testu `testSporteventControllerFindByGoodId()` kreiramo http get request na `/api/v1/sportevents/2` da bi provjerili dostupnost postojećeg sportskog eventa, sportskog eventa pod brojem 2. Analogno u metodi `testSporteventControllerFindByBadId()` pokušavamo dohvatiti nepostojeći sportski event, event koji ima id 2000.

```
1 @Test
2 void testSporteventControllerFindByGoodId() throws Exception {
3     this.mockMvc.perform(get("/api/v1/sportevents/2"))
4         .andDo(print()).andExpect(status().isOk());
5 }
6
7 @Test
8 void testSporteventControllerFindByBadId() throws Exception {
9     this.mockMvc.perform(get("/api/v1/sportevents/2000"))
10        .andDo(print()).andExpect(status().isBadRequest());
11 }
```

Izvorni kod 5.3: testSporteventControllerFindById

Izvođenje ispitanog slučaja: Ispit je uspješan.

### Ispitni slučaj 4: Testiranje kontrolera za sportske lokacije.

Na sličan način testirali smo i funkcionalnosti kontrolera za sportske lokacije odnosno klasu `locationController`.

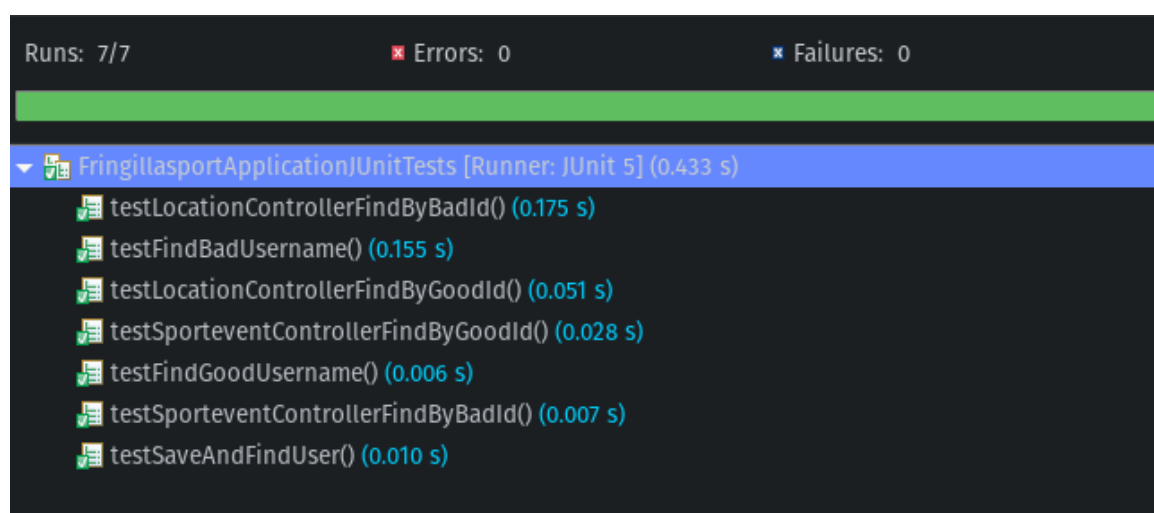
```
1 @Test
2 void testLocationControllerFindByGoodId() throws Exception{
```

```
3      this.mockMvc.perform(get("/api/v1/locations/1")).andDo(print()).andExpect(status().isOK())
4  }
5
6  @Test
7  void testLocationControllerFindByBadId() throws Exception{
8      this.mockMvc.perform(get("/api/v1/locations/1000")).andDo(print()).andExpect(status().is404())
9  }
```

Izvorni kod 5.4: testLocationControllerFindById

Izvođenje ispitnog slučaja: Ispit je uspješan.

Ranije navedene testove pokrenuli smo u razvojnom okruženju Eclipse IDE. Rezultati testova vidljivi su na slici ispod.



Slika 5.1: rezultati JUnit testova

### 5.2.2 Ispitivanje sustava

Proveli smo ispitivanje sustava koristeći radni okvir Selenium<sup>14</sup>. U nastavku ispitujemo 4 ispitna slučaja u kojima testiramo redovne slučajeve, rubne uvjete te namjerno korištenje aplikacije na neočekivani način kako bi se vidjelo na koji način sustav reagira.

Za ispitivanje koristili smo alat **Selenium WebDriver** te programski jezik javu.

<sup>14</sup><https://www.seleniumhq.org/>

**Ispitni slučaj 1: Prijava u sustav**

Da bi korisnik koristio našu web aplikaciju mora se prijaviti u sustav sa već postojećim korisničkim računom ili treba prvo napraviti registraciju pa zatim prijavu. Slijedeći test testira prijavu na stranicu sa već postojećim korisničkim računom za sportaša. Nakon uspješne prijave web aplikacija izbacuje iskočnu poruku koja prikazuje tekst "uspješna prijava". Funkcionalnost je testirana u metodama `testLoginGoodCreds()` i `testLoginBadCreds()`.

**Ulaz:**

1. Otvaranje početne stranice u web pregledniku
2. Klik na gumb prijava
3. Unos korisničkog imena
4. Unos lozinke
5. Klik na gumb prijava

**Očekivani rezultat:**

1. Otvorila se početna stranica
2. Prikzala se stranica za prijavu
3. Popunilo se polje za korisničko ime i šifru
4. Prikazuje se skočna poruka za uspješnu prijavu
5. Prikazuje se profil korisnika

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

```
1 @Test
2 void testLoginGoodCreds() {
3
4     System.setProperty("webdriver.chrome.driver",
5         "/Users/Luka/Applications/WebDriver/chromedriver");
6     WebDriver driver = new ChromeDriver();
7
8     driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
9     driver.get("http://localhost:3000/prijava");
10
11     WebElement element = driver.findElement(
12         By.xpath("//*[@id=\"root\"]//div/div[2]/div[1]/form/div[1]/div[2]/input"));
```



```
13 element.sendKeys("athlete");
14
15 element = driver.findElement(By.xpath(
16     "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[2]/div[2]/input"));
17 element.sendKeys("athlete");
18
19 driver.findElement(By.xpath(
20     "//*[@id=\"root\"] /div/div[2]/div[1]/form/button")).click();
21
22 WebDriverWait wait = new WebDriverWait(driver, 5);
23 Alert alert = wait.until(ExpectedConditions.alertIsPresent());
24 String alertCheck=alert.getText();
25 boolean compRes= alertCheck.contains("Uspjesna");
26
27 assertTrue(compRes);
28 driver.quit();
29 }
```

Izvorni kod 5.5: testLoginGoodCreds

### Ispitni slučaj 2: Prijava s krivim podacima

Analogni test za prijavu napravili smo i krivim korisničkim podacima. Očekivani rezultat je poruka upozorenja sa tekstom "Pogreška"

#### Ulaz:

1. Otvaranje početne stranice u web pregledniku
2. Klik na gumb prijava
3. Unos nepostojećeg korisničkog imena
4. Unos nepostojeće lozinke
5. Klik na gumb prijava

#### Očekivani rezultat:

1. Otvorila se početna stranica
2. Prikzala se stranica za prijavu
3. Popunilo se polje za korisničko ime i šifru
4. Prikazuje se skočna poruka za grešku prilikom prijave

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

```
1 @Test
2 void testLoginBadCreds() {
3     System.setProperty("webdriver.chrome.driver",
4         "/Users/Luka/Applications/WebDriver/chromedriver");
5     WebDriver driver = new ChromeDriver();
6
7     driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
8     driver.get("http://localhost:3000/prijava");
9
10    WebElement element = driver.findElement(By.xpath(
11        "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[1]/div[2]/input"));
12    element.sendKeys("wrongUsername");
13
14    element = driver.findElement(By.xpath(
15        "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[2]/div[2]/input"));
16    element.sendKeys("WrongPassword");
17
18    driver.findElement(By.xpath(
19        "//*[@id=\"root\"] /div/div[2]/div[1]/form/button")).click();
20
21    WebDriverWait wait = new WebDriverWait(driver, 5);
22    Alert alert = wait.until(ExpectedConditions.alertIsPresent());
23    String alertCheck=alert.getText();
24    boolean compRes= alertCheck.contains("Pogreska");
25
26    assertTrue(compRes);
27    driver.quit();
28 }
```

Izvorni kod 5.6: testLoginBadCreds

### Ispitni slučaj 3: Prijava na sportsko događanje

Aplikacija prijavljenim korisnicima nudi mogućnost da se se prijave na neko od nadolazećih sportskih događanja nakon čega sustav prikazuje poruku "Uspješno". Tu funkcionalnost provjerili smo u klasi SeleniumApplyForEventTest metodom testApplyForEvent.

**Ulaz:**

1. Otvaranje početne stranice u web pregledniku
2. Klik na gumb prijava
3. Unos korisničkog imena
4. Unos lozinke
5. Klik na gumb prijava
6. Klik na gumb karta sportskih okupljanja
7. Klik na prvi event u prikazanoj listi sportskih okupljanja
8. Klik na gumb prijava
9. Klik na gumb za potvrdu prijave

**Očekivani rezultat:**

1. Otvorila se početna stranica
2. Prikzala se stranica za prijavu
3. Popunilo se polje za korisničko ime i šifru
4. Prikazuje se skočna poruka za uspješnu prijavu
5. Prikazuje se stranica s korisničkim podacima
6. Prikazuje se stranica s kartom i popisom sportskih okupljanja
7. Prikazuju se detalji o prvom sportskom eventu
8. Prikazuje se skočna poruka za potvrdu prijave
9. Pored sportskog događaja korisniku piše da je prijavljen na njega

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

```
1 @Test
2 void testApplyForEvent() {
3
4     System.setProperty("webdriver.chrome.driver",
5         "/Users/Luka/Applications/WebDriver/chromedriver");
6     WebDriver driver = new ChromeDriver();
7
8     driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
9     driver.get("http://localhost:3000/prijava");
10
11     WebElement element = driver.findElement(By.xpath(
12         "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[1]/div[2]/input"));
```

```
13 element.sendKeys("athlete");
14
15 element = driver.findElement(By.xpath(
16     "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[2]/div[2]/input"));
17 element.sendKeys("athlete");
18
19 driver.findElement(By.xpath(
20     "//*[@id=\"root\"] /div/div[2]/div[1]/form/button")).click();
21
22 WebDriverWait wait = new WebDriverWait(driver, 5);
23 Alert alert = wait.until(ExpectedConditions.alertIsPresent());
24 alert.accept();
25
26 driver.findElement(By.xpath(
27     "//*[@id=\"root\"] /div/div[1]/nav/ul/li[2]/a")).click();
28
29 driver.findElement(By.xpath(
30     "//*[@id=\"root\"] /div/div[2]/div[2]/div[1]/div/h3")).click();
31
32 driver.findElement(By.xpath(
33     "//*[@id=\"root\"] /div/div[2]/div[2]/div[3]/p")).click();
34
35 Alert alert2 = wait.until(ExpectedConditions.alertIsPresent());
36 alert2.accept();
37 Alert alert3 = wait.until(ExpectedConditions.alertIsPresent());
38 boolean checker = alert3.getText().contains("Usp jesno");
39
40 assertTrue(checker);
41 driver.quit();
42 }
```

Izvorni kod 5.7: testApplyForEvent

#### Ispitni slučaj 4: Stvaranje sportske lokacije

Osim prijave na već postojeća sportska događanja, korisnik može u sustav i dodati novu lokaciju na kojoj će se održavati sportska okupljanja. Tu funkcionalnost isprobali smo u klasi SeleniumCreateLocationTest u metodi testCreateLocation().

Stvorena je lokacija pod nazivom "TestnaLokacija". Da bi dodali tu lokaciju simulirali smo klikove na karti kojima bi označili poligon, odnosno površinu koju ta lokacija zauzima na karti.

**Ulaz:**

1. Otvaranje početne stranice u web pregledniku
2. Klik na gumb prijava
3. Unos korisničkog imena
4. Unos lozinke
5. Klik na gumb prijava
6. Klik na gumb stvori sportsku lokaciju
7. Klik na gumb za približavanje karte
8. Klik na gumb za početak ucrtavanja lokacije
9. Odabir 4 točke na mapi
10. Klik na gumb za završetak ucrtavanja lokacije
11. Upis imena lokacije
12. Klik na gumb stvori novu lokaciju

**Očekivani rezultat:**

1. Otvorila se početna stranica
2. Prikazala se stranica za prijavu
3. Popunilo se polje za korisničko ime i šifru
4. Prikazuje se skočna poruka za uspješnu prijavu
5. Prikazuje se stranica s korisničkim podacima
6. Prikazuje se stranica za kreiranje nove sportske lokacije
7. Karta se približi
8. Odaberu se 4 točke na karti
9. Unosi se ime lokacije
10. Prikazuje se skočna poruka o uspješnom kreiranju sportske lokacije

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

```
1 @Test
2 void testCreateLocation() {
3
4     System.setProperty("webdriver.chrome.driver",
```

```
5  "/Users/Luka/Applications/WebDriver/chromedriver");
6  WebDriver driver = new ChromeDriver();
7
8  driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
9  driver.get("http://localhost:3000/prijava");
10
11  WebElement element = driver.findElement(By.xpath(
12  "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[1]/div[2]/input"));
13
14  element.sendKeys("athlete");
15  element = driver.findElement(By.xpath("
16  //*[@id=\"root\"] /div/div[2]/div[1]/form/div[2]/div[2]/input"));
17
18  element.sendKeys("athlete");
19  driver.findElement(By.xpath("
20  //*[@id=\"root\"] /div/div[2]/div[1]/form/button")).click();
21
22  WebDriverWait wait = new WebDriverWait(driver, 5);
23  Alert alert = wait.until(ExpectedConditions.alertIsPresent());
24  alert.accept();
25
26  //Stvori lokaciju
27  driver.findElement(By.xpath("
28  //*[@id=\"root\"] /div/div[1]/nav/ul/li[3]/a")).click();
29
30  //zoom x2
31  driver.findElement(By.xpath("
32  //*[@id=\"root\"] /div/div[2]/div/div[2]/div[1]/div/a[1]")).click();
33
34  driver.findElement(By.xpath("
35  //*[@id=\"root\"] /div/div[2]/div/div[2]/div[1]/div/a[1]")).click();
36
37  //polygon button
38  driver.findElement(By.xpath("
39  //*[@id=\"root\"] /div/div[2]/div/div[2]/div[2]/div/div[1]/div/a")).click();
40
41  Actions actions = new Actions(driver);
42  actions.moveToElement(driver.findElement(By.xpath("
```

```
43     /*[@id=\"root\"]/div/div[2]/div\")), 0, 0);
44
45     actions.moveByOffset(150, 75).click().build().perform();
46     actions.moveByOffset(0, 30).click().build().perform();
47     actions.moveByOffset(30, 0).click().build().perform();
48     actions.moveByOffset(0, -30).click().build().perform();
49
50     //finish
51     driver.findElement(By.xpath("
52     /*[@id=\"root\"]/div/div[2]/div/div[2]/div[2]/div/div[1]/ul/li[1]/a")).click();
53
54     //ime lokacije
55     driver.findElement(By.xpath("
56     /*[@id=\"root\"]/div/div[2]/form/input")).sendKeys("TestnaLokacija");
57
58     //Završi
59     driver.findElement(By.xpath("/*[@id=\"root\"]/div/div[2]/form/h3")).click();
60
61     Alert alert3 = wait.until(ExpectedConditions.alertIsPresent());
62     boolean checker = alert3.getText().contains("uspjesno");
63
64
65     assertTrue(checker);
66     driver.quit();
67 }
```

Izvorni kod 5.8: testCreateLocation

### Ispitni slučaj 5: Pogrešno stvaranje nove lokacije

Proveli smo još jedan test sa dodavanjem lokacije, ali bez da smo "kliknuli" odnosno specificirali gde na karti se ta lokacija nalazi pa zbog toga naravno očekujemo grešku.

#### Ulaz:

1. Otvaranje početne stranice u web pregledniku
2. Klik na gumb prijava
3. Unos korisničkog imena

4. Unos lozinke
5. Klik na gumb prijava
6. Klik na gumb stvori sportsku lokaciju
7. Upis imena lokacije
8. Klik na gumb stvori novu lokaciju

**Očekivani rezultat:**

1. Otvorila se početna stranica
2. Prikazala se stranica za prijavu
3. Popunilo se polje za korisničko ime i šifru
4. Prikazuje se skočna poruka za uspješnu prijavu
5. Prikazuje se stranica s korisničkim podacima
6. Prikazuje se stranica za kreiranje nove sportske lokacije
7. Unosi se ime lokacije
8. Prikazuje se skočna poruka o neuspješnom kreiranju sportske lokacije

**Rezultat:** Sva očekivanja su zadovoljena. Selenium test je prošao.

```
1
2 @Test
3 void testCreateLocation() {
4
5     System.setProperty("webdriver.chrome.driver",
6         "/Users/Luka/Applications/WebDriver/chromedriver");
7     WebDriver driver = new ChromeDriver();
8
9     driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
10    driver.get("http://localhost:3000/prijava");
11
12    WebElement element = driver.findElement(By.xpath(
13        "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[1]/div[2]/input"));
14
15    element.sendKeys("athlete");
16    element = driver.findElement(By.xpath(
17        "//*[@id=\"root\"] /div/div[2]/div[1]/form/div[2]/div[2]/input"));
18
19    element.sendKeys("athlete");
20    driver.findElement(By.xpath(
```

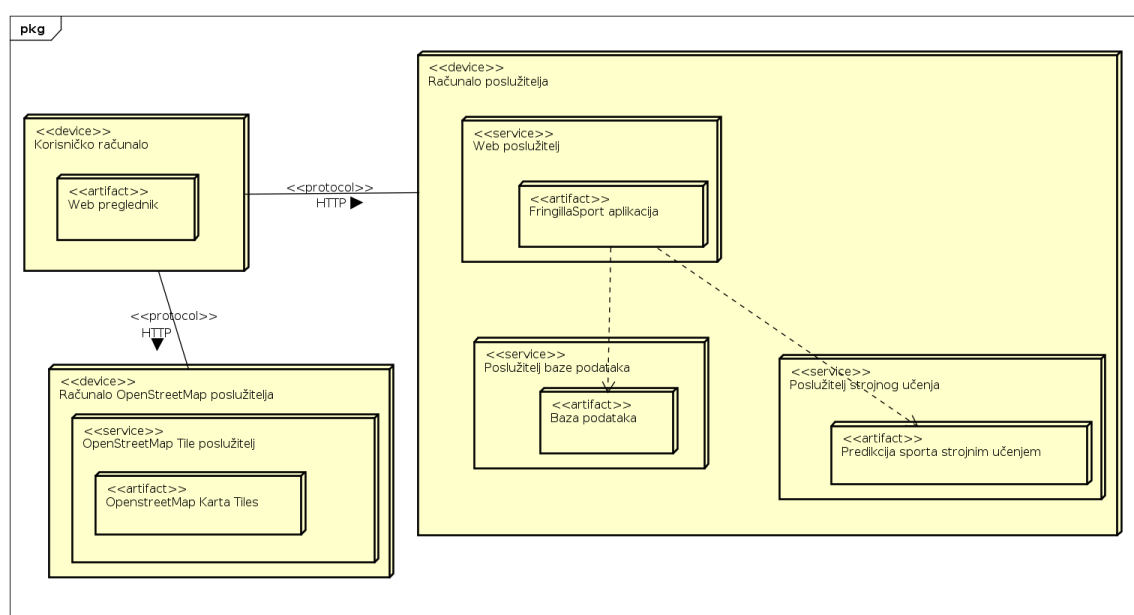


```
21     "//*[@id=\"root\"]/div/div[2]/div[1]/form/button")).click();
22
23     WebDriverWait wait = new WebDriverWait(driver, 5);
24     Alert alert = wait.until(ExpectedConditions.alertIsPresent());
25     alert.accept();
26
27     //Stvori lokaciju
28     driver.findElement(By.xpath(
29         "//*[@id=\"root\"]/div/div[1]/nav/ul/li[3]/a")).click();
30
31
32     //ime lokacije
33     driver.findElement(By.xpath(
34         "//*[@id=\"root\"]/div/div[2]/form/input")).sendKeys("TestnaLokacija");
35
36     //Završi
37     driver.findElement(By.xpath(
38         "//*[@id=\"root\"]/div/div[2]/form/h3")).click();
39
40     Alert alert3 = wait.until(ExpectedConditions.alertIsPresent());
41     boolean checker = alert3.getText().contains("pogreske");
42
43     assertTrue(checker);
44     driver.quit();
45 }
```

Izvorni kod 5.9: testCreateLocation

## 5.3 Dijagram razmještaja

Dijagram razmještaja na slici prikazuje odnos sklopovlja i programske potpore na njemu. Na korisničkom računalu web preglednik HTTP protokolom pristupa web aplikaciji na poslužiteljskom računalu. Na poslužiteljskom računalu nalaze se Web poslužitelj na kojem je pokrenuta web aplikacija te poslužitelj baze podataka na kojem se nalazi baza podataka. Također nalazi se i poslužitelj strojnog učenja na kojemu se nalazi predikcija sporta strojnim učenjem. Korisniku potrebne podatke dostavlja web aplikacija koja ih dobiva komunikacijom s bazom podataka i s predikcijom strojnog učenja.



Slika 5.2: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Aplikacija je puštena u pogon na ubuntu 18.04 serveru. Te je na serveru potrebno imati instaliranu Javu 11.0, Node.js i Python 3.0.

### 5.4.1 Instalacija postgresql baze podataka

Potrebno je instalirati postgresql bazu podataka s naredbom:

```
$ sudo apt install postgresql postgresql-contrib
```

Potom kreiramo bazu podataka za aplikaciju:

```
$ createdb-h localhost -p 5432 -U postgres fringilladb password pass
```

Navedene podatke potrebno je unjesti u datoteku backenda na lokaciji backend/-src/main/resources/application.properties:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/fringilladb
spring.datasource.username=postgres
spring.datasource.password=pass
spring.jpa.hibernate.ddl-auto=create
```

### 5.4.2 Instalacija drugih potrebnih paketa

Za ostale komponente aplikacije potrebno je instalirati programe navedenim naredbama.

```
$ sudo apt install software-properties-common
$ sudo apt install nodejs python3 python3-pip default-jre
```

### 5.4.3 Pokretanje backenda

Potrebno se je pozicionirati u direktorij izvorniKod/backend te pokrenuti naredbu:

```
$ ./mvn spring-boot:run
```

### 5.4.4 Pokretanje frontenda

Frontend aplikacije pokreće se pozicioniranjem u direktorij izvorniKod/frontend te izvršavanjem naredbi:

```
$ npm install
```

```
$ npm start
```

Prva naredba instalira potrebne pakete, dok druga pokreće frontend.

### 5.4.5 Pokretanje servera za predlaganje sporta

Za pokretanje python servera potrebno se je pozicionirati u direktorij izvorniKod/-mlSportPredict te pokrenuti naredbu:

```
$ python server.py
```

Napomena zbog trenutne konfiguracije potrebno je stvoriti link između direktorija izvorniKod/backend/uploads i izvorniKod/frontend/public/documentation.

## 6. Zaključak i budući rad

Naša grupa imala je zadatak izraditi aplikaciju koja omogućuje jednostavnije povezivanje sportaša, trenera i iznajmljivača sportskih prostora te taj proces pojednostavljuje i ubrzava. Ideja aplikacije je proizašla od članova naše grupe, zbog toga su svi članovi bili dodatno motivirani da uspješno i kvalitetno izradimo aplikaciju te uz nju napišemo kvalitetnu dokumentaciju.

Za izradu ove aplikacije odlučili smo se koristiti tehnologije Spring Boot za backend, te React za frontend. Najveći izazov je bio naučiti koristiti navedene tehnologije te povezati frontend i backend. Da bismo lakše prevladali ovaj izazov podijelili smo grupu u 2 manja tima, jedan za frontend te jedan za backend, svaki tim je sadržavao po 3 člana naše grupe. Voditelj naše grupe je vodio oba tima, zbog toga su timovi tijekom cijelog projekta bili cijelo vrijeme na istoj razini te ni jedan tim nije zaostajao za drugim. Ovakav raspored te dobra atmosfera i komunikacija među članovima učinila je rad cijele grupe produktivnijim te s time i olakšala prevladavanje svih izazova i prepreka.

Projekt je bio podijeljen u dvije faze kroz 15 tjedana. Prva faza je uključivala okupljanje tima, smišljanja ideje za aplikaciju, podjelu u timove, dodjelu projektnog zadatka, rad na dokumentaciji i implementaciju osnovnih funkcionalnosti aplikacije. Druga faza je uključivala završetak svih funkcionalnosti backenda i frontenda, testiranje aplikacije te dovršetak dokumentacije i cijelog projekta.

Prilikom izrade ovog projekta naučili smo uz dosada navede tehnologije koristiti sustav za upravljanje izvornim kodom Git te alate TEXstudio i Astah koji služe za pisanje dokumentacije i crtanje dijagrama. Osim navedenih tehnologija naučili smo pisati kvalitetnu dokumentaciju te kvalitetno oblikovati kod koji je iznimno važan za daljnu nadogradnju projekta, testiranje i tražanje grešaka u kodu. Možda i najvažnija lekcija naučena prilikom izrade ovog projekta je važnost grupnog rada, ispunjavanje svojih obaveza na vrijeme te suradnja i komunikacijama s kolegama. Sve navedene naučene lekcije su od iznimne važnosti za naš daljnji razvoj i uspješnost u struci.

Za brže i kvalitetnije ostvaranje projekta potrebno je imati već određeno iskustvo u korištenim tehnologijama te iskustvo u sličnim projektima. Osim toga, jako

je važno raditi na što manje projekata paralelno, na taj način imali bi više vremena i veći fokus za naš.

Naša aplikacija "FringillaSport" mogla bi se nadograditi na brojne načine. Jedna od ideja za dodatne funkcionalnosti je ocjenjivanje sportaša, trenera te prostora za sport, sportaši bi nakon sudjelovanja na nekom događaju mogli ocijeniti ostale sportaše, trenera te prostor za sport s ciljem kvalitetnije usluge aplikacije i informacijama o kvaliteti sportskih događaja.

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

*U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.*

#### 1. sastanak

- Datum: 5. listopada 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - sastanak a asistentom
  - objašnjenje zadatka
  - rasprava o predloženom zadatku

#### 2. sastanak

- Datum: 17. listopada 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - rasprava o funkcionalnim zahtjevima i obrascima uporabe
  - razgovor o iskustvima pojedinih članova u različitim tehnologijama

#### 3. sastanak

- Datum: 19. listopada 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - sastanak s asistentom
  - dogovor o tehnologijama koje će se koristiti u projektu

#### 4. sastanak

- Datum: 28. listopada 2020.

- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - dogovor o daljnoj podjeli zadataka
  - rasprava o bazi podataka

#### 5. sastanak

- Datum: 5. studenoga 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - povezivanje frontenda i backenda
  - detaljnija rasprava o bazi podataka

#### 6. sastanak

- Datum: 9. studenoga 2020.
- Prisustvovali: M.Marfat, L.Nuić, M.Rašić, L.Srdarev
- Teme sastanka:
  - sastanak s asistentom
  - pregled dosadašnjeg rada

#### 7. sastanak

- Datum: 2. prosinca 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - sastanak s asistenticom - evaluacija dosadašnjeg rada

#### 8. sastanak

- Datum: 23. prosinca 2020.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, B.Paradžik, M.Rašić, L.Srdarev
- Teme sastanka:
  - pregled napravljenog dijela projekta
  - plan nastavka projekta
  - podjela zadataka

#### 9. sastanak

- Datum: 29. prosinca 2020.



- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, M.Rašić, L.Srdarev, B.Paradžik
- Teme sastanka:
  - pregled napravljenog od zadnjeg sastanka
  - podjela novih zadataka za backend i frontend

## 10. sastanak

- Datum: 2. siječnja 2021.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, M.Rašić, L.Srdarev
- Teme sastanka:
  - pregled napravljenog od zadnjeg sastanka
  - podjela novih zadataka za backend i frontend

## 11. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: L.Ilić, L.Nuić, B.Paradžik, L.Srdarev
- Teme sastanka:
  - pregled napravljenih funkcionalnosti backenda
  - rješavanje dosadašnjih problema u implementaciji backenda

## 12. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: G.Crnogorac, M.Marfat, L.Nuić, M.Rašić
- Teme sastanka:
  - pregled napravljenih funkcionalnosti frontenda
  - podjela novih zadataka za frontend

## 13. sastanak

- Datum: 6. siječnja 2021.
- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, M.Rašić, L.Srdarev, B.Paradžik
- Teme sastanka:
  - pregled napravljenih funkcionalnosti backenda
  - pregled rada aplikacije
  - utvrđivanje i rješavanje grešaka aplikacije
  - aktualizacija dokumentacije

## 14. sastanak

- Datum: 7. siječnja 2021.

- Prisustvovali: G.Crnogorac, L.Ilić, M.Marfat, L.Nuić, M.Rašić, L.Srdarev, B.Paradžik
- Teme sastanka:
  - sastanak s asistentom - demonstracija alfa inacice

## Tablica aktivnosti

### Kontinuirano osvježavanje

*Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.*

	Lovro Nućić	Grgur Crnogorac	Luka Ilić	Marko Marfat	Berislav Paradžik	Marin Rašić	Luka Srdarev
Upravljanje projektom	10						
Opis projektnog zadatka					2	3	
Funkcionalni zahtjevi			1	1		1	
Opis pojedinih obrazaca			1	1		1	
Dijagram obrazaca		4					
Sekvencijski dijagrami				3			
Opis ostalih zahtjeva				1			
Arhitektura i dizajn sustava			3				8
Baza podataka			4				10
Dijagram razreda	10						
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti		1.5					
Korištene tehnologije i alati							
Ispitivanje programskog rješenja			12				12
Dijagram razmještaja		1.5					
Upute za puštanje u pogon	1						
Dnevnik sastajanja	1					0.5	
Zaključak i budući rad							
Popis literature							

	Lovro Nuić	Grgur Crnogorac	Luka Ilić	Marko Marfat	Berislav Paradžik	Marin Rašić	Luka Srdarev
<i>Backend</i>	80		25		4		25
<i>Frontend</i>		45		35		40	

## Dijagrami pregleda promjena

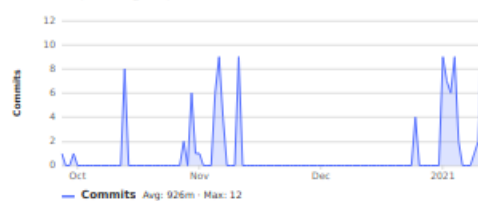
### Commits to master

Excluding merge commits. Limited to 6,000 commits.



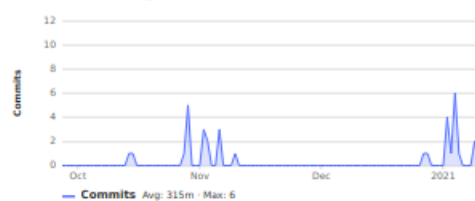
### Inuic

100 commits (lovro.inuic@fer.hr)



### mmarfat

34 commits (marko.marfat@fer.hr)



### marin-rasic

23 commits (mrasic123@gmail.com)



### bparadzik

20 commits (berislav.paradzik@fer.hr)



### Grgur Crnogorac

20 commits (grgur.crnogorac@gmail.com)



### Luka Srdarev

14 commits (luka.srdarev@fer.hr)



### Luka Ilić

14 commits (luka.ilic@fer.hr)



Slika 6.1: Dijagram pregleda promjena