

# Detaljan opis problema i potencijalna korist projekta

---

## Opis problema

---

Unatoč rastućoj popularnosti i raznovrsnosti tabletop role-playing igara (u nastavku TTRPG), mnogi igrači, posebno oni noviji svijetu TTRPG-eva, suočavaju se s izazovom pronalaska adekvatnih grupa za igranje. Trenutačno ne postoji centralizirana platforma koja bi sustavno povezivala igrače, pružala jednostavan način za pronalazak novih grupa ili olakšavala komunikaciju s iskusnim voditeljima igara. Većina entuzijasta koje zanima svijet ovakvih igara se zbog toga osjećaju izolirano, a potraga za grupama s istim interesima često postaje dugotrajna, frustrirajuća i najčešće neuspješna. Ovakvo stanje ima negativan utjecaj na TTRPG zajednicu zbog čega se propušta jasno dostupna prilika za jačanje i širenje domaće TTRPG scene.

## Korist projekta

---

Kako bi se riješio problem izolacije igrača i stvorilo poticajno okruženje za rast i razvoj TTRPG zajednice, ovim se projektom razvija centralizirana aplikacija koja bi služila kao sveobuhvatan alat za igrače i voditelje igara. Ova platforma ne bi bila samo prostor za traženje grupa, već i mjesto gdje se igrači i voditelji igara mogu međusobno povezivati, organizirati događaje i igre te graditi zajednicu.

## Ciljna skupina korisnika

---

Ciljna skupina korisnika za ovu web aplikaciju obuhvaća širok spektar entuzijasta TTRPG-eva. Taj se spektar može svrstati u nekoliko ključnih kategorija:

### Početnici u TTRPG-evima

Uglavnom mlađi igrači pretežno u dobnoj skupini 18-30 godina, zainteresirani za ulazak u svijet TTRPG-eva, ali bez iskustva ili kontakata unutar zajednice.

### Iskusni igrači TTRPG-eva

Igrači s više godina iskustva koji su već igrali razne sustave (Dungeons & Dragons, Pathfinder, QuestRPG, Knave, itd.). Znaju kakve igre traže i što žele izbjegavati, ali zbog nekog razloga nemaju pristup grupi s istim interesima preko svojih kontakata.

## Voditelji TTRPG igara

Igrači koji znaju i žele voditi igre, često iskusni igrači (ali mogu biti i početnici), uživaju u stvaranju priča i svjetova, ali nemaju grupu s kojom bi mogli te svjetove i priče istraživati i graditi.

## Izdavači TTRPG sadržaja

Organizacije koje izdaju TTRPG sadržaj i žele skupiti igrače za testiranje sadržaja koji je još u proizvodnji. Žele pronaći raznovrsne skupine igrača.

## Dućani s igrama, udruge

Dućani ili udruge koje žele organizirati veće događaje u svojim prostorima. Traže grupu igrača koji bi popunili njihov prostor na raznim događajima.

# Okviran pregled zahtjeva

---

## Funkcijski zahtjevi

---

### 1. Korisničke funkcionalnosti:

- **Neregistrirani korisnici:**
  - pretraživanje javno dostupnih nadolazećih igara
  - registracija u sustav
  - prijava u sustav
- **Registrirani (privatni) korisnici:**
  - pretraživanje svih nadolazećih igara
  - pregled nadolazećih događaja (igre na koje su prijavljeni ili njihove vlastite)
  - stvaranje vlastitih igara (online i lokaliziranih)
  - odobravanje/odbijanje prijava na vlastite igre
  - pregled i uređivanje vlastitih igara
  - prijava na igre
- **Registrirani (poslovni) korisnici:**
  - pretraživanje svih nadolazećih igara
  - pregled nadolazećih događaja (vlastite igre)
  - uređivanje profila organizacije vidljivog svim korisnicima
  - stvaranje vlastitih igara (online i točno-lokacijskih)

- odobravanje/odbijanje prijava na vlastite igre
- pregled i uređivanje vlastitih igara

## 2. Tipovi igara:

- **Javne i privatne igre:**
  - javne igre su dostupne za pregled neregistriranim korisnicima
  - privatne igre su dostupne samo registriranim korisnicima
- **Online i lokalizirane igre:**
  - online igre nemaju parametar lokacije
  - lokalizirane igre prikazuju okvirnu lokaciju, a točna lokacija je dostupna tek nakon odobrene prijave
  - poslovni korisnici mogu organizirati "točno-lokacijske" igre s javno prikazanom lokacijom
- **Igre s prijavom i bez prijave:**
  - igre s prijavom zahtijevaju od igrača ispunjavanje prijavnog obrasca
  - igre bez prijave omogućuju neposredno pridruživanje igri bez odobrenja organizatora
- **Popunjene i dostupne igre:**
  - popunjene igre su i dalje vidljive u pretraživaču, ali bez mogućnosti prijave
  - dostupne igre omogućuju prijavu ili dodavanje igrača

## Nefunkcijski zahtjevi

---

### 1. Performanse:

- sustav mora omogućiti brzo pretraživanje igara (manje od 5 sekundi), uključujući filtriranje po različitim kriterijima
- aplikacija mora biti respozivna i omogućiti optimalno iskustvo na različitim veličinama ekrana

### 2. Sigurnost:

- korisnički podatci trebaju biti zaštićeni—neregistrirani korisnici ne smiju moći vidjeti privatne igre; ne bi trebalo biti moguće vidjeti točnu lokaciju lokalizirane igre ako prijava nije odobrena; etc.
- prijavljeni korisnici trebaju imati siguran i zaštićen pristup svojim osobnim podacima i igrama

### 3. Skalabilnost:

- aplikacija mora podržavati velik broj istovremenih korisnika (više od 100) bez gubitka performansi
- treba omogućiti jednostavno širenje funkcionalnosti u budućnosti (modularni dizajn)

#### 4. Održavanje:

- aplikacija mora biti lako održiva, uz mogućnost redovitog ažuriranja sustava bez prekida u radu
- kôd treba biti dobro dokumentiran radi lakšeg održavanja i dodavanja novih značajki

## Prilagodbe rješenja

---

Zbog prirode problema i raznolikog okruženja koje TTRPG-evi pružaju svojim igračima, rješenje mora biti prilagodljivo za raznoliku uporabu u TTRPG zajednici. U nastavku su opisani načini na koje je TTRPGFinder prilagođen kako bi udovoljio ovakvim zahtjevima.

## Vrste igara

---

Već spomenut sustav koji kategorizira igre kroz razne parametre jedan je način na koji je TTRPGFinder web aplikacija prilagođeno rješenje. Parametri prije napomenuti oni su za koje se Nat20Mappers tim odlučio, ali je moguće dodati razne druge parametre za još efikasniju pretragu igara.

## Uređivanje profila

---

Poslovni korisnici imaju svoje profile na TTRPGFinder stranici, te imaju sljedeća polja koja mogu uređivati po svojoj volji:

- naziv organizacije/udruga
- logo (vizualni identitet organizacije u obliku loga)
- opis (kratak opis organizacije)
- adresa
- telefon
- email adresa
- link na službenu web stranicu

## Stvaranje posebne forme za prijave na igru

---

Svi korisnici pri stvaranju igre na TTRPGFinder sustavu imaju mogućnost napraviti vlastitu formu za prijavu na njihovu igru koja će se prikazivati korisnicima koji se žele prijaviti. Njihova forma može imati minimalno 1, a maksimalno 100 pitanja na koje drugi korisnik mora odgovoriti u prijavi na igru. Ova pitanja mogu biti bilo kakva i korisnik sam određuje kojim redoslijedom će ona biti prikazana.

## Stvaranje posebnih igara

---

Svi korisnici pri stvaranju igre na TTRPGFinder sustavu, imaju kompletnu slobodu u stvaranju igre te uređuju sljedeće informacije:

- naziv igre
- tip igre (naznačeno kategorijama prije spomenutim)
- kratki opis
- očekivana duljina
- složenost
- preporučena razina iskustva
- potrebni materijali
- sustav u kojem se igra
- je li homebrew (prati li sva pravila ili ne)
- komunikacijski kanali
- timestamp početka
- lokacija (ako je lokalizirana igra)
- vremenska zona (ako je online)
- slike prošlih igara ili crteži tematike igre (uglavnom .png ili .jpg datoteke)

## Slična rješenja

---

### Roll20

---

Roll20 je jedna od najpopularnijih TTRPG web stranica koja se najviše koristi za virtualne/online igre. Kao jednu od funkcionalnosti koju nudi svojim korisnicima je funkcija traženja grupa, ali to nije glavna funkcionalnost stranice već jedna od sporednih.

### Startplaying.games

---

Startplaying.games web je platforma za traženje TTRPG igara i grupa te je najbližnja TTRPGFinder projektu u funkcionalnostima. Omogućuje igračima traženje igara i grupa za igranje, ali dio koji ovu platformu razlikuje od TTRPGFinder web aplikacije je činjenica da je glavni fokus startplaying.games platforme omogućavanje rada iskusnim voditeljima igara kako bi mogli dobiti prihod od svojih vještina u igranju, dok to TTRPGFinder web aplikaciji nije cilj.

## Reddit (lfg)

---

Reddit podforum kojem je cilj spajati igrače u grupe i spajati igrače s grupama koje imaju iste interese kao oni sami. Ciljem vrlo slično TTRPGFinder web aplikaciji, ali izvedbom vrlo različito rješenje. Dok bi TTRPGFinder sam implementirao sustav korisnika i igara, ovdje se sve mora prilagoditi Reddit sustavu i pravilima pa tako igrači moraju imati Reddit račun, a igre nije lako tražiti jer se mora koristiti Reddit tražilica.

## Mogućnosti nadogradnje rješenja

---

Iako je rješenje modelirano, postoje nadogradnje s kojima bi rješenje bolje funkcioniralo. Iako ove nadogradnje nisu dio zadatka, njihova je implementacija moguća ako tim ima dovoljno vremena za osmisлити ju i zastupiti na web aplikaciji TTRPGFinder.

## Napredna pretraživanja

---

Napredni filteri osim onih osmišljenih, kao pretraživanje po temama igara, žanrovima, i raznim sistemima igara.

## Sustav ocjenjivanja

---

Nakon odigrane igre bi igrači mogli imati priliku ostaviti ocjenu na voditelja igre (bio to privatni ili poslovni korisnik) te bi se ta ocjena prikazivala kad god netko pritisne na ime korisnika.

## Kalendar

---

Implementacije kompliciranijeg sustava kalendara koji pruža korisnicima preglednost svih njihovih događaja na jednom mjestu, s opcijom podcrtavanja bilo kakvog vremenskog preklapanja.

## Gamifikacija

---

Uvođenje sustava postignuća gdje korisnici mogu osvajati nagrade ili bedževe za sudjelovanje u

igrama ili ostavljanje ocjene. Stvoriti kakvu rang listu korisnika kako bi se potaklo sudjelovanje na igrama.

## Sustav preporuka

---

Uvođenje sustava preporuka koji s informacijama o korisnikovim prijašnjim igrama i konekcijama s ostalim korisnicima preporučuje korisniku igre na koje još nije prijavljen.

## Funkcijski zahtjevi

---

### Dionici:

1. Neregistrirani korisnici
2. Privatni korisnici
3. Poslovni korisnici
4. Razvojni tim
5. Administratori sustava
6. Menadžment
7. Tehnička podrška
8. Zajednica entuzijasta TTRPG-eva

### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:

Prijava u sustav (UC1), registracija u sustav (UC2), pretraživanje javnih igara (UC3)

2. Privatni korisnik (inicijator) može:

Pretraživati igre (UC3), stvarati vlastite igre (UC4), uređivati vlastite stvorene igra (UC5), pregledavati nadolazeće događaje na koje je prijavljen (UC6), pregledavati prijave na svoje stvorene igre (UC7)

Pregledavati profile poslovnih korisnika (UC3.1), prijavljivati se na igre (UC3.2), stvarati posebne forme za prijave za vlastite igre (UC4.1), označavati prijave odobravanjem/ odbijanjem (UC7.1)

3. Poslovni korisnik (inicijator) može:

Pretraživati igre (UC3), stvarati vlastite igre (UC4), uređivati vlastite stvorene igra (UC5),

pregledavati nadolazeće događaje na koje je prijavljen (UC6), pregledavati prijave na svoje stvorene igre (UC7)

Pregledavati profile poslovnih korisnika (UC3.1), stvarati posebne forme za prijave za vlastite igre (UC4.1), označavati prijave odobravanjem/odbijanjem (UC7.1), uređivati vlastiti profil (UC8)

4. OpenStreetMap (sudionik).

## Obrasci uporabe

### Opis obrazaca uporabe

#### UC 1 - Registracija korisnika u sustav

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Registracija korisnika u sustav
- Sudionici: /
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik ispunjava formu za registraciju i šalje POST request s ispunjenim informacijama i svojim SSID-em.
  - ii. Aplikacija pregledava informacije, provjerava jedinstvenost korisničkog imena i poštivanje pravila forme.
  - iii. Aplikacija prikazuje korisniku da je registracija uspješno zaprimljena.
- Opis mogućih odstupanja:
  - i. Ako korisničko ime nije jedinstveno ili postoji greška u formi, korisniku se prikazuje poruka koja ga informira o problemu i vraća forma s pogrešnim podacima izbrisanim.
  - ii. Korisnik zatim ponavlja proces ispunjavanja forme za registraciju.

#### UC 2 - Prijava korisnika u sustav

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Prijava korisnika u sustav
- Sudionici: /
- Preduvjet: Korisnik mora imati registrirani račun.



- Opis osnovnog tijeka:
  - i. Korisnik ispunjava formu za prijavu i šalje zahtjev s ispunjenim informacijama i svojim SSID-em.
  - ii. Aplikacija pregledava informacije u prijavi, provjerava korisničko ime i lozinku, te pridružuje korisnički račun SSID-u.
  - iii. Aplikacija prikazuje stranicu s korisničkim računom.
- Opis mogućih odstupanja:
  - i. Ako korisnik unese krivu lozinku, korisničko ime ne postoji, ili registracija nije odobrena, korisnik dobiva poruku koja ga obavještava o tome.

### UC 3 - Pretraga igara

- Glavni sudionik: Korisnik
- Cilj: Pretraga nadolazećih igara s određenim filterima
- Sudionici: OpenStreetMap API
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik određuje filtere za pretragu igara i šalje zahtjev s filterima i SSID-em.
  - ii. Aplikacija pretražuje igre u sustavu koje prolaze sve filtere, te filtrira privatne igre ako SSID nije povezan s korisničkim računom.
  - iii. Aplikacija prikazuje osnovne informacije o filtriranim igrama i njihove kreatore.
- Opis mogućih odstupanja:
  - i. (UC3.1) Ako korisnik klikne na profil poslovnog korisnika koji je stvorio igru, prikazuju se osnovne informacije, lokacija (OSM API) i link na profil tog poslovnog korisnika.
  - ii. (UC9) U slučaju da je SSID povezan s privatnim računom, korisniku se prikazuje i mogućnost prijave na igru pored same igre.
  - iii. a) Ako se korisnik odluči prijaviti na igru, on dobije formu za prijavu na igru koju može ispuniti i poslati na server.
  - iv. b) U bilo kojem trenutku korisnik može odustati od prijave i vraća se na pretragu igara.

### UC 4 - Stvaranje vlastitih igara

- Glavni sudionik: Registrirani korisnik
- Cilj: Stvaranje igre
- Sudionici: OpenStreetMap API
- Preduvjet: /

- Opis osnovnog tijeka:
  - i. Korisnik ispunjava formu za stvaranje igre i šalje zahtjev s informacijama i SSID-em.
  - ii. Jedna od informacija za ispuniti je lokacija na karti, koja se prikazuje preko OSM API akтора.
  - iii. Aplikacija provjerava podatke i pravila forme, te je li SSID povezan s korisnikom.
  - iv. Aplikacija prikazuje novo-stvorenu igru.
- Opis mogućih odstupanja:
  - i. (UC4.1) Ako igra uključuje dodatne forme za prijavu, korisnik može kreirati vlastitu formu za prijavu na tu igru i poslati ju s igrom.

## UC 5 - Uređivanje vlastitih stvorenih igara

- Glavni sudionik: Registrirani korisnik
- Cilj: Uređivanje svojstava stvorenih igara
- Sudionici: /
- Preduvjet: Korisnik mora imati barem jednu postojeću igru.
- Opis osnovnog tijeka:
  - i. Korisnik ispunjava formu za promjenu postavki igre i šalje POST request s informacijama i SSID-em.
  - ii. Aplikacija provjerava podatke, pravila forme i povezanost korisnika s igrom.
  - iii. Aplikacija prikazuje novo uređenu igru.

## UC 6 - Pregled nadolazećih događaja

- Glavni sudionik: Registrirani korisnik
- Cilj: Pregled nadolazećih igara
- Sudionici: /
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik automatski traži podatke o nadolazećim igrama.
  - ii. Aplikacija pretražuje igre na kojima je korisnik prijavljen ili ih je stvorio.
  - iii. Aplikacija prikazuje listu nadolazećih događaja i njihove datume.

## UC 7 - Pregled prijava na igru

- Glavni sudionik: Registrirani korisnik
- Cilj: Pregled svih aktivnih prijava na igru

- Sudionici: /
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik šalje zahtjev za pregled prijava na igru.
  - ii. Aplikacija vraća listu prijava s detaljima.
  - iii. Aplikacija prikazuje sve prijave u listi.
- Opis mogućih odstupanja:
  - i. (UC7.1) Korisnik može označiti prijave kao prihvaćene, odbijene ili ih izbrisati ako se ne dvoumi.

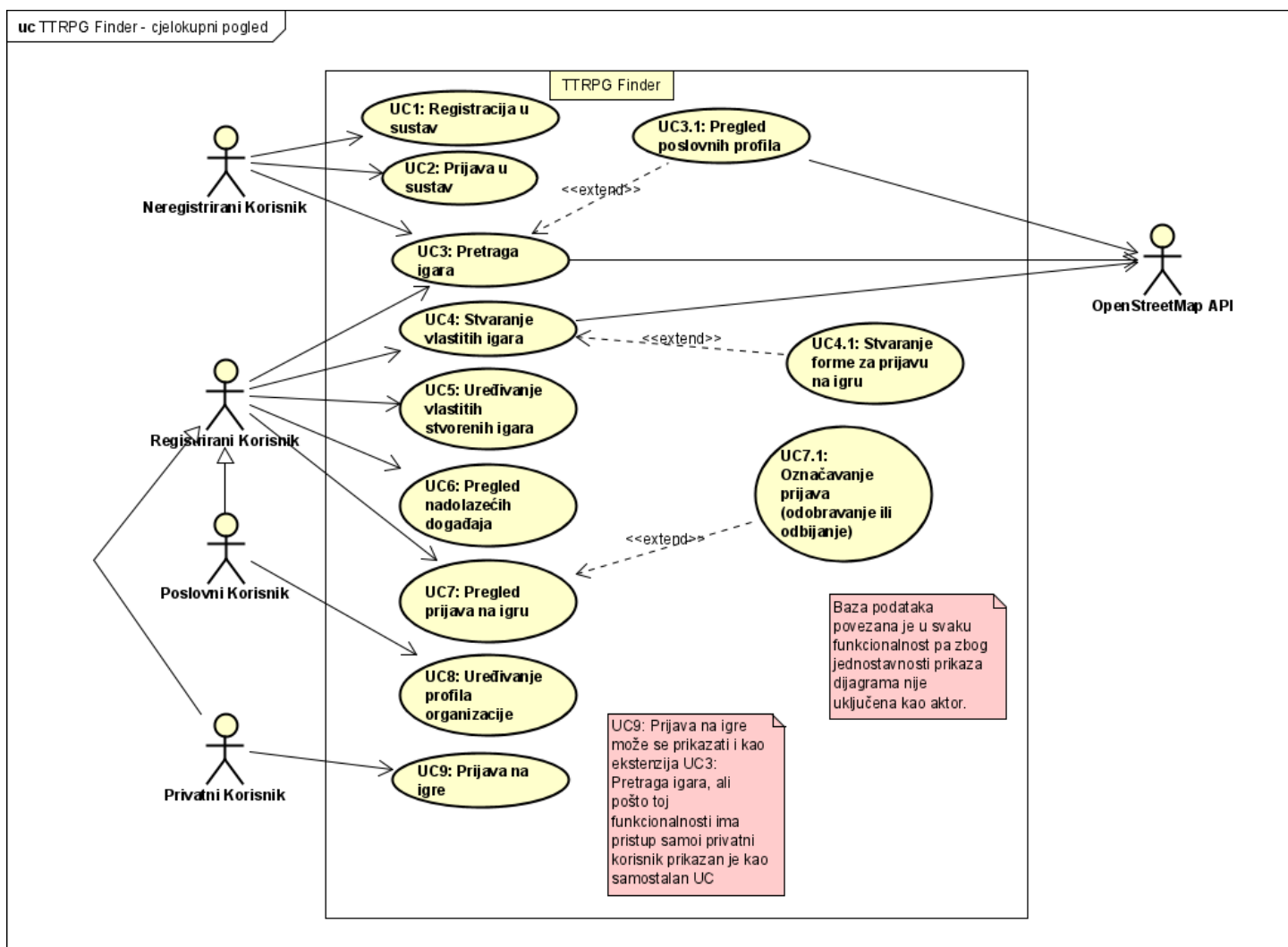
## UC 8 - Uređivanje profila organizacije

- Glavni sudionik: Poslovni korisnik
- Cilj: Uređivanje informacija na profilu organizacije
- Sudionici: /
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik pristupa profilu organizacije i dobiva formu za uređivanje podataka.
  - ii. Aplikacija provjerava podatke, pravila forme i povezanost SSID-a s poslovnim korisnikom.
  - iii. Aplikacija prikazuje ažurirane informacije na profilu.

## UC 9 - Prijava na igre

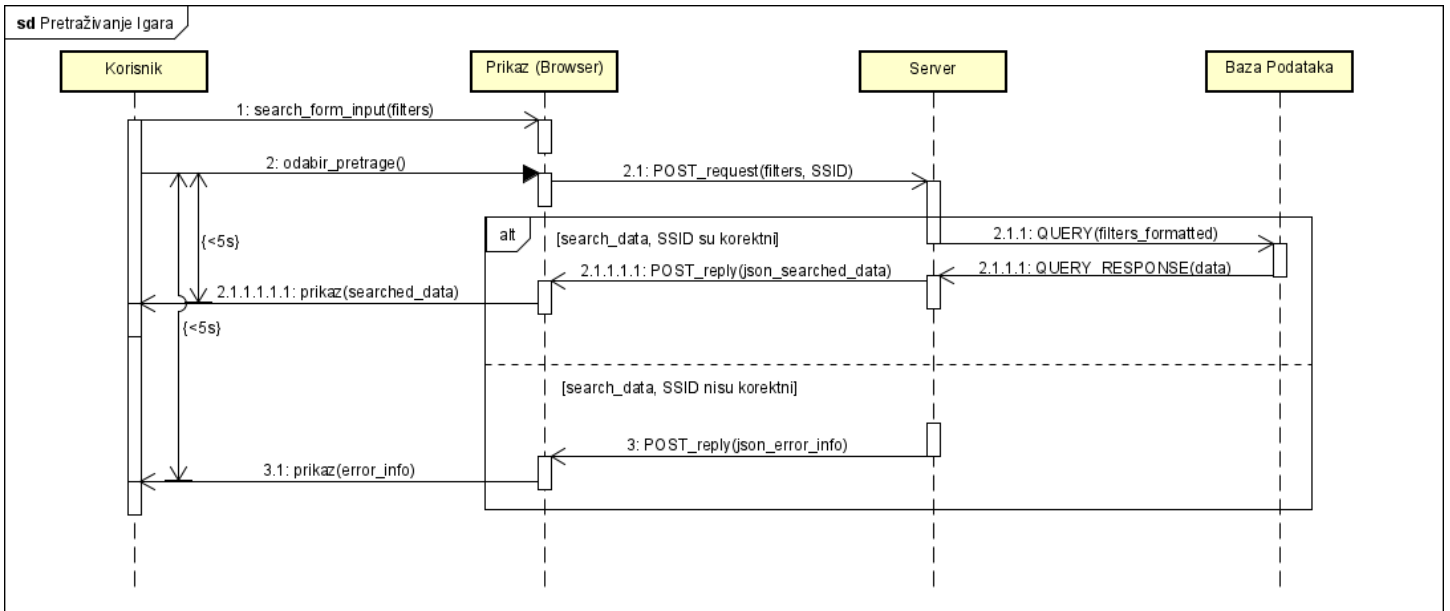
- Glavni sudionik: Privatni korisnik
- Cilj: Prijava na igru
- Sudionici: /
- Preduvjet: /
- Opis osnovnog tijeka:
  - i. Korisnik klikne na prijavu za igru i šalje zahtjev za prijavu na igru (ili gumb za prijavu ili formu za prijavu).
  - ii. Aplikacija provjerava SSID i korisničko ime, te šalje način prijave.
  - iii. Korisnik šalje prijavu na server (ili pritisne gumb za prijavu, ili ispuni formu za prijavu pa zatim pritisne gumb za prijavu).
  - iv. Aplikacija pregledava podatke i informira korisnika o uspjehu ili grešci u prijavi.

# Dijagram obrazaca uporabe

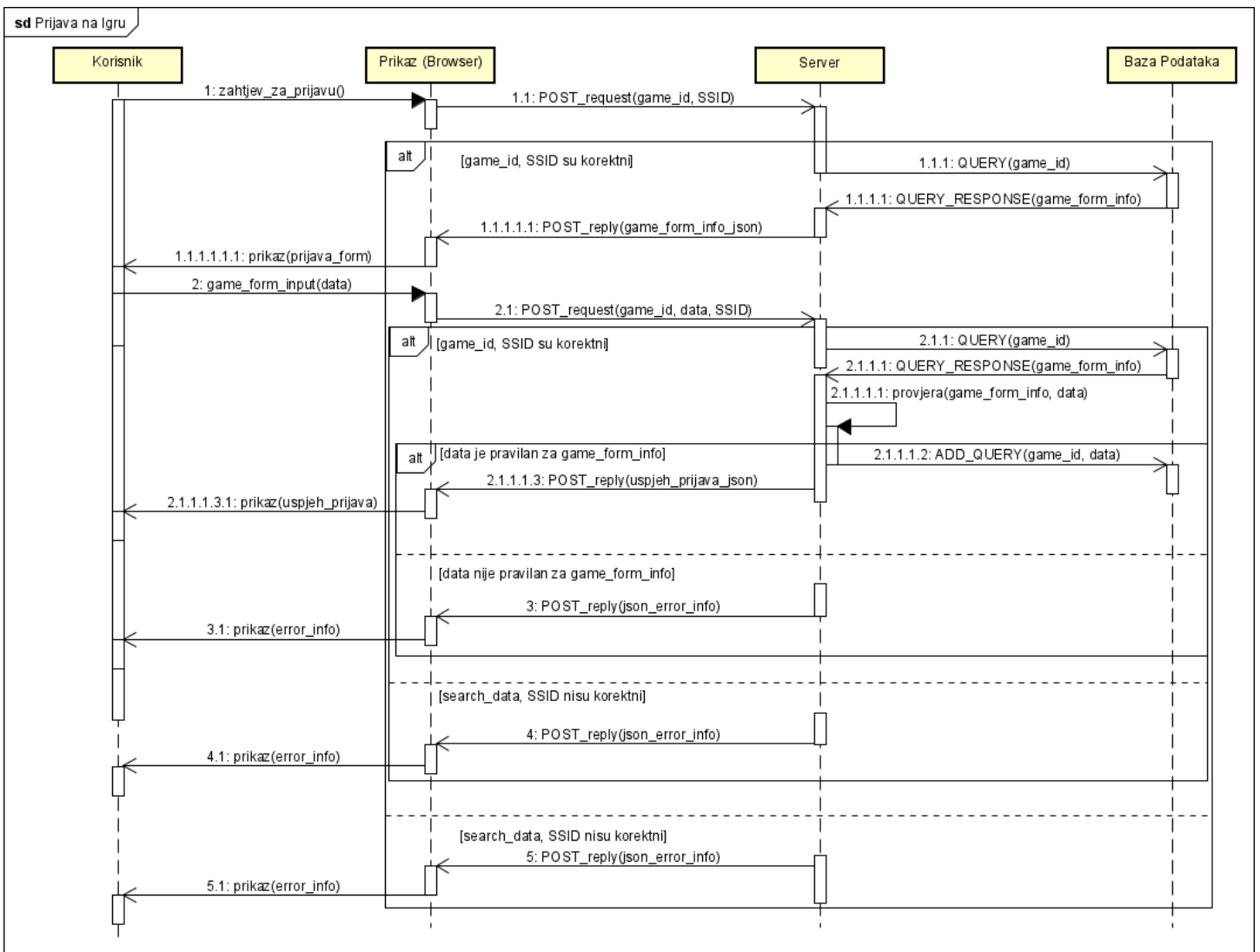


## Sekvencijski dijagrami

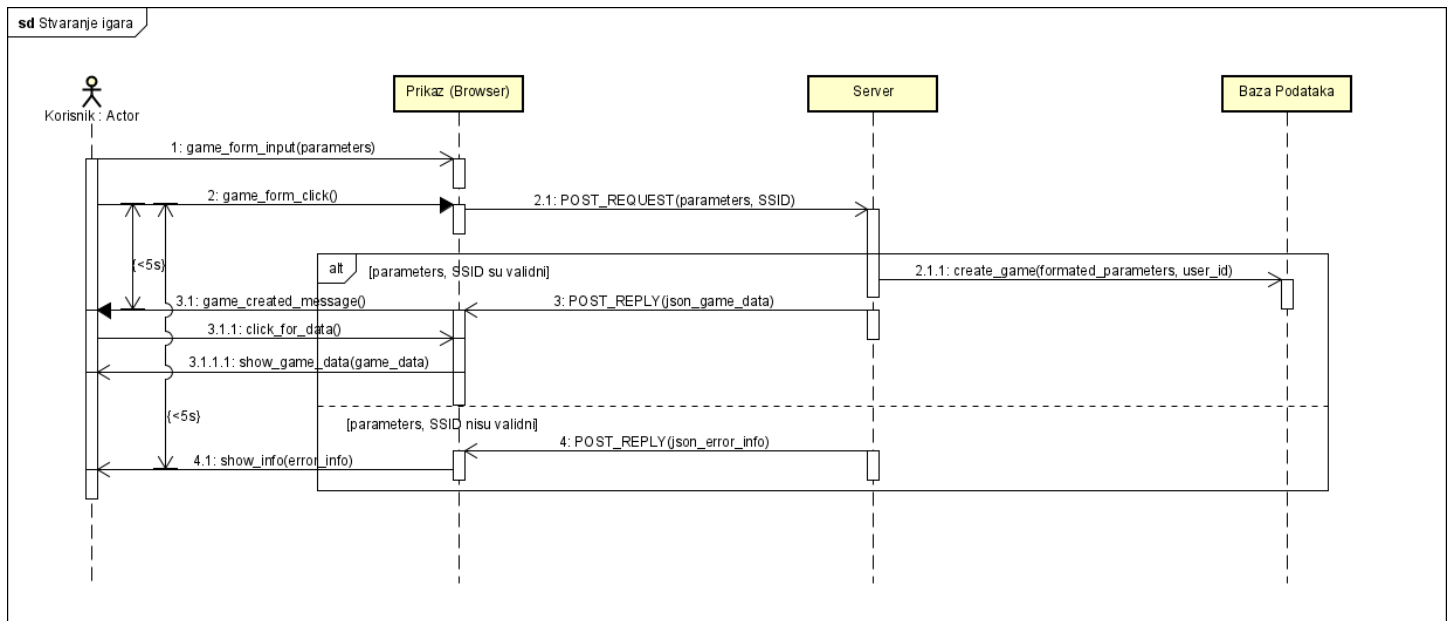
### Pretraživanje Igara



## Prijava na Igru



# Stvaranje Igre



# Nefunkcijski zahtjevi

## 1. Performanse:

- Sustav mora omogućiti brzo pretraživanje igara (manje od 5 sekundi), uključujući filtriranje po različitim kriterijima.
- Aplikacija mora biti responzivna i omogućiti optimalno iskustvo na različitim veličinama ekrana.

## 2. Sigurnost:

- Korisnički podatci trebaju biti zaštićeni—neregistrirani korisnici ne smiju moći vidjeti privatne igre; ne bi trebalo biti moguće vidjeti točnu lokaciju lokalizirane igre ako prijava nije odobrena; etc.
- Prijavljeni korisnici trebaju imati siguran i zaštićen pristup svojim osobnim podacima i igrama.

## 3. Skalabilnost:

- Aplikacija mora podržavati velik broj istovremenih korisnika (više od 100) bez gubitka performansi.
- Treba omogućiti jednostavno širenje funkcionalnosti u budućnosti (modularni dizajn).

## 4. Održavanje:

- Aplikacija mora biti lako održiva, uz mogućnost redovitog ažuriranja sustava bez prekida u radu.
- Kôd treba biti dobro dokumentiran radi lakšeg održavanja i dodavanja novih značajki.

# Opis arhitekture

---

## Stil arhitekture: Klijent-poslužitelj

---

Odabir ove arhitekture temeljen je na razdvajanju odgovornosti gdje klijent (React) rukuje s prezentacijom dok poslužitelj (Spring Boot) upravlja poslovnom logikom i bilo kakvim pristupom podacima (bazi podataka). Ovim smo postupkom raspodijelili tim na više manjih koji rade na slabopovezanim problematikama sustava.

### Klijent: React

React koristi CBA (component-based architecture), ali i SPA (single page application) koji korisniku omogućava puno bolji sustav interakcije bez konstantnog osvježavanja stranice.

### Poslužitelj: Spring Boot, PostgreSQL

U slučaju TTRPG Finder stranice, poslužitelj napravljen u Spring Bootu koristi MVC arhitekturu s bitnom razlikom u tome da je **pregled** zamijenjen JSON odgovorima koji se šalju klijentu. Na serveru se ne generiraju nikakve view-komponente, već se to održava u Reactu na klijentskoj strani.

## Detaljan Opis slojeva

---

### Frontend sloj - klijent (React)

**Uloga:** Prezentacija sadržaja i korisnička interakcija.

**Tehnologije:**

- **React** (razvoj korisničkog sučelja)
- **React Router** (navigacija unutar SPA sustava)
- **Axios** (slanje HTTP zahtjeva)
- **CSS** (stiliziranje)

### Backend sloj - poslužitelj (Spring Boot)

## Prezentacijski sloj (Controller Layer)

**Uloga:** Obrada HTTP zahtjeva, primanje podataka od klijenta i slanje podataka klijentima.

**Tehnologije:**

- **Spring Web** (rukovanje HTTP zahtjevima)
- **Spring Security** (autentifikacija i autorizacija)
- **JSON** (serijalizacija podataka)

## Servisni sloj (Business Logic Layer)

**Uloga:** Obavlja poslovne operacije, validira i obrađuje unos, poziva repozitориjski sloj za dohvat ili pohranu podataka.

**Tehnologije:**

- **Spring Boot**

## Repozitorijski sloj (Repository Layer)

**Uloga:** Komunicira s bazom podataka.

**Tehnologije:**

- **Spring Data JPA**

## Baza podataka - nastavak na poslužitelja (PostgreSQL)

**Uloga:** Upravljanje podacima i poslovnim pravilima

**Tehnologije:**

- **PostgreSQL** (pohrana i upravljanje podacima)

## Opis komunikacije između slojeva

---

### Klijent (React) - Poslužitelj (Spring Boot)

**Komunikacija:** Klijent (React) šalje HTTP zahtjeve poslužitelju (Spring Boot) kako bi dobio podatke koje korisnik traži ili pohranio informacije koje korisnik unosi.

**HTTP Zahtjevi:** React aplikacija koristi Axios ili Fetch za slanje zahtjeva prema backendu. Ovi zahtjevi mogu biti GET, POST, PUT, ili DELETE, ovisno o operaciji koju klijent želi izvršiti (npr. dohvat podataka,



kreiranje novog entiteta, ažuriranje postojećeg, ili brisanje podataka).

Autentifikacija i autorizacija: Prije nego što poslužitelj odgovori na zahtjev, provodi se autentifikacija i autorizacija korisnika putem Spring Security. Ako korisnik nije autentificiran ili nema odgovarajuće privilegije, poslužitelj vraća odgovarajući statusni kod (npr. 401 Unauthorized ili 403 Forbidden).

JSON odgovori: Nakon što su zahtjevi prošli validaciju, poslužitelj odgovara s JSON podacima koji predstavljaju rezultate operacija (npr. korisničke informacije, status poruka, lista objekata). Ovi podaci se zatim prikazuju korisniku putem React aplikacije.

## Backend - Baza podataka

Backend dohvaća i manipulira podacima u bazi podataka kako bi imao informaciju za vratiti klijentu ili sprema informacije koje mu je klijent poslao.

### Poslužitelj (Spring Boot) - Baza podataka (PostgreSQL)

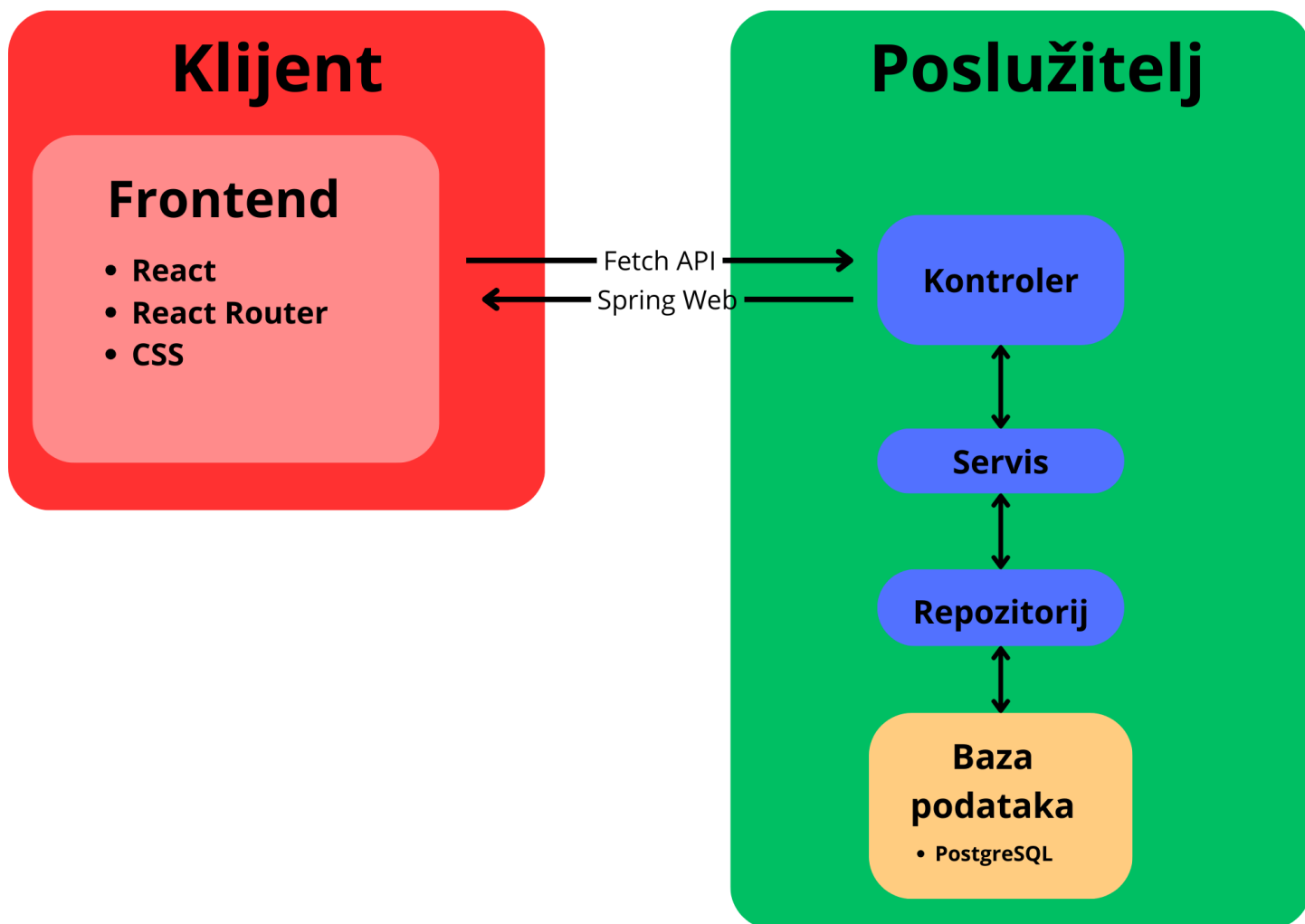
Komunikacija: Poslužitelj (Spring Boot) koristi Repozitorski sloj za komunikaciju s bazom podataka (PostgreSQL) kako bi dohvaćao, manipulirao ili pohranjivao podatke.

Repozitorski sloj: Kada poslužitelj treba pristupiti podacima, koristi Spring Data JPA kako bi komunicirao s bazom podataka. Ovaj sloj omogućuje obavljanje CRUD (Create, Read, Update, Delete) operacija bez potrebe za pisanjem SQL upita, jer Spring Data JPA generira potrebne SQL upite na temelju definicija u repozitoriju.

Poslovna logika (Servisni sloj): Prije nego što poslužitelj pristupi bazi podataka, servisni sloj obavlja poslovnu logiku. To uključuje provjeru valjanosti podataka, unos poslovnih pravila i obrada podataka.

Baza podataka (PostgreSQL): Poslužitelj poziva repozitorij da izvrši operacije poput dohvaćanja podataka, spremanja novih podataka ili ažuriranja postojećih. PostgreSQL pohranjuje ove podatke, upravlja vezama među entitetima i osigurava integritet podataka.

Transakcije: Ako se radi o operacijama koje uključuju više promjena podataka, Spring će osigurati da se transakcije izvrše u cijelosti ili se ponište u slučaju greške (kroz @Transactional).



## Baza podataka

Baza podataka je implementirana preko PostgreSQL-a, a baza je podijeljena na normalizirane tablice. Za učinkovito traženje informacija za svaku tablicu je stvoren indeks za primarni ključ, a u slučaju primarnog ključa od više atributa atributi u indeksu idu od najopćenitijeg do najspecifičnijeg. Uz to su kreirani indeksi za jedinstvene karakteristike korisničkog imena i imena organizacijskog profila, u slučaju da se pretraga vrši preko tih jedinstvenih atributa. Tablice sistema i pravilnika se popunjavaju preko superusera baze podataka, dok se ostale tablice dinamično pune od backend strane programa. To se događa nakon što korisnik kreira i uređuje profil, igru, prijavu za igre itd.

## Opis tablica

### Korisnik

Atribut	Tip podatka	Opis varijable
---------	-------------	----------------

Atribut	Tip podatka	Opis varijable
user_id	INT	Jedinstveni identifikator korisnika(Primarni ključ)
username	Varchar(20)	Jedinstveni username
user_password	Varchar(20)	Lozinka

## Poslovni korisnik

Atribut	Tip podatka	Opis varijable
user_id	INT	Jedinstveni identifikator korisnika(Primarni, Strani ključ)
company_id	INT	Strani ključ

## Privatni korisnik

Atribut	Tip podatka	Opis varijable
user_id	INT	Jedinstveni identifikator korisnika(Primarni, strani ključ)

## Org profil

Atribut	Tip podatka	Opis varijable
company_id	INT	Jedinstveni identifikator kompanije(Primarni ključ)
company_name	Varchar(50)	Jedinstveno ime kompanije
company_des	Varchar(400)	Opcionalan opis kompanije
company_web	Varchar(100)	Opcionalna web stranica kompanije
company_address	Varchar(200)	Opcionalna adresa kompanije
company_phone	Varchar(20)	Opcionalan broj telefona kompanije
user_id	INT	Identifikator zaposlenika (poslovnog korisnika)(Strani ključ)
company_logo	Varchar(200)	Lokacija logoa kompanije

## Prijava

Atribut	Tip	Opis varijable
---------	-----	----------------

	podatka	
prijava_game_id	INT	Jedinstveni identifikator igre za koju se korisnik prijavljuje(Primarni, Strani ključ)
prijava_user_id	INT	Jedinstveni identifikator prijavljenog igrača(Primarni, strani ključ)
status	Varchar(20)	Status prijave

## Pitanje

Atribut	Tip podatka	Opis varijable
pitanje_game_id	INT	Jedinstveni identifikator igre za koju se postavlja pitanje(Primarni, Strani ključ)
questionText	Varchar(500)	Jedinstveni tekst pitanja(Primarni ključ)

## Odgovor

Atribut	Tip podatka	Opis varijable
answerText	Varchar(500)	Jedinstveni tekst odgovora(Primarni ključ)
pitanjeGameId	INT	Jedinstveni identifikator igre od pitanja(Primarni, strani ključ)
questionText	Varchar(500)	Jedinstveni tekst pitanja(Primarni, strani ključ)
prijavaGameId	INT	Jedinstveni identifikator igre od prijave(Primarni, strani ključ)
prijavaUserId	INT	Jedinstveni identifikator korisnika koji se prijavljuje(Primarni, strani ključ)

## Igra

Atribut	Tip podatka	Opis varijable
game_id	INT	Jedinstveni identifikator igre(Primarni ključ)
game_name	Varchar(50)	Ime igre
description	Varchar(400)	Opcionalni opis igre
max_player	INT	Opcionalni maksimalni broj igrača

Atribut	Tip podatka	Opis varijable
is_private	bool	Opisuje je li igra privatna ili ne
is_homebrew	bool	Opisuje je li igra "homebrew" ili ne
requires_form	bool	Opisuje je li potrebno ispuniti dokument za prijavu
start_ts	Timestamp	Opcionalni timestamp za početak kampanje ako još nije započela
est_length	Varchar(100)	Procjena za duljinu kampanje
comm_channel	Varchar(100)	Komunikacijski kanali za kampanju
ruleset	Varchar(50)	Oznaka pravilnika
gm_user_id	INT	Jedinstvena oznaka igrača voditelja(Strani ključ)

### Točno-lokacijska igra

Atribut	Tip podatka	Opis varijable
game_id	INT	Jedinstveni identifikator igre(Primarni, strani ključ)
exactLat	FLOAT	geografska širina
exactLng	FLOAT	geografska dužina

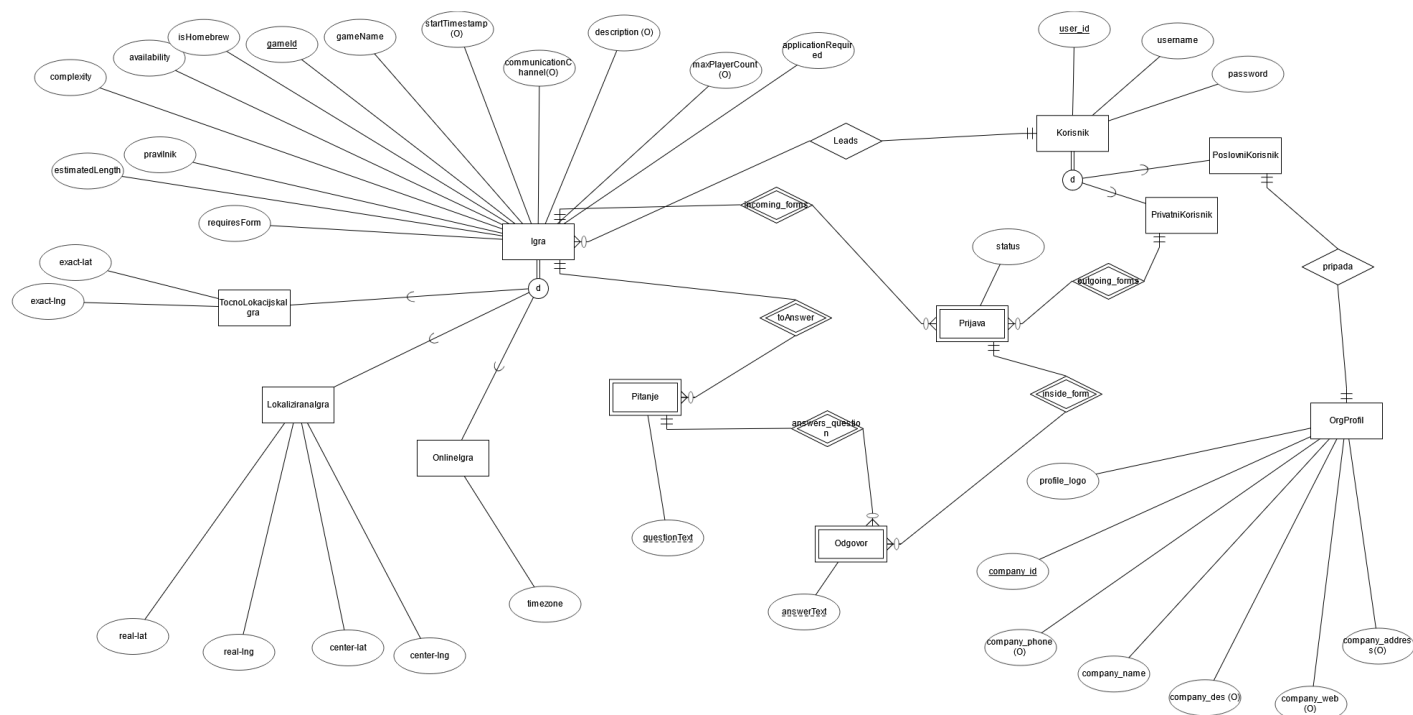
### Lokalizirana igra

Atribut	Tip podatka	Opis varijable
game_id	INT	Jedinstveni identifikator igre(Primarni, strani ključ)
realLat	FLOAT	geografska širina
realLng	FLOAT	geografska dužina
fakeLat	FLOAT	generirana prikrivena geografska širina
fakeLng	FLOAT	generirana prikrivena geografska dužina

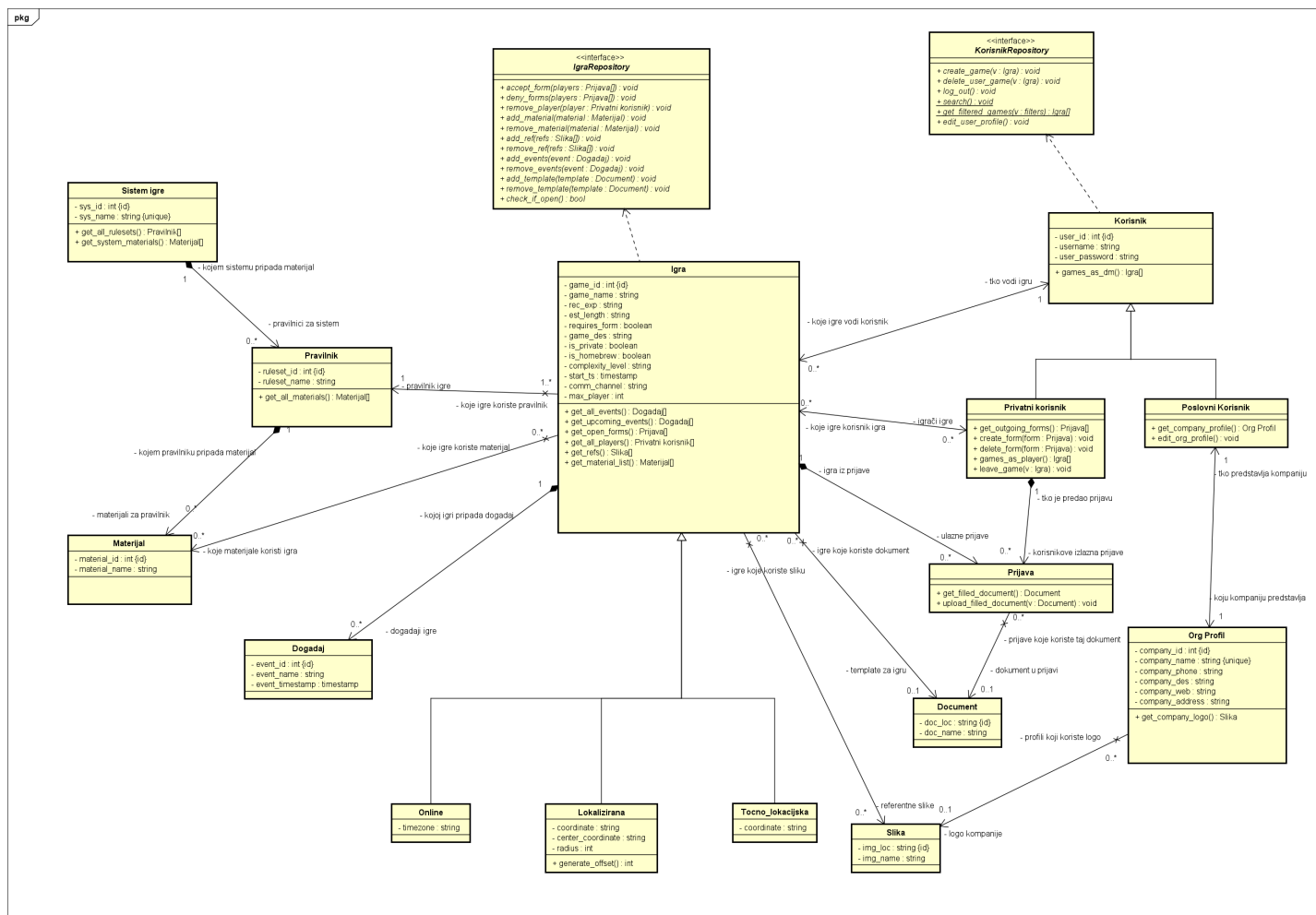
### Online igra

Atribut	Tip podatka	Opis varijable
---------	-------------	----------------

Atribut	Tip podatka	Opis varijable
game_id	INT	Jedinstveni identifikator igre(Primarni ključ)
Timezone	Varchar(20)	Vremenska zona



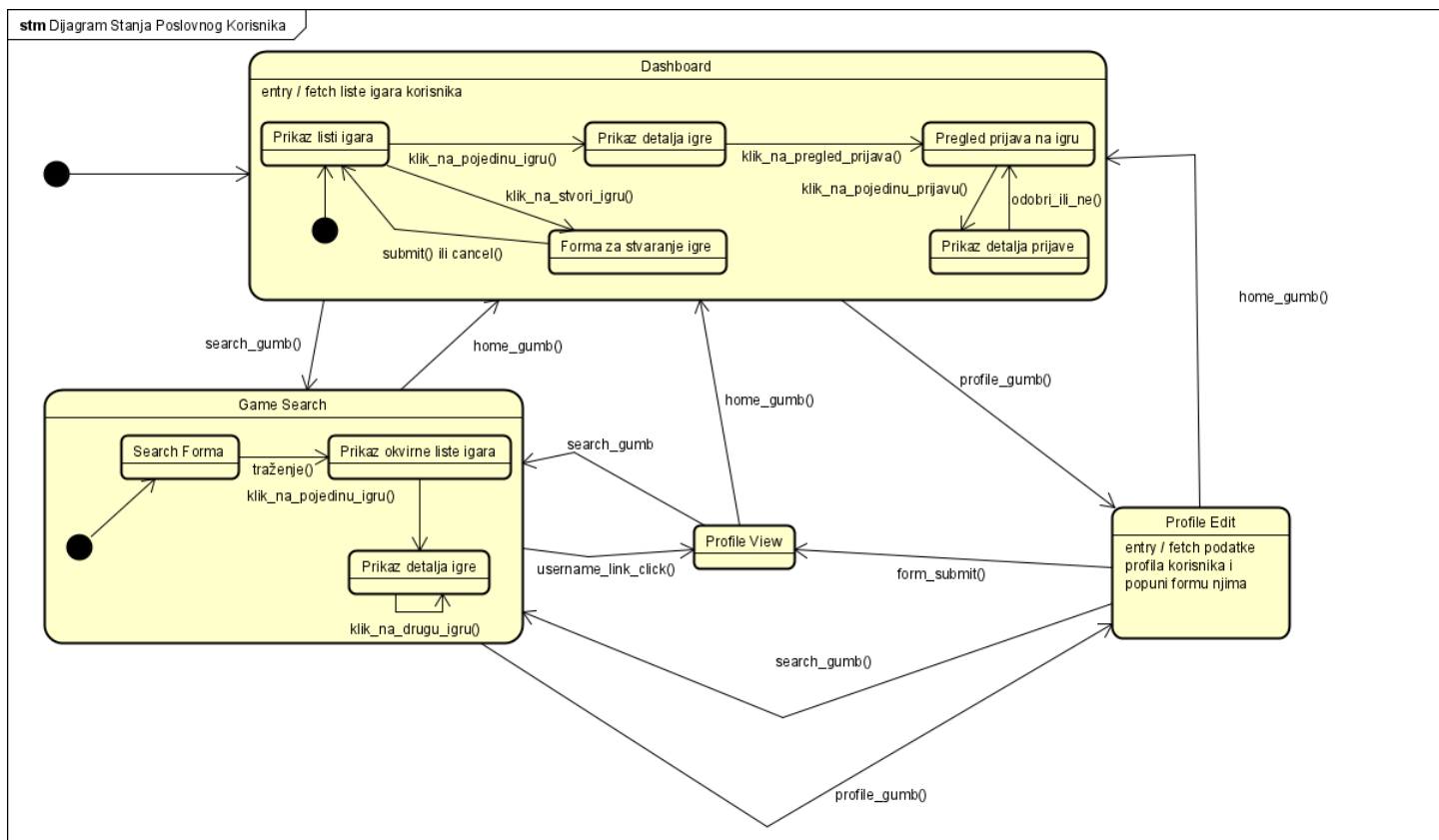
## Dijagram razreda



*dio 2. revizije*

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

## Dijagram stanja

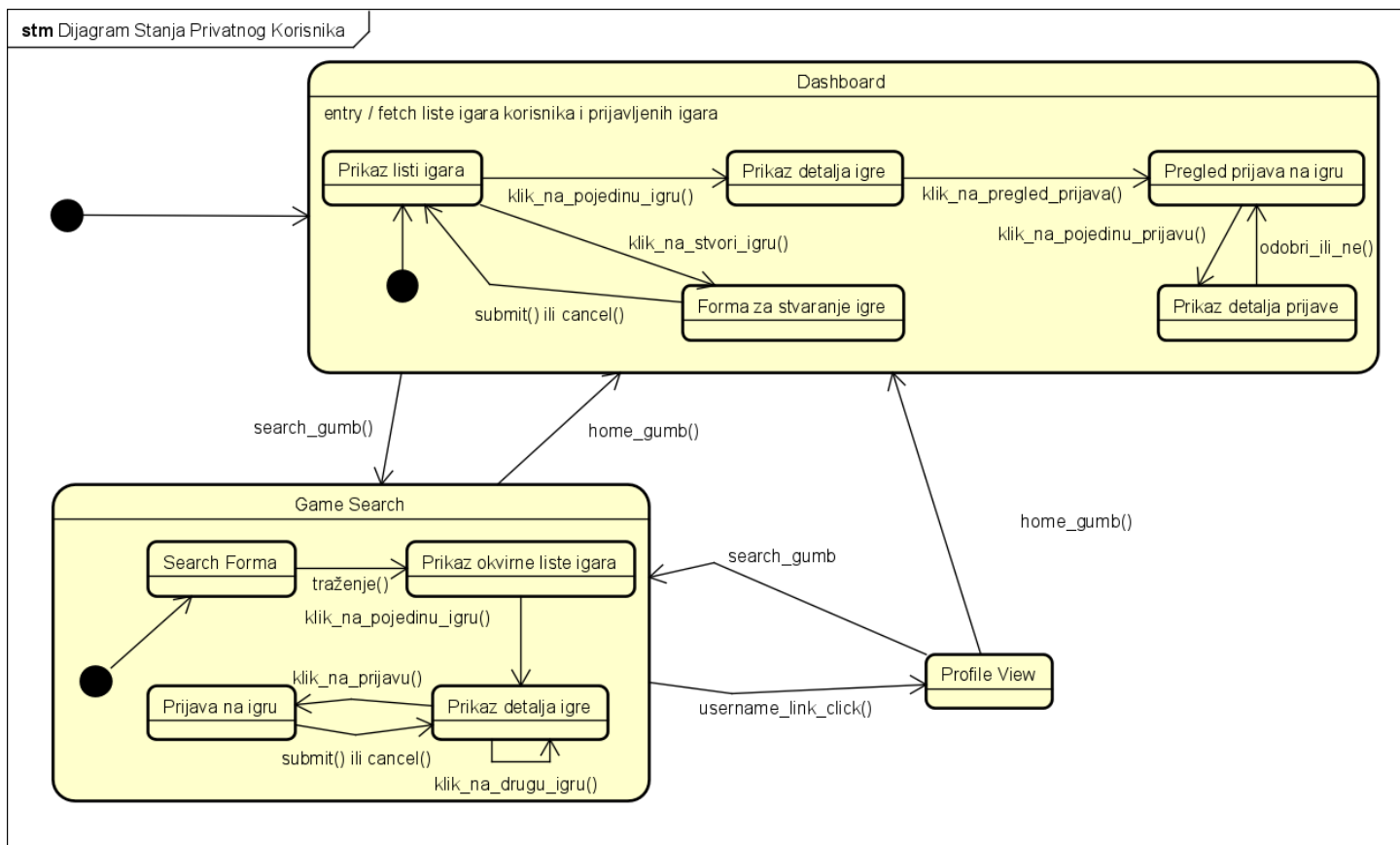


Poslovni korisnik ima "Dashboard" stranicu koja je reprezentirana kao stanje. Unutar tog stanja, nalaze se podstanja prikaza onoga što se pokazuje na "Dashboard" stranici. Lista igri automatski je prikazana, a stvaranje igre i prikaz detalja igre pokazuju se kada korisnik klikne na određene dijelove. Korisnik na "Dashboard" stanju/stranici može pregledati prijave na igre, i prikazati detalje prijave.

Korisnik može pretraživati igre kada se nalazi u stanju "Game Search", te to radi manevriranjem po unutarnjim stanjima stanja "Game Search".

Korisnik može pregledati tuđi profil ili urediti svoj u stanjima "Profile View" i "Profile Edit"

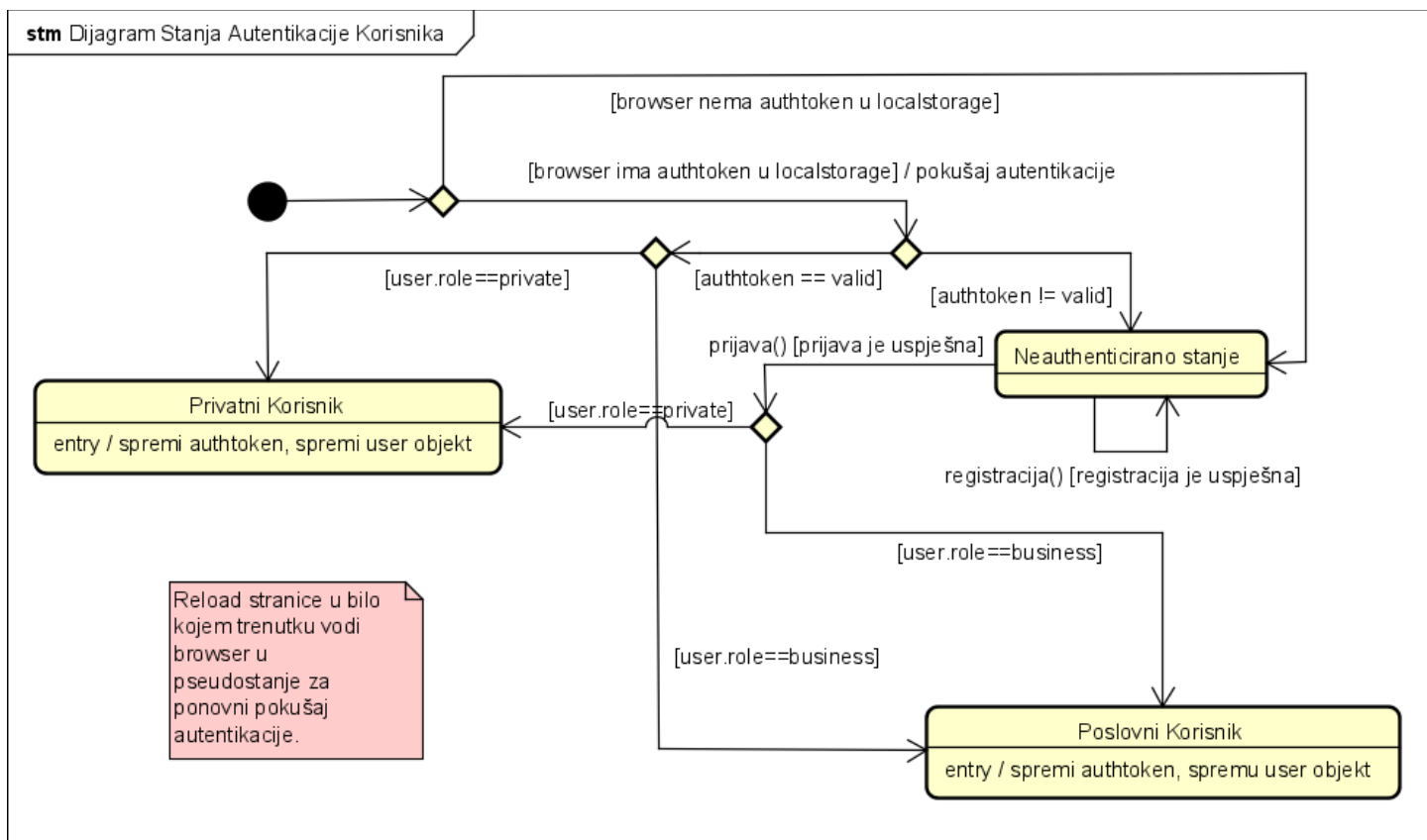




Privatni korisnik ima "Dashboard" stranicu koja je reprezentirana kao stanje. Unutar tog stanja, nalaze se podstanja prikaza onoga što se pokazuje na "Dashboard" stranici. Lista igri automatski je prikazana, a stvaranje igre i prikaz detalja igre pokazuju se kada korisnik klikne na određene dijelove. Korisnik na "Dashboard" stanju/stranici može pregledati prijave na igre, i prikazati detalje prijave.

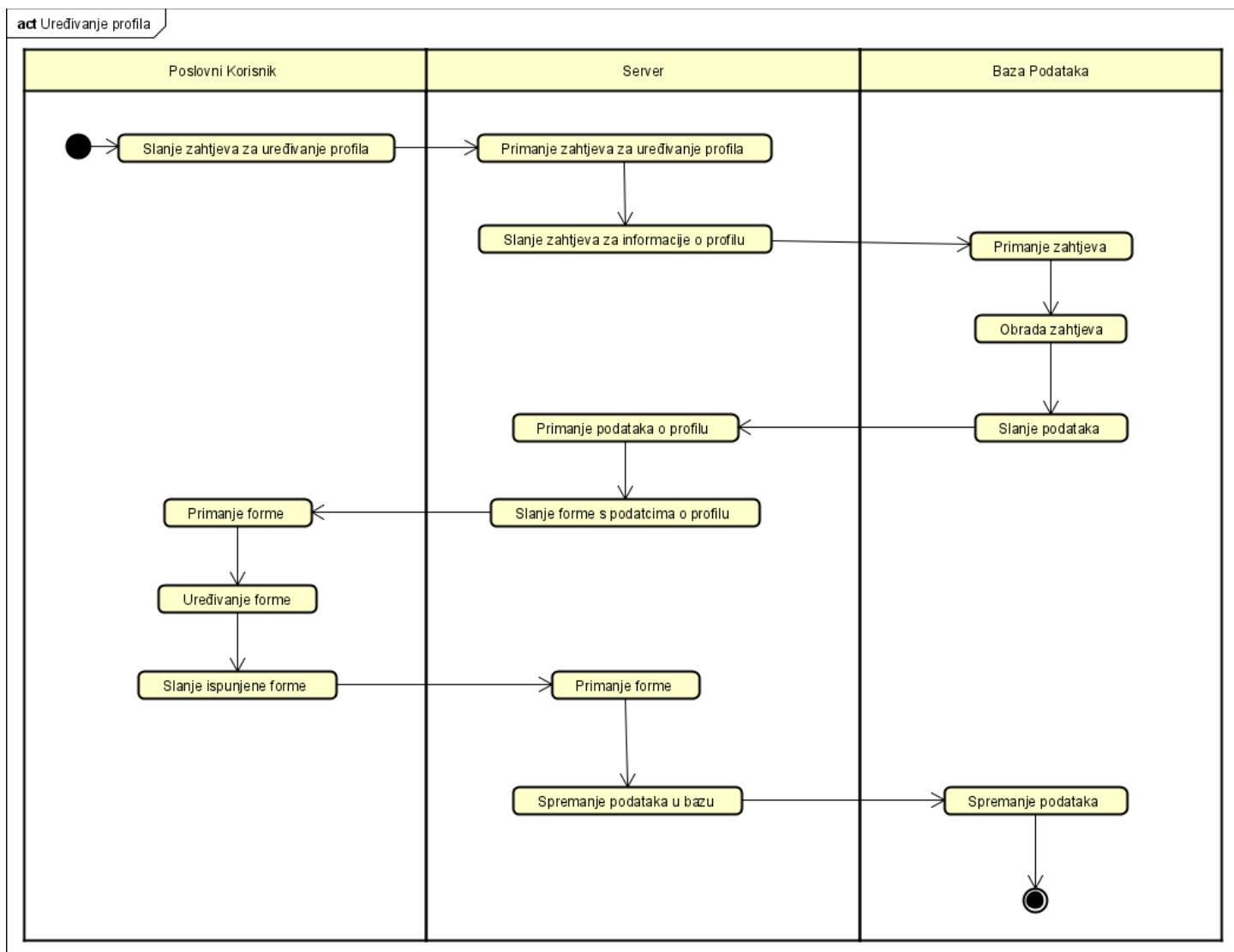
Korisnik može pretraživati igre kada se nalazi u stanju "Game Search", te to radi manevriranjem po unutarnjim stanjima stanja "Game Search". Za razliku od poslovnog korisnika, ovaj korisnik se na ovom stanju može i prijaviti na igru.

Korisnik može pregledati tuđi profil ili urediti svoj u stanjima "Profile View", ali nema stanje za uređivanje vlastitog profila.



Autentikacija korisnika radi se preko prijave ili registracije na stranicu, ali pri učitavanju stranice pregledava se ima li browser u sebi authtoken kojim se na serveru može odrediti koji korisnik trenutno radi na browseru.

## Dijagram aktivnosti



Opis: Ovaj dijagram prikazuje proces uređivanja profila poslovnog korisnika kroz interakciju sa serverom i bazom podataka. Proces uključuje tri glavna aktera: **Poslovni korisnik**, **Server**, i **Baza podataka**.

## Koraci:

### Poslovni korisnik

1. Započinje proces slanjem zahtjeva za uređivanje profila.
2. Prima formu za uređivanje.
3. Nakon uređivanja forme, šalje popunjenu formu nazad.

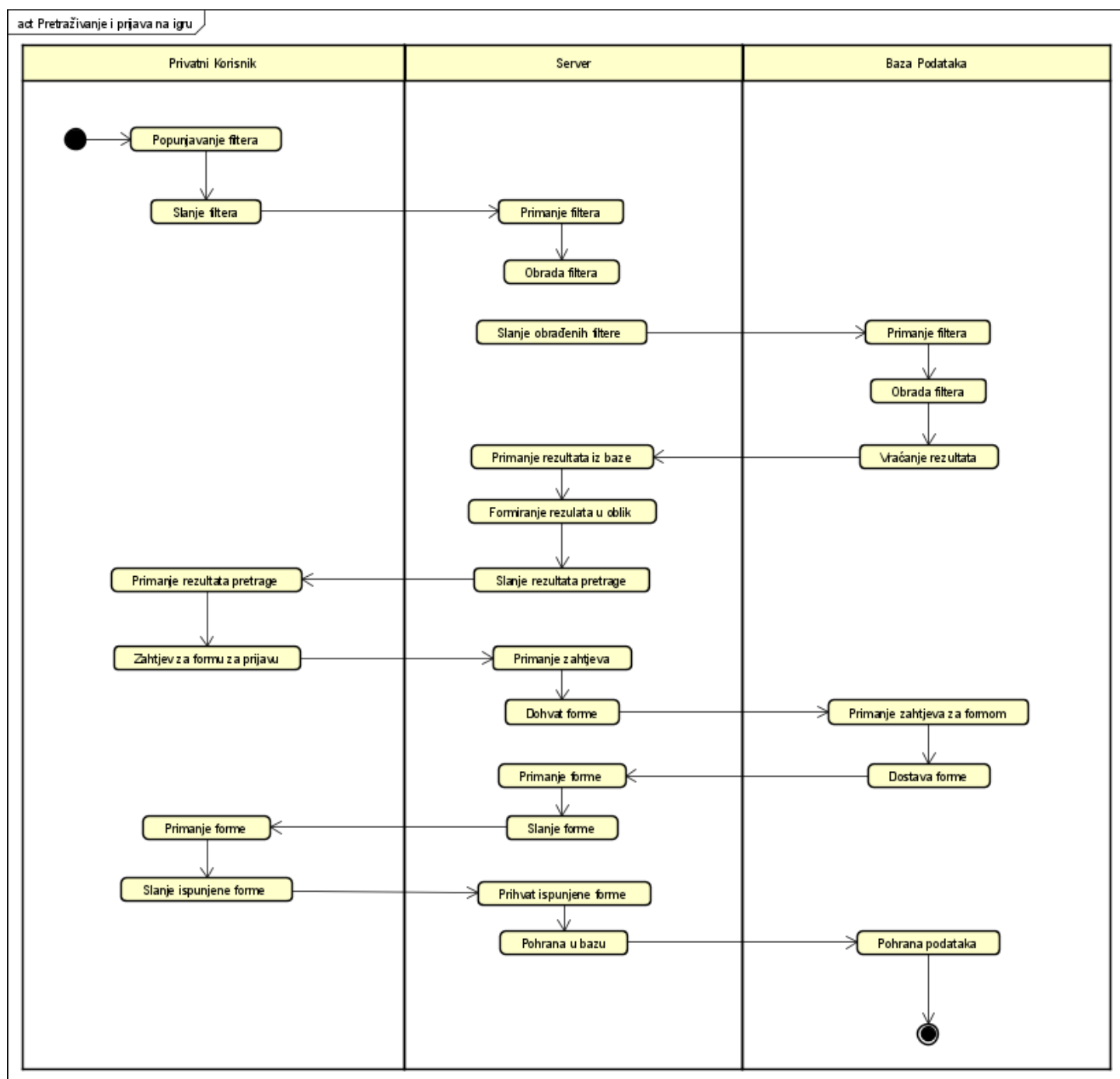
### Server

1. Prima zahtjev za uređivanje profila od korisnika.

2. Prosljeđuje zahtjev za informacijama o profilu bazi podataka.
3. Prima podatke o profilu iz baze i šalje ih korisniku kao formu.
4. Prima popunjenu formu od korisnika i prosljeđuje je bazi podataka.

## Baza podataka

1. Prima zahtjev od servera za informacijama o profilu, obrađuje ga i šalje podatke nazad.
2. Prima popunjenu formu i sprema ažurirane podatke u bazu.



Opis: Ovaj dijagram prikazuje proces pretraživanja igara i prijave privatnog korisnika na određenu

igru. Proces uključuje tri glavna aktera: **Privatni korisnik**, **Server**, i **Baza podataka**.

## Koraci:

---

### Privatni korisnik

1. Započinje proces popunjavanjem filtera za pretraživanje igara.
2. Prima rezultate pretrage.
3. Podnosi zahtjev za prijavu na igru.
4. Popunjava formu i šalje je.

### Server

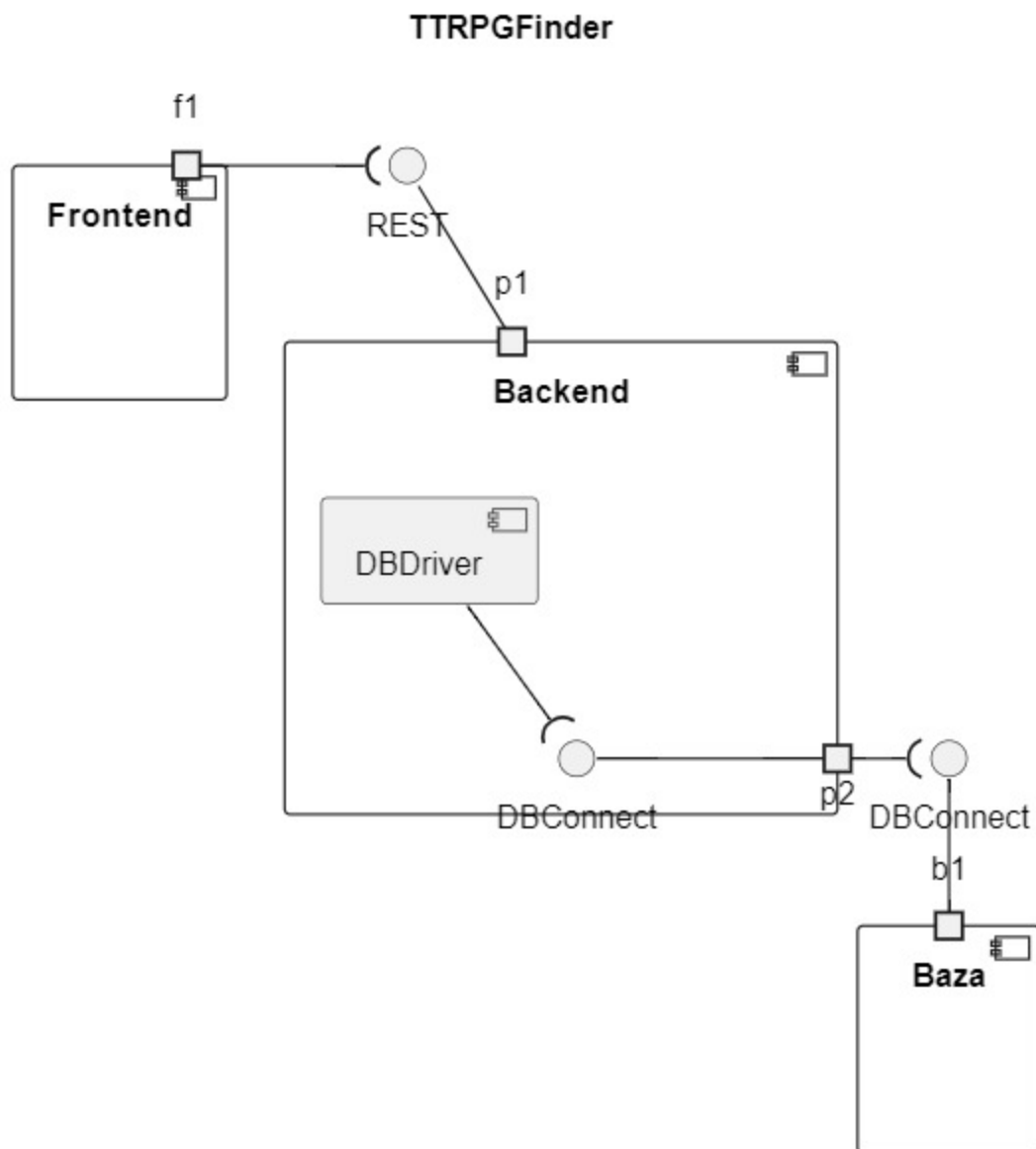
1. Prima filtere od korisnika, obrađuje ih i prosljeđuje bazi podataka.
2. Prima rezultate pretrage iz baze, formatira ih i šalje korisniku.
3. Na zahtjev za prijavu, preuzima formu iz baze i šalje je korisniku.
4. Nakon što korisnik popuni formu, prihvata podatke i prosljeđuje ih bazi.

### Baza podataka

1. Prima i obrađuje filtere za pretragu, a zatim vraća rezultate serveru.
2. Na zahtjev za prijavu, preuzima odgovarajuću formu i šalje je serveru.
3. Prima popunjenu formu od servera i sprema podatke u bazu.

## Dijagram komponenti

---



Sustav TTRPGFinder sastoji se od tri glavne komponente: Frontenda, Backenda i baze podataka, koje zajedno omogućuju rad sustava. Frontend predstavlja dio sustava koji je vidljiv korisnicima i s kojim oni izravno komuniciraju. Služi kao korisničko sučelje te šalje zahtjeve prema Backendu putem REST API-ja. Ova komunikacija odvija se kroz priključak označen kao f1.

Backend čini središnji dio sustava, odgovoran za obradu zahtjeva koje prima od Frontenda te za upravljanje podacima. Komunicira s Frontendom putem priključka p1, dok je za povezivanje s bazom podataka zadužen priključak p2. Unutar Backenda nalazi se podkomponenta zvana DBDriver, koja omogućuje povezivanje i interakciju s bazom podataka. DBDriver olakšava Backend komponenti čitanje, zapisivanje i upravljanje podacima.

Baza podataka predstavlja skladište u kojem se čuvaju svi podaci sustava. Povezana je s Backend komponentom putem konektora označenog kao DBConnect. Na dijagramu je prikazana s priključkom b1, koji omogućuje ovu vezu.

Frontend je odgovoran za interakciju s korisnicima, Backend obrađuje podatke i provodi poslovnu logiku, dok baza podataka osigurava sigurno i organizirano pohranjivanje podataka

# Korištene tehnologije i alati

---

## Frontend

---

### 1. Visual Studio Code (VSCode)

VSCode je korišten za pisanje koda, uređivanje dokumentacije i testiranje rješenja. Kao lagani i snažni editor, VSCode je omogućio efikasno kodiranje i integraciju s različitim alatima i ekstenzijama.

[Više o VSCode-u](#)

### 2. GitHub Desktop

GitHub Desktop korišten je za verzioniranje koda i sinkronizaciju timskog rada na projektu. Omogućio je lakše upravljanje repozitorijima i praćenje promjena u kodu tijekom razvoja.

[Više o GitHub Desktopu](#)

### 3. React + Vite (v18)

**React** je korišten za razvoj korisničkog sučelja (frontend), omogućujući brzo i dinamično renderiranje aplikacije. **Vite** je korišten kao alat za brzo postavljanje razvojnih okruženja, što je omogućilo brzu iteraciju i minimalno vrijeme čekanja pri pokretanju aplikacije.

[Više o Reactu](#)

[Više o Vite-u](#)

### 4. Node.js (v8.17.0)

Node.js je korišten za razvoj mock backenda pomoću **Express.js**, omogućujući jednostavno postavljanje i rukovanje serverom za potrebe testiranja aplikacije.

[Više o Node.js](#)

### 5. npm (Node Package Manager, v11)

npm je korišten za upravljanje paketima i ovisnostima u projektu. Omogućio je jednostavno instaliranje potrebnih biblioteka i alata za razvoj aplikacije.

[Više o npm-u](#)

### 6. Leaflet API

Leaflet je korišten za implementaciju interaktivne mape na search stranici aplikacije.

[Više o Leaflet API-u](#)

### 7. ChatGPT

ChatGPT je korišten za pomoć u razvoju i implementaciji određenih funkcionalnosti. Korišten je za analiziranje koda i brže učenje implementacije određenih aspekata potrebnih za stvaranje aplikacije.

[Više o ChatGPT-u](#)

## Dokumentacija

---

### 1. Astah UML

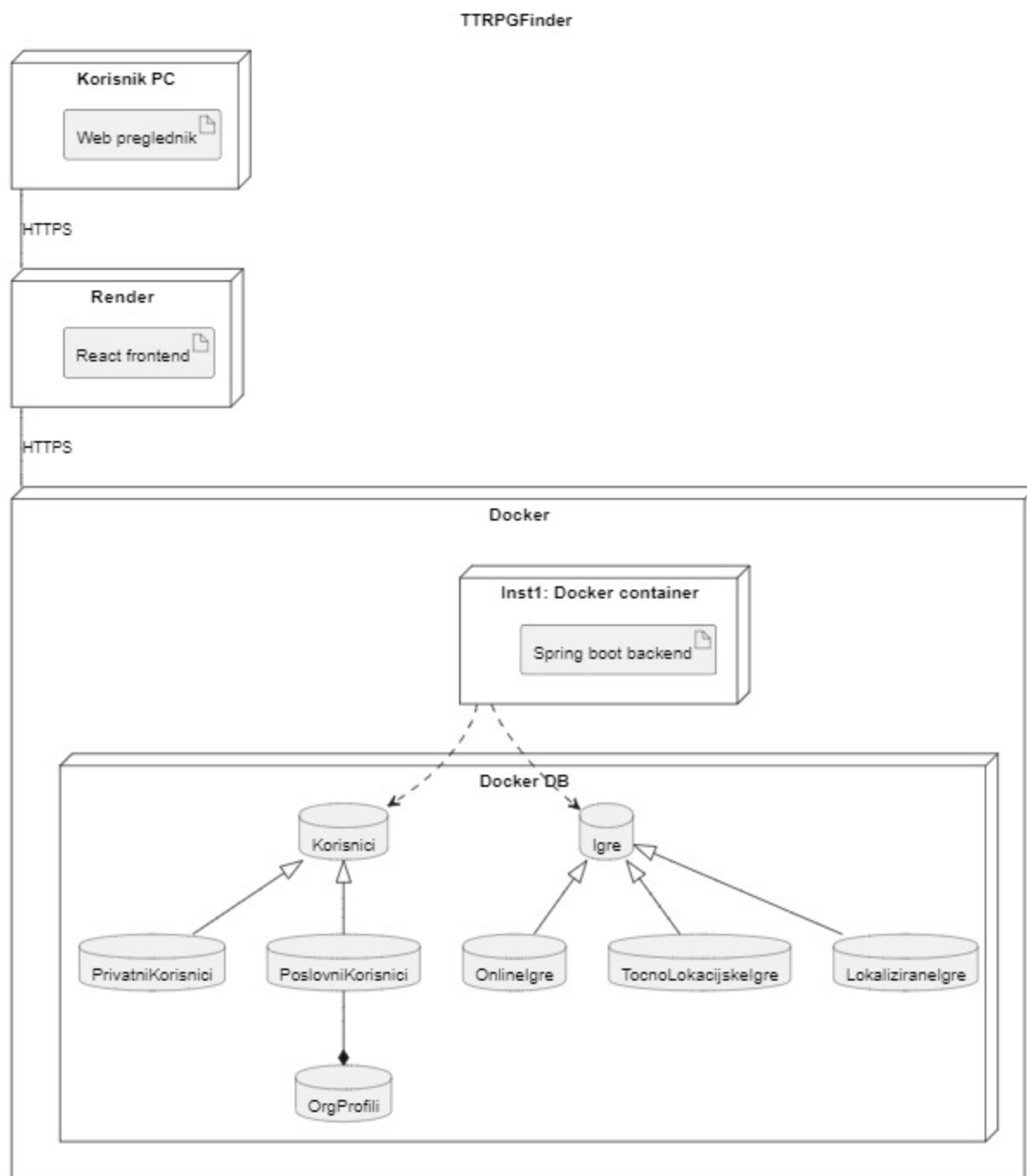
Astah UML je alat za modeliranje i stvaranje UML dijagrama. Koristili smo ga za izradu različitih dijagrama, uključujući klasne dijagrame, sekvencijske dijagrame, dijagrame aktivnosti, dijagrame stanja, dijagrame use case-ova i razreda. Ovi dijagrami pomogli su u vizualizaciji strukture i ponašanja sustava, što je olakšalo razumijevanje i organizaciju projekta.

[Više o Astah UML-u](#)

## Dijagram razmještaja

---





Dijagram

razmještaja sustava TTRPGFinder prikazuje arhitekturu koja se sastoji od korisničkog sloja, frontenda, backenda i baze podataka, pri čemu su backend i baza smješteni unutar Docker okruženja.

Korisnici sustavu pristupaju putem svojih uređaja koristeći web preglednik, a komunikacija se odvija preko HTTPS-a kako bi se osigurala sigurnost podataka. Frontend aplikacija, razvijena u Reactu, smještena je na platformi Render i služi kao posrednik između korisnika i backend sustava. Korisnici šalju zahtjeve frontendu, koji ih zatim prosljeđuje backend komponenti.

Backend je implementiran kao Spring Boot aplikacija i pokreće se unutar Docker okruženja. Njegova je uloga obrada korisničkih zahtjeva i upravljanje poslovnom logikom. Backend komunicira s bazom podataka, također smještenom u Docker okruženju, za dohvaćanje i pohranu podataka. Baza

podataka strukturirana je u nekoliko entiteta, uključujući korisnike i igre. Podaci o korisnicima obuhvaćaju privatne i poslovne korisnike te organizacijske profile, dok podaci o igrama uključuju online igre, igre s preciznom lokacijom i lokalizirane igre.

Korisnički uređaji komuniciraju s frontendom, frontend s backendom, a backend s bazom podataka, čime se osigurava modularnost, sigurnost i jednostavnost održavanja sustava.

Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

## Ispitivanje komponenti

---

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u razredima sustava. Ovdje je potrebno izolirati svaku komponentu kako bi se testirala njezina ispravnost i reakcija na različite scenarije.

### Zadaci:

1. Razviti minimalno **6 ispitnih slučajeva** koji obuhvaćaju:
  - **Redovne slučajeve:** testiranje uobičajenog ponašanja funkcionalnosti.
  - **Rubne uvjete:** provjera ulaznih podataka na granici valjanosti.
  - **Izazivanje pogreške (exception throwing):** testiranje reakcije na iznimke.
  - **Nepostojeće funkcionalnosti:** provjera reakcije na poziv neimplementirane funkcionalnosti.

### Struktura ispitivanja:

Za svaki ispitni slučaj potrebno je:

1. Opišite funkcionalnost koju testirate (npr. dodavanje korisnika, validacija podataka).
2. Navedite ispitni slučaj:
  - Ulazne podatke.
  - Očekivane rezultate.
  - Dobivene rezultate (prolaz/pad ispitivanja).
3. Opišite postupak provođenja ispitivanja

4. U Gitu moraju biti dostupni izvorni kodovi ispitnih slučajeva.

## Ispitivanje sustava

---

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja, uz posebnu pažnju na međusobnu povezanost svih komponenti. Ispitivanje treba obuhvatiti sve aspekte sustava i njegovu interakciju s korisnicima.

### Zadaci:

Razviti minimalno **4 ispitna slučaja** koji obuhvaćaju:

- **Redovne slučajeve:** očekivano ponašanje sustava.
- **Rubne uvjete:** reakcija sustava na granične ulaze.
- **Poziv nepostojećih funkcionalnosti:** testiranje kako sustav reagira na neimplementirane ili neispravne funkcije.

### Struktura ispitnih slučajeva za Selenium:

#### 1. Ulazi:

- Konkretni podaci koji se unose u sustav (npr. korisničko ime i lozinka za prijavu).
- Simulacija korisničkih akcija (klikanje, unos teksta, navigacija).

#### 2. Koraci ispitivanja:

- Npr. Detaljan opis koraka koje Selenium izvršava:
  - a. Otvoriti aplikaciju u pregledniku.
  - b. Unijeti podatke u formu.
  - c. Kliknuti na gumb za potvrdu.
  - d. Verificirati očekivane rezultate (npr. prijava uspješna).

#### 3. Očekivani izlaz:

- Očekivani rezultat ispitivanja (npr. korisnik preusmjeren na početnu stranicu nakon prijave).

#### 4. Dobiveni izlaz:

- Priložiti logove, screenshotove ili izvještaje s generiranim rezultatima (npr. iz JUnit-a ili Selenija).

## Alati za ispitivanje sustava:

- **Selenium IDE** ili prikladan obzirom na vaše razvojno okruženje:
  - Jednostavan alat za snimanje korisničkih akcija u pregledniku i automatsko ponavljanje testova.
  - Preporučuje se za osnovne ispitne slučajeve.
- **Selenium WebDriver:**
  - Omogućuje pisanje naprednih testova u različitim programskim jezicima (Java, Python, C#).
  - Preporučuje se za složenije testove, koji zahtijevaju detaljno prilagodbu i automatizaciju.

## Primjeri ispitivanja sa Seleniumom:

### 1. Formular za prijavu:

u dokumentaciji obavezna su specifičnija ispitivanja vaše aplikacije! - **Ulaz:** Korisničko ime = "user@fer.ugnz.hr", Lozinka = "password123".

- **\*\*Koraci\*\*:**

1. Otvoriti aplikaciju.
2. Unijeti korisničko ime i lozinku.
3. Kliknuti na "Prijava".
4. Provjeriti je li korisnik preusmjeren na početnu stranicu.

- **\*\*Očekivani izlaz\*\*:** "Prijava uspješna."

### 2. Rubni uvjet – nevažeća lozinka:

- **Ulaz:** Korisničko ime = "user@example.com", Lozinka = "malimedo".
- **Koraci:**
  - a. Unijeti podatke u formu za prijavu.

b. Kliknuti na "Prijava".

c. Provjeriti je li prikazana poruka o grešci.

- **Očekivani izlaz:** "Pogrešno korisničko ime ili lozinka."

## Prezentacija rezultata

Za oba tipa ispitivanja (komponenti i sustava) potrebno je:

- Jasno dokumentirati ulaze, korake, očekivane i dobivene rezultate.
- Priložiti slike ekrana, logove ili izvješća generirana alatima (npr. JUnit ili Selenium).
- Detaljno opisati ponašanje sustava, posebno u rubnim uvjetima.
- Navesti broj otkrivenih grešaka!

## TEST 1 - Registracija + Prijava za poslovnog korisnika

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Podaci za registraciju:
- Korisničko ime: ImeOrganizacijeZaTestiranje
- Lozinka: OrGaNiZACIJA123
- Email: privatniMail@email.mail
- Naziv organizacije: ogrNadSvima
- Podaci za prijavu:
- Korisničko ime: ImeOrganizacijeZaTestiranje
- Lozinka: OrGaNiZACIJA123

### 2. KORACI ISPITIVANJA:

- Otvaranje aplikacije:
  - Pokreće se Chrome preglednik i otvara URL aplikacije <http://localhost:5173/>.
- Navigacija na stranicu za registraciju:
  - Koristi se XPath za pronalazak navigacijskog izbornika.
  - Klik na gumb za registraciju.
- Unos podataka za registraciju:
  - Korisničko ime: unosi se ImeOrganizacijeZaTestiranje.

- Email: unosi se privatniMail@email.mail.
- Lozinka: unosi se OrGaNiZACIJA123.
- Ponovljena lozinka: unosi se OrGaNiZACIJA123.
- Klik na polje "Predstavlja organizaciju".
- Unos naziva organizacije: ogrNadSvima.
- Dovršetak registracije:
  - Klik na gumb za potvrdu registracije.
  - Prijava s novo registriranim podacima:
  - Unos korisničkog imena ImeOrganizacijeZaTestiranje.
  - Unos lozinke OrGaNiZACIJA123.
  - Klik na gumb za prijavu.
  - Provjera uspješnosti prijave:
  - Provjerava se je li korisnik preusmjeren na svoj profil.
  - Ako je tekst na gumbu za navigaciju "Profile", prijava je uspješna.

### 3. OČEKIVANI IZLAZ:

- Nakon registracije i prijave korisnik je preusmjeren na svoj dashboard
- Korisnik vidi "Profile" (označuje da je ulogiran kao business user)

### 4. DOBIVENI IZLAZ:

```

Main x
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Kliknut signup gumb
Uneseno ime ImeOrganizacijeZaTestiranje u polje za username
Unesen email privatniMail@email.mail u polje za email
Unesena lozinka OrGaNiZACIJA123 u polje za lozinku
Unesena lozinka OrGaNiZACIJA123 u polje za ponovan upis lozinke
Kliknut checkbox za stvaranje računa organizacije
Uneseno ime ogrNadSvima u polje za unos imena organizacije
Kliknut signup gumb za završetak registracije
Uneseno ime ImeOrganizacijeZaTestiranje u polje za unos imena na login stranici
Unesena lozinka OrGaNiZACIJA123 u polje za unos lozinke na login stranici
Kliknut gumb login
Pomakni miš na nav bar i provjeri ima li gumb 'Profile'
Test uspješan
    
```

## TEST 2 - Prijava postojećeg privatnog korisnika

## 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Podaci za prijavu:
- Korisničko ime: greg
- Lozinka: password456

## 2. KORACI ISPITIVANJA:

- Pokreće se Chrome preglednik i otvara URL aplikacije <http://localhost:5173/>
- Koristi se XPath za pronalazak navigacijskog izbornika
- Klik na gumb za prijavu
- Korisničko ime: unosi se greg
- Lozinka: unosi se password456
- Nakon unosa podataka klikne se na gumb za potvrdu prijave
- Koristi se XPath za provjeru prisutnosti navigacijskog gumba "Search"

## 3. OČEKIVANI IZLAZ:

- Nakon prijave, navigacijski izbornik prikazuje gumb "Search" na drugom mjestu, što ukazuje na uspješan login postojećeg privatnog korisnika
- Poslovni korisnik bi imao na 2. mjestu navbara gumb "Profile"

## 4. DOBIVENI IZLAZ:

```

Main x
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime greg u polje za username
Unesena lozinka password456 u polje za lozinku
Kliknut gumb login
Pomakni miš na nav bar i provjeri jeli gumb 'Search' na drugom mjestu
Test uspješan
    
```

## TEST 3 - Testiranje funkcionalnosti pretraživanja lokalnih igara kao poslovni korisnik

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>

- Podaci za prijavu: fez(korisničko ime) i password123(lozinka)
- Klik na polje izbornika "Game type"
- Klik na izbor "Local"
- Upis radijusa: 5 km

## 2. KORACI ISPITIVANJA:

- Pokreće se Chrome preglednik i otvara URL aplikacije <http://localhost:5173/>
- Navigacijski izbornik koristi se za pronalaženje i klik na gumb "Login"
- Polje za korisničko ime ispunjava se s vrijednošću "fez"
- Polje za lozinku ispunjava se s vrijednošću "password123"
- Klik na gumb za prijavu
- Klik na gumb "Search" u navigacijskom izborniku
- Klik na padajući izbornik za odabir vrste igre
- Klik na opciju "Local"
- Radijus pretraživanja postavlja se na 5 km
- Klik na gumb "Apply Filters" za potvrdu pretraživanja

## 3. OČEKIVANI IZLAZ

- Broji se broj igara u rezultatu pretraživanja
- Ako postoji samo jedna igra, test je uspješan, ako ne test nije uspješan

## 4. DOBIVENI IZLAZ



```
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime fez u polje za username
Unesena lozinka password123 u polje za lozinku
Kliknut gumb login
Pomakni miš na nav bar
Klikni search gumb
Klikni na Drop down meni za game type
Klikni na opciju 'Local'
U polje za unos radiusa unesi vrijednost 5
Klikni na gumb Apply Filters
Pobroji koliko igri je sa filterom vraćeno
Provjeri jeli dobar broj igri vraćen
Test uspješan
```

○

## TEST 4 - Testiranje funkcionalnosti pretraživanja svih igara za koje treba prijava kao privatni korisnik

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Korisničko ime: greg
- Lozinka: password456
- Klik na gumb "Search" u navigacijskom izborniku
- Klik na gumb "Application Type"

### 2. KORACI ISPITIVANJA

- Chrome preglednik se pokreće s pomoću Seleniuma
- Otvara se aplikacija na URL-u: <http://localhost:5173/>
- Navigacijski izbornik koristi se za pronalaženje gumba "Login"
- Klik na gumb "Login"
- Polje za korisničko ime ispunjava se s vrijednošću greg
- Polje za lozinku ispunjava se s vrijednošću password456
- Klik na gumb za prijavu
- Navigacijski izbornik koristi se za pronalaženje gumba "Search"
- Klik na gumb "Search"
- Klik na gumb za označavanje "Application Required"

- Klik na gumb "Apply Filters" za potvrdu pretraživanja
- Broji se broj prikazanih igara u rezultatu pretraživanja

### 3. OČEKIVANI IZLAZ

- Ako je broj igara jednak očekivanom broju (7), ispisuje se: "Test uspješan"
- Ako broj igara nije jednak očekivanom broju ili dođe do pogreške, ispisuje se: "Test neuspješan"

### 4. DOBIVENI IZLAZ

```
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime greg u polje za username
Unesena lozinka password456 u polje za lozinku
Kliknut gumb login
Pomakni miš na nav bar
Klikni search gumb
Klikni na check box 'Application Required' kako bi ju uključio
Klikni na gumb Apply Filters
Pobroji koliko igri je sa filterom vraćeno
Provjeri jeli dobar broj igri vraćen
Test uspješan
```

## TEST 5 - Testiranje funkcionalnosti izmjene imena igre kao privatni korisnik

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Korisničko ime: greg
- Lozinka: password456
- Klik na gumb "Edit" u listi korisnikovih igara
- Upis novog imena igre: NovolmeZalgru
- Klik na gumb "Submit" u formi za izmjenu igre

### 2. KORACI ISPITIVANJA

- Chrome preglednik se pokreće s pomoću Seleniuma
- Otvara se aplikacija na URL-u: <http://localhost:5173/>
- Navigacijski izbornik koristi se za pronalaženje gumba "Login"
- Klik na gumb "Login"

- Polje za korisničko ime ispunjava se s vrijednošću greg
- Polje za lozinku ispunjava se s vrijednošću password456
- Klik na gumb za prijavu
- Klik na gumb "Edit" igre
- Polje za ime igre ispunjava se s vrijednošću NovoImeZaIgru
- Klik na gumb "Submit" za potvrdu izmjene igre
- Provjerava se jeli novo ime igre prikazano na listi

### 3. OČEKIVANI IZLAZ

- Ako je ime igre jednako onome koje je upisano, ispisuje se: "Test uspješan"
- Ako ime igre nije jednako onome koje je upisano ili dođe do pogreške, ispisuje se: "Test neuspješan"

### 4. DOBIVENI IZLAZ

```
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime greg u polje za username
Unesena lozinka password456 u polje za lozinku
Kliknut gumb login
Klikni na edit gumb
Unesi novo ime igre NovoImeZaIgru
Klikni na gumb Submit
Provjeri jeli se ime igre promjenilo
Test uspješan
```

## TEST 6 - Testiranje funkcionalnosti stvaranja igre kao privatni korisnik

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Korisničko ime: greg
- Lozinka: password456
- Klik na gumb "+" u listi korisnikovih igara
- Upis imena igre: Testiranje
- Klik na gumb "Medium" u formi za opciju Complexity Level
- Upis estimated length igre: 1 sat

- Upis pravila igre: Nema pravila
- Upis Max broja igrača: 456
- Klik na gumb "No" u formi za opciju Form Required
- Upis komunikacijskog kanala igre: Discord
- Klik na gumb "No" u formi za opciju Is Homebrew
- Klik na gumb "Submit" u formi
- Klik na gumb "Search"
- Upis imena igre u polje za pretraživanje po imenu: Testiranje
- Klik na gumb "Include Full Games"
- Klik na gumb "Application Required"
- Klik na gumb "Apply Filters"

## 2. KORACI ISPITIVANJA

- Chrome preglednik se pokreće s pomoću Seleniuma
- Otvara se aplikacija na URL-u: <http://localhost:5173/>
- Navigacijski izbornik koristi se za pronalaženje gumba "Login"
- Klik na gumb "Login"
- Polje za korisničko ime ispunjava se s vrijednošću greg
- Polje za lozinku ispunjava se s vrijednošću password456
- Klik na gumb za prijavu
- Klik na gumb "+" u listi korisnikovih igara
- Polje za ime igre ispunjava se s vrijednošću "Testiranje"
- Odabire se opcija "Medium" kao Complexity Level igre
- Polje za predviđeno trajanje igre ispunjava se s vrijednošću "1 sat"
- Polje za pravila ispunjava se s vrijednošću "Nema pravila"
- Polje za maksimalan broj igrača ispunjava se s vrijednošću "456"
- Odabire se opcija "No" za opciju Form Required
- Polje za komunikacijski kanal igre ispunjava se s vrijednošću "Discord"
- Odabire se opcija "No" za opciju Is Homebrew
- Klik na gumb "Submit" za potvrdu stvaranja igre
- Klik na gumb "Search"
- Polje za pretraživanje po imenu igre ispunjava se s vrijednošću "Testiranje"
- Klik na gumb "Include Full Games"

- Klik na gumb "Application Required"
- Klik na gumb "Apply Filters"
- Provjerava se jeli igra pronađena

### 3. OČEKIVANI IZLAZ

- Ako je igra pronađena, ispisuje se: "Test uspješan"
- Ako igra nije pronađena ili dođe do pogreške, ispisuje se: "Test neuspješan"

### 4. DOBIVENI IZLAZ

```
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime greg u polje za username
Unesena lozinka password456 u polje za lozinku
Kliknut gumb login
Klikni na gumb za stvaranje igre
Unesi ime Testiranje igre u polje za Game Title
Odaberi complexity level medium
Unesi otprilike trajanje igre 1 sat u polje za Estimated length
Unesi pravila igre Nema pravila u polje za Rules
Unesi max broj igrača igre 456 u polje za Max Number Of Players
Odaberi Form Required No
Unesi komunikacijski kanal igre Discord u polje za Communication Channel
Odaberi Homebrew No
Pritisni Submit
Pomak miša na nav bar
Klikni na gumb Search
Unesi ime igre Testiranje u polje za pretraživanje po nazivu
Klikni na Checkbox 'Include Full Games'
Klikni na Checkbox 'Application Required'
Klikni na gumb 'Apply Filters'
Provjeri jeli se igra pronašla
Test uspješan
```

## TEST 7 - Testiranje funkcionalnosti pregleda nadolazeće prijave kao privatni korisnik

### 1. ULAZI:

- URL aplikacije: <http://localhost:5173/>
- Korisničko ime: greg

- Lozinka: password456
- Klik na gumb "Incoming Requests" na kartici igre
- Klik na gumb "View Form Answers" u formi za pregled nadolazećih prijava
- Klik na gumb "Deny" u formi za pregled nadolazećih prijava
- Klik na gumb "Close" u formi nadolazećih prijava

## 2. KORACI ISPITIVANJA

- Chrome preglednik se pokreće s pomoću Seleniuma
- Otvara se aplikacija na URL-u: <http://localhost:5173/>
- Navigacijski izbornik koristi se za pronalaženje gumba "Login"
- Klik na gumb "Login"
- Polje za korisničko ime ispunjava se s vrijednošću greg
- Polje za lozinku ispunjava se s vrijednošću password456
- Klik na gumb za prijavu
- Klik na gumb "Incoming Requests" u listi korisnikovih igara
- Klik na gumb "View Form Answers" u formi nadolazećih prijava
- Klik na gumb "Deny" u formi nadolazećih prijava
- Klik na gumb "Close" u formi nadolazećih prijava

## 3. OČEKIVANI IZLAZ

- Ako se nije dogodila pogreška u procesu, ispisuje se: "Test uspješan"
- Ako dođe do pogreške, ispisuje se: "Test neuspješan"

## 4. DOBIVENI IZLAZ

```
D:\Java\jdk-19.0.2\bin\java.exe ...
Pomak miša na nav bar
Kliknut login gumb
Uneseno ime greg u polje za username
Unesena lozinka password456 u polje za lozinku
Kliknut gumb login
Klikni na gumb za prikaz nadolazećih prijava
Klik na gumb a prikaz odgovora korisnika
Klik na gumb 'Deny'
Klik na gumb 'Close'
Test uspješan
```

Broj otkrivenih pogrešaka tijekom testiranja sustava: 1 Tijekom testiranja je otkrivena pogreška ako se tijekom stvaranja igre ostavi default opcije "Online" pri stvaranju igre će se poslati i koordinate igre što ne bi trebalo biti moguće

Sve reference i literatura koja je pomogla pri ostvarivanju projekta:

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

## Vrijeme izrade projekta

Za izradu projektnog zadatka imali smo vremenski okvir od 17. Listopada 2024. do 24. Siječnja 2025. (ukupno 14 tjedana, 12 ne uključujući zimske praznike). Ovaj period je od nas zahtijevao kvalitetnu organizaciju rada i jasnu podjelu zadataka među članovima tima.

## Podjela zadataka

Prije prve predaje, podjela zadataka bila je u omjeru 3:3:1 (fronted:backend:baza), dok smo nakon prve predaje ustanovili da je implementacija backend-a u zaostatku, pa smo se prilagodili na podjelu 3:4 (frontend:backend), čime je jedan član trebao uložiti vremena u učenje Spring Boot razvojnog okvira.

## Tehnički izazovi i njihovo rješavanje

Jedan od glavnih tehničkih izazova s kojima smo se susreli bio je proces integracije frontenda i backenda. Komplikacije su nastale zbog nedostatka dobre komunikacije između članova tima koji su radili na frontend i backend dijelu. Implementacija određenih funkcionalnosti također je bila otežana zbog različitih radnih navika članova tima, što je dovelo do kašnjenja u implementaciji. Iako smo uspjeli prevladati ove izazove kroz pojačanu suradnju i fokus na testiranje, smatramo da bi bolja podorganizacija backend tima mogla značajno poboljšati učinkovitost.

# Stečena znanja

---

Tijekom izrade projekta, frontend tim je naučio koristiti React i testirati aplikaciju uz pomoć mock backenda pokrenutog na Node.js Express serveru. S druge strane, backend tim stekao je iskustvo u radu s Spring Boot frameworkom, što nam je omogućilo uspješno implementiranje većine željenih funkcionalnosti.

## Dodatna znanja potrebna za bržu i kvalitetniju realizaciju

---

Za bržu i kvalitetniju realizaciju projekta bilo bi korisno dodatno znanje iz područja REST web developementa kao i znanje iz područja protokola sigurnosti web-a, posebice u pogledu pisanja i praćenja opisa komunikacije između frontenda i backenda. Jasna dokumentacija i definirani standardi za API komunikaciju značajno bi ubrzali integraciju i smanjili broj nesporazuma među podtimovima.

## Nedostatne funkcionalnosti

---

- **Google Authentication** na stranici.
- **Edit igara** na stranici.
- **Prijava** na igru.

## Perspektive za nastavak rada u projektnoj grupi

---

U budućnosti, bilo bi korisno fokusirati se na dodatnu optimizaciju rada tima i poboljšanje komunikacije, kako bi se smanjila kašnjenja u implementaciji i omogućila bolja koordinacija između različitih dijelova tima.

### Revizija 0.0.0

- **Opis:** Kreirana inicijalna Home stranica
- **Autor:** Vito Vrbić
- **Datum:** 18.10.2024.

### Revizija 0.0.1

- **Opis:** Ažurirana Home stranica. Dodan popis članova tima sa svojim ulogama u razvoju projekta i



nastavnici zaduženi za tim.

- **Autor:** Vito Vrbić
- **Datum:** 23.10.2024.

### Revizija 0.1.0

- **Opis:** Kreirana inicijalna stranica za Opis projektnog zadatka.
  - Dodana sekcija *Opis problema* koja objašnjava problem i motivaciju za projekt.
  - Dodana sekcija *Korist projekta* koja objašnjava ciljeve i akcije za rješavanje opisanog problema.
  - Dodana sekcija *Ciljana skupina korisnika* koja opisuje korisnike koji će imati koristi od projekta.
  - Dodan popis *Funkcijskih zahtjeva*, razvrstanih na "Korisničke funkcionalnosti" i "Tipovi igara".
  - Dodan popis *Nefunkcijskih zahtjeva* s očekivanjima za performanse, sigurnost, skalabilnost i održavanje.
  - Dodana sekcija *Prilagodba rješenja* s odabranim pristupima za pojedine dijelove projekta.
  - Dodana sekcija *Slična rješenja* s poveznicama na slične usluge.
  - Dodana sekcija *Mogućnosti nadogradnje rješenja* s mogućnostima za dodatne funkcionalnosti.
- **Autor:** Vito Vrbić
- **Datum:** 23.10.2024.

### Revizija 0.2.0

- **Opis:** Kreirana inicijalna stranica za Specifikaciju programske potpore
  - Dodan popis *Dionika*.
  - Dodan popis *Aktora* i njihovih funkcionalnih zahtjeva.
  - Dodana slika *Dijagrama obrazaca uporabe*.
- **Autor:** Vito Vrbić
- **Datum:** 23.10.2024.

### Revizija 0.2.1

- **Opis:** Ažurirana stranica za Specifikaciju programske potpore
  - Dodan popis *Obrazaca uporabe*, svaki detaljno objašnjen s glavnim sudionikom, ciljem, sudionicima, preduvjetima, osnovnim tijekom i mogućim odstupanjima.
- **Autor:** Vito Vrbić
- **Datum:** 24.10.2024.

### Revizija 0.2.2

- **Opis:** Ažurirana stranica za Specifikaciju programske potpore
  - Dodan OpenStreetMap kao aktor.
  - Detaljno objašnjen osnovni tijek obrasca 3.2.
  - Dodan preduvjet za obrazac 5.
  - Dodani sekvencijski dijagrami za *Pretraživanje Igara*, *Prijava na Igru*, i *Stvaranje Igre*.
- **Autori:** Vito Vrbić, Maksimilijan Hižman
- **Datum:** 27.10.2024.

### Revizija 0.2.3

- **Opis:** Ažurirana stranica za Specifikaciju programske potpore
  - Detaljno objašnjeni funkcionalni zahtjevi svih aktora.
  - Obrazac uporabe 3.2 je numeriran kao *Obrazac uporabe 9*.
  - Ažurirani sekvencijski dijagrami za *Prijava na Igru* i *Pretraživanje Igara*.
- **Autor:** Vito Vrbić
- **Datum:** 28.10.2024.

### Revizija 0.4.0

- **Opis:** Kreirana inicijalna stranica za Arhitektura i dizajn sustava
  - Opisan odabrani stil arhitekture i korištena tehnologija za frontend, backend i bazu podataka.
  - Dodana slika komunikacije Klijenta i Poslužitelja.
  - Popis atributa, tipova podataka i varijabli za dijelove projekta.
- **Autor:** Vito Vrbić
- **Datum:** 9.11.2024.

### Revizija 0.5.0

- **Opis:** Kreirana inicijalna stranica za Implementaciju i korisničko sučelje.
- **Autor:** Vito Vrbić
- **Datum:** 9.11.2024.

### Revizija 0.6.0

- **Opis:** Kreirana inicijalna stranica za Zaključak i budući rad.
- **Autor:** Vito Vrbić

- **Datum:** 9.11.2024.

#### Revizija 0.7.0

- **Opis:** Kreirana inicijalna stranica za Popis literature s navedenom literaturom korištenom u projektu.
- **Autor:** Vito Vrbić
- **Datum:** 9.11.2024.

#### Revizija 0.8.0

- **Opis:** Kreirana inicijalna stranica za Prikaz aktivnosti grupe.
- **Autor:** Vito Vrbić
- **Datum:** 9.11.2024.

#### Revizija 0.9.0

- **Opis:** Kreirana inicijalna stranica za Dnevnik promjena dokumentacije.
- **Autor:** Vito Vrbić
- **Datum:** 9.11.2024.

#### Revizija 0.4.1

- **Opis:** Ažurirana stranica za Specifikaciju programske potpore
  - Opisana vrsta i implementacija baze podataka, dodan dijagram baze podataka.
  - Dodan opis tehnologije za slanje HTTP zahtjeva i slojeva prezentacije, servisa i repozitorija.
  - Ažurirani opisi komunikacije između Klijenta i Poslužitelja te Poslužitelja i Baze podataka.
  - Dodan dijagram razreda.
- **Autori:** Luka Alfirević, Vito Vrbić
- **Datum:** 10.11.2024.

#### Revizija 0.8.1

- **Opis:** Ažurirana stranica za Prikaz aktivnosti grupe s dodanim zapisima sastanaka.
- **Autor:** Luka Alfirević
- **Datum:** 10.11.2024.

#### Revizija 0.8.2

- **Opis:** Ažurirana stranica za Prikaz aktivnosti grupe s unosima aktivnosti članova tima.
- **Autori:** Vito Vrbić, Luka Alfirević

- **Datum:** 11.11.2024.

#### Revizija 0.9.1

- **Opis:** Ažurirana stranica za Dnevnik promjena dokumentacije s unesenim promjenama.
- **Autor:** Vito Vrbić
- **Datum:** 11.11.2024.

#### Revizija 0.9.2

- **Opis:** Pregledana gramatika git wiki-a.
- **Autor:** Maksimilijan Hižman
- **Datum:** 12.11.2024.

#### Revizija 1.0.1

- **Opis:** Preoblikovana Home.md stranica zbog priličenja dobivenog predloška.
- **Autor:** Vito Vrbić
- **Datum:** 27.12.2024.

#### Revizija 1.0.2

- **Opis:** Dodana tri dijagrama stanja.
- **Autor:** Vito Vrbić
- **Datum:** 1.1.2025.

#### Revizija 1.0.3

- **Opis:** Dodani opisi korištene tehnologije i alata za frontend.
- **Autor:** Vito Vrbić
- **Datum:** 1.1.2025.

#### Revizija 1.0.4

- **Opis:** Dodana stranica za detaljan opis testiranja.
- **Autor:** Vito Vrbić
- **Datum:** 9.1.2025.

#### Revizija 1.0.5

- **Opis:** Dodani opisi dijagrama stanja i uređena zaduženja članova.

- **Autor:** Vito Vrbić
- **Datum:** 19.1.2025.

### Revizija 1.0.6

- **Opis:** Dodano izvješće testiranja sustava sa alatom Selenium WebdDiver.
- **Autor:** Patrick Kliček, Maksimilijan Hižman
- **Datum:** 24.1.2025.

### Revizija 1.0.7

- **Opis:** Dodan zaključak.
- **Autor:** Vito Vrbić
- **Datum:** 24.1.2025.

### Revizija 1.1.0

- **Opis:** Dodani dijagrami komponenata i razmještaja.
- **Autor:** Nikola Kramarić
- **Datum:** 24.1.2025.

## Dnevnik sastajanja

---

### 1. sastanak

- **Datum:** 14. listopada 2024.
- **Prisustvovali:** V.Vrbić, M.Hižman, P.Kliček, L.Alfirević, V.Lekić, P.Blašković, N.Kramarić
- **Teme sastanka:**

- Podjela članova u uloge i timove
- Uspostavljanje voditelja za frontend, backend i cjelokupni projekt
- Raspravljanje o generalnoj logici projekta
- Raspravljanje o budućim zadatcima za pojedine članove
- Upoznavanje, team-building
- Raspravljanje o svačijim vještinama, iskustvima i preferencijama
- Raspravljanje programskih jezika, framework-ova i softvera za implementaciju projekta

- **Zaključci sastanka:**

- Uloga voditelja je dodijeljena V.Vrbiću
- Kreirana su tri tima:
  - Frontend(V.Vrbić, M.Hižman, P.Kliček) na čelu s V.Vrbićem
  - Backend(V.Lekić, P.Blašković, N.Kramarić, L.Alfirević) na čelu s V.Lekićem
  - Baza podataka(L.Alfirević)
- React i Spring Boot su odabrani kao frameworkovi za projekt, a PostgreSQL je odabran za bazu podataka
- Članovima su dodijeljene ispravne uloge na zajedničkom Discord serveru

## 2. sastanak

- Datum: 21. listopada 2024.
- Prisustvovali: V.Vrbić, M.Hižman, P.Kliček, A.Luka
- Teme sastanka:
  - Dizajn web aplikacije
  - Implementacija frontend strane projekta
  - User interface
  - Podjela zadataka unutar frontend tima
- Zaključci sastanka:
  - Odabran je generalni dizajn web aplikacije te je određen broj web stranica u sklopu web aplikacije

## 3. sastanak

- Datum: 9. studenoga 2024.
- Prisustvovali: V.Lekić, N.Kramarić
- Teme sastanka:
  - Provjeravanje poslovne logike backend strane projekta
  - Raspravljanje o servisima i klasnim podjelama
  - Uspoređivanje napretka

## 4. sastanak

- Datum: 10. studenoga 2024.
- Prisustvovali: V.Vrbić, L.Alfirević, V.Lekić

- Teme sastanka:

- Raspravljanje o http zahtjevima i odgovorima
- Raspravljanje o dohvatima informacija sa servera na browser i logici kod osvježivanja stranice
- Usklađivanje klasa u backend strani projekta i dijagrama razreda
- Usklađivanje frontend i backend strane projekta
- Podjela zadataka prije roka za prvu reviziju projekta

- Zaključci sastanka:

- Backend kod i dijagram razreda su usklađeni
- Frontend i backend su usklađeni radi osnovnih funkcionalnosti zahtjeva:
  - Login
  - Sign-up
  - Pretraživanje igara
- Dogovorene su specifikacije http zahtjeva i kako su oni procesuirani, s naglaskom na zahtjevima koji su relevantni za pretraživanje igara
- Raspravljan issue: <https://github.com/vito-vrbic/Nat20Mappers/issues/13>

## Plan rada

---

## Aktivnosti

---

## Vremenski angažman

---

## Dokumentacija

- Opis projektnog zadatka (Vito Vrbić - 2h)
- Funkcionalni zahtjevi (Vito Vrbić - 3h)
- Opis pojedinih obrazaca (Vito Vrbić - 1h, Maksimilijan Hižman - 0.5h)
- Dijagram obrazaca (Vito Vrbić 2.5h)
- Sekvencijski dijagrami (Vito Vrbić - 2h, Luka Alfrević - 2h)
- Opis ostalih zahtjeva (Vito Vrbić - 0.5h)

- Arhitektura i dizajn sustava (Vito Vrbić - 1h)
- Baza podataka (Luka Alfirević - 1.5h + 1.5h)
- Dijagram razreda (Luka Alfirević - 6h)
- Dijagram stanja (Vito Vrbić - 1.5h)
- Dijagram aktivnosti (Vito Vrbić - 1h)
- Dijagram komponenti
- Korištene tehnologije i alati (Vito Vrbić - 1.5h Vedran Lekić 1h)
- Dijagram razmještaja
- Ispitivanje programskog rješenja (Patrick Kliček - 1.5h)
- Upute za puštanje u pogon(Vedran Lekić 1h)
- Dnevnik sastajanja (Luka Alfirević - 1h)
- Dnevnik promjena dokumentacije (Patrick Kliček - 2.5h)
- Zaključak i budući rad
- Popis literature
- Prezentacija (Vito Vrbić - 0.5h)

## Izrada projektnog rješenja

- Izrada: Frontend (Vito Vrbić - 12h + 26h, Maksimilijan Hižman - 8.5h + 18h, Patrick Kliček - 10h + 18h)
- Izrada: Baza podataka (Luka Alfirević - 10h + 5h)
- Izrada: Backend (Patrik Blašković - 4.5h, Nikola Kramarić - 8h, Vedran Lekić - 27+13 h, Luka Alfirević - 22h)

## Ostalo

- Upravljanje projektom (Vito Vrbić - 12h + 10h)
- Ispitivanje programskog rješenja (Maksimilijan Hižman - 3h, Nikola Kramarić - 1h, Patrick Kliček 5h)
- Uređivanje dokumenta o zaključcima ispitivanja (Maksimilijan Hižman - 0.5h)
- Provjeravanje pravopisa (Maksimilijan Hižman - 2h)
- Deployment (Patrik Blašković - 2h, Vedran Lekić 1h)

## Doprinosi

---



## Dokumentacija

- Opis projektnog zadatka (Vito Vrbić)
- Funkcionalni zahtjevi (Vito Vrbić)
- Opis pojedinih obrazaca (Vito Vrbić)
- Provjera obrazaca (Maksimilijan Hižman)
- Dijagram obrazaca (Vito Vrbić)
- Sekvencijski dijagrami (Vito Vrbić - 3 dijagrama, Luka Alfirević - 2 dijagrama (koristili smo 1 od njih))
- Opis ostalih zahtjeva (Vito Vrbić)
- Arhitektura i dizajn sustava (Vito Vrbić, Vedran Lekić)
- Baza podataka (Luka Alfirević)
- Dijagram razreda (Luka Alfirević)
- Dijagram stanja (Vito Vrbić)
- Dijagram aktivnosti (Vito Vrbić)
- Dijagram komponenti
- Korištene tehnologije i alati (Vito Vrbić, Vedran Lekić)
- Dijagram razmještaja
- Ispitivanje programskog rješenja (Patrick Kliček)
- Upute za puštanje u pogon (Vedran Lekić)
- Dnevnik sastajanja (Luka Alfirević)
- Dnevnik promjena dokumentacije (Patrick Kliček)
- Zaključak i budući rad
- Popis literature
- Provjera pravopisa, čitljivosti i općega dojma (Maksimilijan Hižman)
- Prezentacija (Vito Vrbić)

## Izrada projektnog rješenja

- Kompletan početni dizajn frontenda (Patrick Kliček)
- Naknadni dizajn frontenda (Vito Vrbić)
- Home Page (Stiliziranje - Maksimilijan Hižman, Patrick Kliček, Vito Vrbić)
- Search Page (Stiliziranje - Patrick Kliček, Vito Vrbić; Funkcionalnost - Vito Vrbić, Maksimilijan Hižman)
- Google Auth (Funkcionalnost i stiliziranje na frontendu - Maksimilijan Hižman)

- Login Page (Stiliziranje - Maksimilijan Hižman, Patrick Kliček, Vito Vrbić; Funkcionalnost - Vito Vrbić)
- Signup Page (Stiliziranje - Maksimilijan Hižman, Patrick Kliček; Funkcionalnost - Vito Vrbić)
- Dashboard Page (Stiliziranje - Patrick Kliček; Funkcionalnost - Vito Vrbić)
- Profile Page (Stiliziranje - Vito Vrbić; Funkcionalnost - Vito Vrbić)
- Profile Edit Page (Stiliziranje - Vito Vrbić; Funkcionalnost - Vito Vrbić)
- Forma za stvaranje igre (Stiliziranje - Patrick Kliček; Funkcionalnost - Patrick Kliček)
- Forma za izmjenu igre (Stiliziranje - Patrick Kliček; Funkcionalnost - Patrick Kliček)
- Forma za pregled i obradu nadolazećih igara (Stiliziranje - Patrick Kliček; Funkcionalnost - Patrick Kliček)
- Routing na frontendu (Vito Vrbić)
- Baza podataka (Luka Alfrević)
- Izrade klasa (Vedran Lekić, Patrik Blašković)
- Dockerizacija i deployment (Patrik Blašković)
- H2 baza podataka (Vedran Lekić)
- Search, Login, Logout Kontroleri (Vedran Lekić, Nikola Kramarić)
- Service & Repository development (Vedran Lekić, Nikola Kramarić)
- Matematičko modeliranje haversine funkcije (Vedran Lekić)

## Ostalo

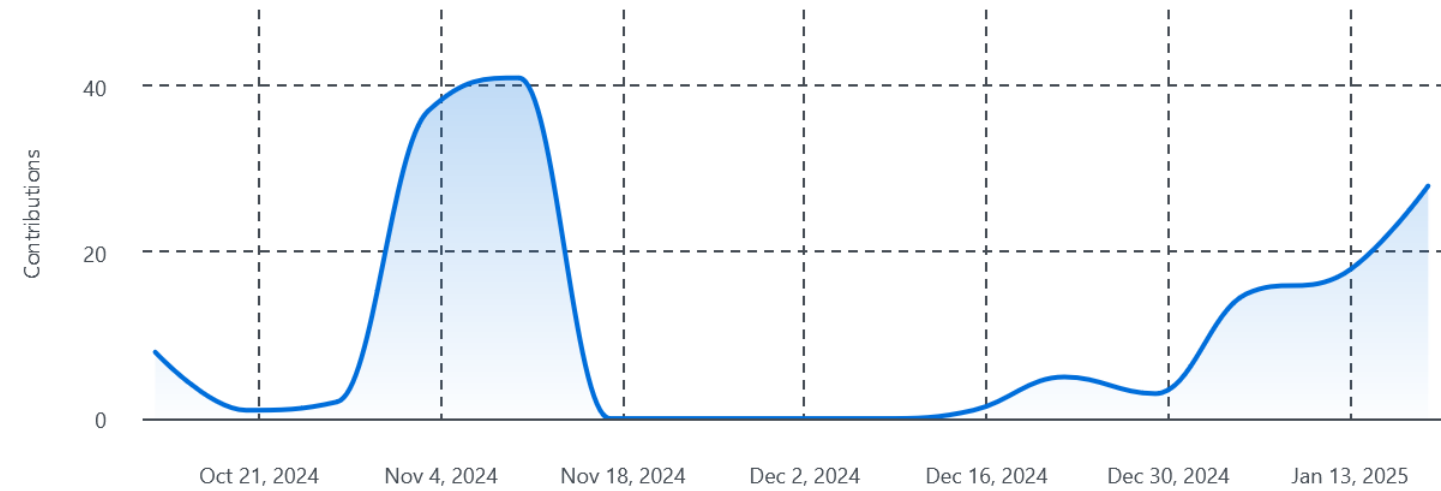
- Testiranje login i signup stranice (Maksimilijan Hižman, Vedran Lekić)
- Testiranje search funkcionalnosti (Maksimilijan Hižman, Vedran Lekić, Nikola Kramarić)
- Interna dokumentacija komunikacije frontenda i backenda (Vito Vrbić)
- Ispitivanje sustava (Patrick Kliček)

## Dijagram pregleda promjena

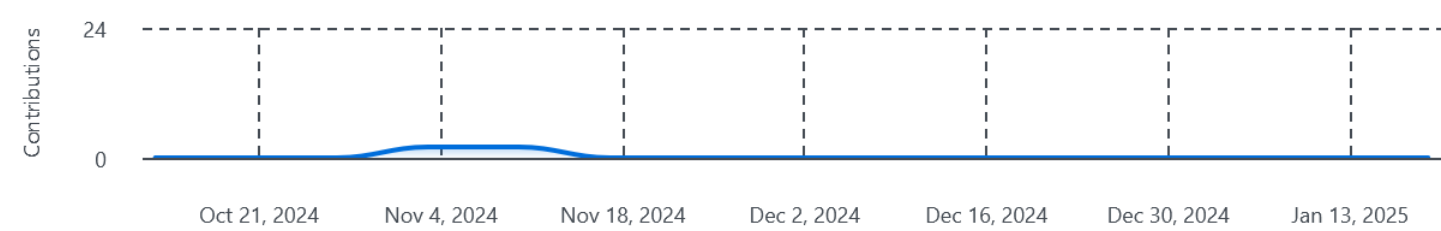
---

Commits over time

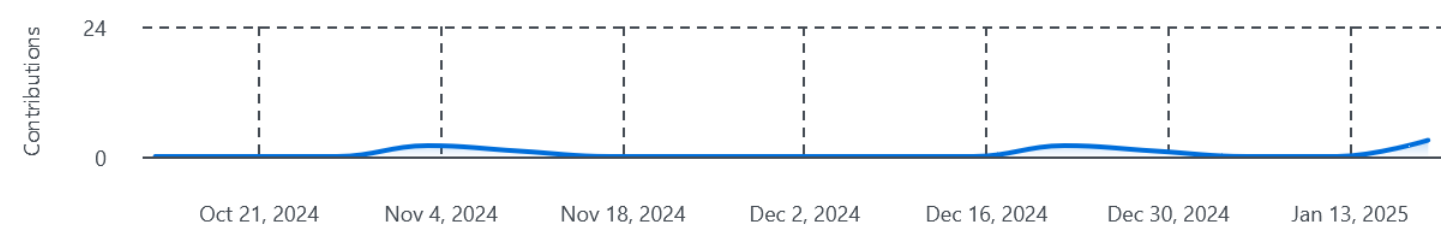
Weekly from Oct 13, 2024 to Jan 19, 2025



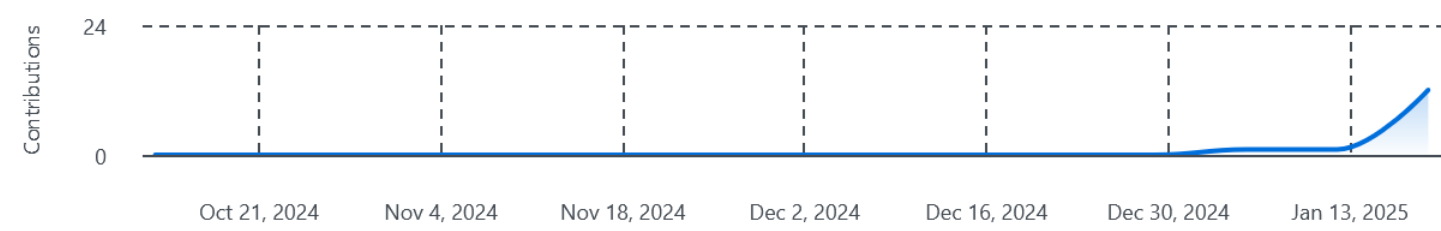
Kramaric79's Commits



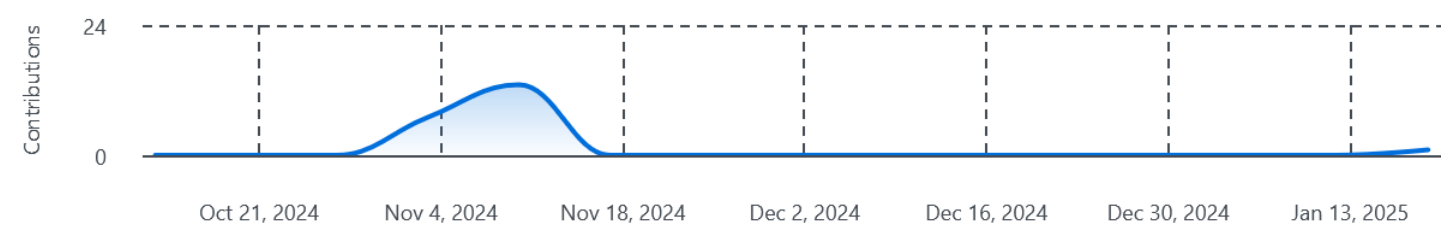
PatrickKlicek's Commits



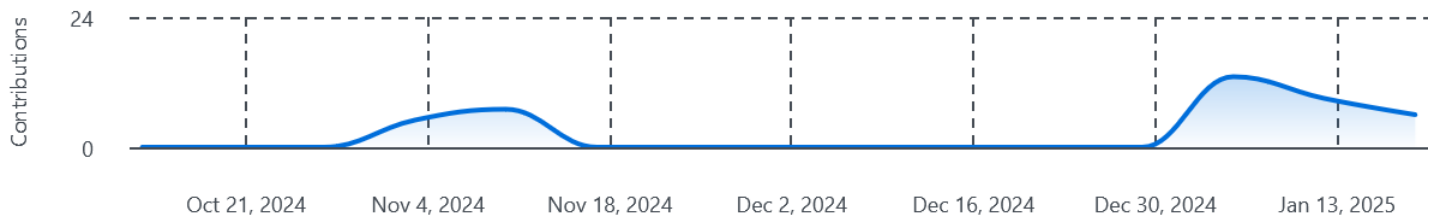
Canththinkzzz's Commits



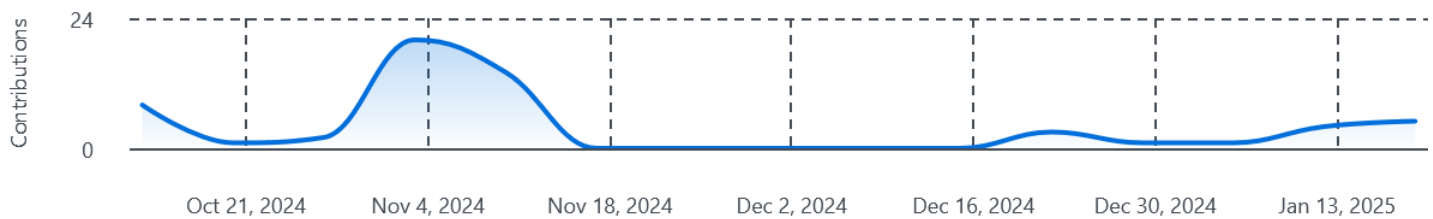
VedranLekic's Commits



## PatrikB1's Commits



## vito-vrbic's Commits



# Ključni izazovi i rješenja

- Opis izazova: Tijekom izrade projekta, suočili smo se s nekoliko ključnih izazova. Jedan od najznačajnijih bio je kašnjenje u razvoju, osobito zbog nedostatka koordinacije između frontend i backend timova. Integracija između ova dva dijela aplikacije bila je komplicirana, jer su se pojavili problemi u komunikaciji i nesuglasice u pogledu implementacije određenih funkcionalnosti. Dodatno, radne navike i organizacija unutar backend tima dovele su do zakašnjenja u implementaciji, što je usporilo napredak cijelog projekta.
- Rješenja: Kako bi se prevladali ovi izazovi, poduzeti su određeni koraci kako bi se poboljšala suradnja i organizacija. Prvo, povećana je komunikacija među članovima tima, organizirali smo redovite sastanke za koordinaciju između frontenda i backenda, čime je olakšana integracija. Također, nakon prve predaje, prilagodili smo podjelu zadataka, povećavši angažman backend tima kako bismo nadoknadili zaostatke. Osim toga, usmjerili smo veći fokus na testiranje i reviziju koda, što je omogućilo pravovremeno otkrivanje problema i njihovo rješavanje prije finalne verzije.
- Naučene lekcije: Iz ovog iskustva naučili smo važnost jasne podjele zadataka i bolje organizacije unutar timova, osobito kod složenijih projekata koji zahtijevaju integraciju različitih tehnologija. Također, uočili smo koliko je bitno unaprijed definirati komunikacijske kanale između frontenda i backenda te postaviti jasne smjernice za razvoj API-ja i njihovu dokumentaciju. Povećana suradnja i transparentnost u komunikaciji doprinijeli su bržem rješavanju problema i uspješnijoj realizaciji projekta.

U ovom odjeljku dokumentacije dajemo detaljne smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije. Cilj je naravno olakšati postavljanje aplikacije na razvojnom, ispitnom i

produkcijском okruženju.

## 1.Instalacija

Ovdje treba navesti korake potrebne za instalaciju svih potrebnih komponenti:

- Node.js i npm: Node.js v20.13.1, npm 10.5.2
- React: React 18.3.1
- Vite: Vite 4.3.9
- Maven: Apache Maven 4.0.0
- Docker: Docker 27.3.1
- PostgreSQL klijent psql: PostgreSQL 14.15
- Git: Git 2.47.0
- Eclipse
- Visual Studio Code
- **Preuzimanje:** Kloniranje github repozitorija koji se nalazi na stranici: <https://github.com/vito-vrbic/Nat20Mappers>

```
git clone https://github.com/vito-vrbic/Nat20Mappers.git
```

Pozicionirajte se u direktorij ./Frontend/ttrpg-finder-frontend U direktoriju ove aplikacije frontend ovisnosti se instaliraju sljedećom naredbom: npm install Pokretanje lokalnog razvojnog poslužitelja: Za pokretanje frontend aplikacije lokalno koristi se naredba:

- npm run dev
- Pristup frontend aplikaciji:
- Nakon pokretanja, aplikacija je dostupna na URL-u:
- <http://localhost:5000>
- 

Globalni pristup staroj frontend aplikaciji: <https://nat20mappers-frontend.onrender.com/> iako je aplikacija integrirana **statički**.

- Statičku integraciju možete izvesti na sljedeći način:

Pozicionirajte se u ttrpg-finder-frontend folder preko cmd-a (start gumb -> command-prompt ikona na Windowsima). Unutar foldera mora biti **package.json**. Zatim naredite npm install te run build. Rezultatne podatke assets i index.html prebacite u vaš static folder koji će se nalaziti na

Eclipse Maven projektnoj datoteci.

Frontend aplikacija postavljena je na Render platformi i dostupna globalno putem URL-a:

**\*\*<https://nat20mappers-backend.onrender.com/> \*\***

## 2. Postavke

Detaljne upute za konfiguraciju aplikacije: Aplikacija je konfigurirana sljedećim svojstvima:

spring.application.name=TTRPGBE-1

## Baza Postavke

spring.datasource.url=jdbc:[postgresql://localhost:5432/ttrpg\\_base\\_second\\_edition](#)

spring.datasource.username=postgres spring.datasource.password=lozinka spring.datasource.driver-class-name=org.postgresql.Driver

## JPA Postavke

spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect spring.jpa.hibernate.ddl-auto=update server.port=5000 server.error.include-message=always server.error.include-binding-errors=always spring.security.user.name=Vedran spring.security.user.password=Nardev

spring.security.oauth2.client.provider.google.authorization-uri=<https://accounts.google.com/o/oauth2/v2/auth> spring.security.oauth2.client.provider.google.token-uri=<https://oauth2.googleapis.com/token> spring.security.oauth2.client.provider.google.user-info-uri=<https://openidconnect.googleapis.com/v1/userinfo> spring.security.oauth2.client.provider.google.jwk-set-uri=<https://www.googleapis.com/oauth2/v3/certs> spring.security.oauth2.client.provider.google.user-name-attribute=name

spring.security.oauth2.client.registration.google-login.client-id=1056620738197-emfedlalpcfsfbm6aluspomv2a9fjfea.apps.googleusercontent.com  
spring.security.oauth2.client.registration.google-login.client-secret=GOCSPX-gheavnqh-0qMRdActLCwAhg7SMb\_ spring.security.oauth2.client.registration.google-login.scope=profile,email,openid spring.security.oauth2.client.registration.google-login.redirect-uri=<http://localhost:5000/login/oauth2/code/google> spring.security.oauth2.client.registration.google-login.provider=google spring.security.oauth2.client.registration.google-login.client-name=Google-Login

spring.security.oauth2.client.registration.google-signup.client-id=1056620738197-vs7puak6vajj5ff74glf02du1jq62chg.apps.googleusercontent.com

```
spring.security.oauth2.client.registration.google-signup.client-secret=GOCSPX-
XI5g7kuv4-4vZVuwHPUrOHzWt09 spring.security.oauth2.client.registration.google-
signup.scope=profile,email,openid spring.security.oauth2.client.registration.google-signup.redirect-
uri=http://localhost:5000/signup/oauth2/code/google
spring.security.oauth2.client.registration.google-signup.provider=google
spring.security.oauth2.client.registration.google-signup.client-name=Google-Signup
```

# Kako pokrenuti Google autentifikaciju

## Google Cloud Console:

1. Idite na [Google Cloud Console](#).
2. Kreirajte novi projekt.
3. Omogućite **Google+ API** u **API-ji i usluge > Knjižnica**.
4. Kreirajte OAuth2 klijentske vjerodajnice:
  - Odaberite **Web aplikacija**.
  - Postavite **Autorizirane URI-je za preusmjeravanje**:

```
http://localhost:5000/login/oauth2/code/google
http://localhost:5000/signup/oauth2/code/google
```

Upute za pokretanje aplikacije u različitim okruženjima:

- **Razvojno okruženje:**

```
npm run dev
```

- **Produksijsko okruženje:**

- Prevođenje aplikacije:

```
npm run build
```

- Pokretanje poslužitelja:

```
npm start
```

## 4. Upute za administratore

Smjernice za administratore aplikacije nakon puštanja u pogon:

- **Pristup administratorskom sučelju:**

Potrebno je uočiti da je moguće logirati se Sigurnosno preko Google autorizacije kao i preko username: Vedran password: Nardev konfiguracije u desktop verziji aplikacije na (<http://localhost:5000/login/oauth2/code/google>) te mock signup na (<http://localhost:5000/signup/oauth2/code/google>)

- **Redovito održavanje:**

- Arhiviranje baze podataka.
- Pregled logova.

```
git pull origin main
```

```
npm install
```

```
npm run build
```

```
npm start
```

## Render platforma (Cloud Deploy)

Render je popularna cloud platforma koja omogućuje jednostavno postavljanje i upravljanje aplikacijama. Priprema repozitorija:

Osigurajte da vaš projekt sadrži konfiguracijske datoteke potrebne za deploy, poput render.yaml ili Dockerfile.

yaml datoteka:

services:

- type: web

name: my-web-app

env: node

buildCommand: npm install && npm run build

startCommand: npm start

plan: free Ova konfiguracija definira osnovne postavke za pokretanje Node.js aplikacije na Render



platformi.

Postavljanje aplikacije na [Render](#):

Prijavite se na Render.

Kreirajte novi Web Service i povežite ga s vašim GitHub repozitorijem.

**Konfigurirajte postavke prema potrebi (npr. build i start komande).**

Dodajte potrebne environment varijable (npr. DATABASE\_URL, API\_KEY).

Pokretanje aplikacije: Render će automatski preuzeti vaš repozitorij, instalirati sve ovisnosti te izgraditi i pokrenuti aplikaciju. Nakon deploya, aplikaciji možete pristupiti putem generiranog URL-a (<https://nat20mappers-backend.onrender.com>).

Opis prisutpa aplikaciji na javnom poslužitelju Pristup aplikaciji Dokumentirajte postupak i pružite jasne smjernice za korištenje aplikacije na javnom poslužitelju. Navedite ograničenja, ako koristite besplatni plan performanse vam mogu biti značajno ograničene.

U uputama obuhvatite kako korisnici mogu pristupiti aplikaciji putem internetskog preglednika. Priložite korake za pristup administratorskom sučelju ako je primjenjivo.

- **Pokretanje aplikacije:**
- Render će automatski preuzeti repozitorij, instalirati ovisnosti i
- pokrenuti aplikaciju. Nakon deploya, aplikaciji možete pristupiti
- putem generiranog URL-a (<https://nat20mappers-backend.onrender.com>).

\*\*

- Posjetite stranicu na <https://nat20mappers-backend.onrender.com>
- Logirajte se u glavni sistem preko sistemskih konfiguracija ili google log in-a
- Registrirajte se kao poslovni ili privatni korisnik
- Radite igre, tražite igre i uživajte

## Korištene tehnologije i alati za "Backend"

**Cilj:** Jasno ćemo i precizno opisati tehnologije korištene u projektu kako bi se olakšalo održavanje, proširenje i suradnja u timu.

# Tehnologije koje smo koristili u projektu

---

## 1. Spring Boot

---

- **Verzija:** 3.3.7
- **Opis:** Okvir koji pojednostavljuje razvoj Spring aplikacija pružajući zadane konfiguracije i ugrađene poslužitelje.
- **Primjena:** Korišten za izradu web aplikacija, REST API-ja i integraciju s bazama podataka.

## 2. H2 Baza Podataka

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Ugrađena baza podataka u memoriji ili temeljena na datotekama, idealna za razvoj i testiranje.
- **Primjena:** Korištena za brzo testiranje i razvoj bez potrebe za vanjskom bazom podataka.

## 3. Spring Security

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Okvir za osiguravanje aplikacija, pružajući mogućnosti autentifikacije i autorizacije.
- **Primjena:** Korišten za zaštitu aplikacije i integraciju s Google OAuth2 za prijavu i registraciju.

## 4. OAuth2 Client (Google Integracija)

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Biblioteka za autentifikaciju temeljenu na OAuth2, poput Google prijave i registracije.
- **Primjena:** Omogućuje korisnicima prijavu i registraciju putem Google računa.

## 5. Thymeleaf

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Poslužiteljska templating platforma za renderiranje HTML pogleda u web aplikacijama.
- **Primjena:** Korišten za generiranje dinamičkih web stranica.

## 6. Java

---

- **Verzija:** 21
- **Opis:** Programski jezik za izradu robusnih i skalabilnih aplikacija.
- **Primjena:** Glavni programski jezik korišten u projektu.

## 7. JSON Web Token (JWT)

---

- **Verzija:** 0.11.5
- **Opis:** Standard za siguran prijenos informacija kao JSON objekata, često korišten za autentifikaciju.
- **Primjena:** Korišten za generiranje i provjeru tokena za autentifikaciju korisnika.

## 8. PostgreSQL

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Moćan, open-source sustav za upravljanje relacijskim bazama podataka za produkcijska okruženja.
- **Primjena:** Uključen kao ovisnost, ali trenutno nije konfiguriran za korištenje.

## 9. Lombok

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Biblioteka koja smanjuje količinu standardnog koda automatskim generiranjem gettera, settera i drugih uobičajenih metoda.
- **Primjena:** Korišten za pojednostavljenje Java koda.

## 10. Jackson

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Biblioteka za obradu JSON podataka, korištena za serijalizaciju i deserijalizaciju u REST API-ima.
- **Primjena:** Korišten za pretvorbu Java objekata u JSON i obrnuto.

## 11. Spring Boot Maven Plugin

---

- **Verzija:** Zadana (upravlja Spring Boot)
- **Opis:** Maven plugin za pakiranje i pokretanje Spring Boot aplikacija.

- **Primjena:** Korišten za build i pokretanje aplikacije.

## 12. Eclipse IDE za Java Developere (uključuje Incubating komponente)

- **Verzija:** 2022-12 (4.26.0)
- **Build ID:** 20221201-1913
- **Opis:** Integrirano razvojno okruženje (IDE) specijalizirano za razvoj Java aplikacija. Uključuje alate za pisanje, debugiranje i upravljanje projektima.
- **Ključne Značajke:**
  - **Editor Koda:** Napredno označavanje sintakse, automatsko dovršavanje koda i alati za refaktoriranje.
  - **Debugger:** Integrirani alati za otklanjanje grešaka u Java aplikacijama.
  - **Maven Integracija:** Podrška za upravljanje Maven projektima.
  - **Git Integracija:** Podrška za verzioniranje koda putem Gita.
  - **Incubating Komponente:** Eksperimentalne značajke i dodaci koji su još u razvoju.
- **Primjena:** Korišten za razvoj, debugiranje i upravljanje projektom.

### Sažetak

Tehnologija	Verzija	Svrha
Spring Boot	3.3.7	Glavni okvir za aplikaciju.
H2 Baza podataka	Zadana	Ugrađena baza za razvoj.
Spring Security	Zadana	Autentifikacija i autorizacija.
OAuth2 Client	Zadana	Google prijava/registracija.
Thymeleaf	Zadana	Generiranje HTML pogleda.
Java	21	Programski jezik.
JWT (JJWT)	0.11.5	Tokeni za autentifikaciju.
PostgreSQL	Zadana	Produksijska baza podataka.
Lombok	Zadana	Smanjenje standardnog koda.

Tehnologija	Verzija	Svrha
Jackson	Zadana	Obrada JSON podataka.
Spring Boot Maven Plugin	Zadana	Build i pokretanje aplikacije.
Eclipse IDE za Java Developere	2022-12 (4.26.0)	Razvojno okruženje za Java aplikacije.

\*\*Docker 27.4.1

# Docker

## Verzija: 27.4.1

### Opis

Docker je platforma za razvoj, isporuku i pokretanje aplikacija korištenjem kontejnera. Kontejneri omogućuju pakiranje aplikacija i svih njihovih ovisnosti u jednu jedinicu, čime se osigurava dosljedno okruženje za razvoj, testiranje i produkciju.

### Ključne Značajke

- **Kontejnerizacija:** Aplikacije se izoliraju u kontejnere, što omogućuje dosljedno pokretanje na bilo kojem sustavu.
- **Lako Pokretanje:** Kontejneri se mogu brzo pokrenuti i zaustaviti, što ubrzava razvoj i testiranje.
- **Portabilnost:** Aplikacije pakirane u Docker kontejnere mogu se pokretati na bilo kojem sustavu koji podržava Docker.
- **Ekstenzibilnost:** Podržava razne dodatke i integracije s drugim alatima (npr. Kubernetes, CI/CD alati).

### Primjena

- **Razvoj:** Omogućuje razvoj u izoliranom okruženju, čime se izbjegavaju problemi s ovisnostima.
- **Testiranje:** Kontejneri se mogu koristiti za brzo postavljanje testnih okruženja.
- **Produkcija:** Aplikacije se mogu jednostavno deployati u produkcijsko okruženje bez promjena koda.

### Osnovne Naredbe

- Pokretanje Kontejnera:

```
docker run <naziv_slike>
```

# SQL Server 17

---

## Verzija: 17

---

### Opis

SQL Server 17 je relacijski management sistem za baze podataka (RDBMS) koji je razvila tvrtka Microsoft. Ova verzija donosi brojna poboljšanja u performansama, sigurnosti i upravljanju podacima, čineći ga jednim od najpopularnijih sustava za upravljanje bazama podataka u svijetu.