*Univerzitet u Novom Sadu*

*- Fakultet tehničkih nauka -*

**SEMINARSKI RAD**

**Nastavni predmet : Tehnologije I sistemi eUprave**

**Naziv teme : Prekršajni sud**

Student

Valentina Gabrić SR39/2021

**Jun, 2024**

S A D R Ž A J

[1 Sažetak 3](#_Toc170557304)

[2 Ključne reči 3](#_Toc170557305)

[**3** **Uvod** 4](#_Toc170557306)

[**4** **Slična istraživanja** 5](#_Toc170557307)

[4.1 Korišćene tehnologije 6](#_Toc170557308)

[**5** **Specifikacija zahteva** 8](#_Toc170557309)

[5.1 Slučajevi korišćenja: 8](#_Toc170557310)

[5.2 **Specifikacija nefunkcionalnih zahteva** 13](#_Toc170557311)

[**6** **Specifikacija dizajna** 14](#_Toc170557312)

[6.1 Arhitektura sistema 14](#_Toc170557313)

[6.2 Dizajn baze podataka 14](#_Toc170557314)

[6.3 . Korisnički interfejs 14](#_Toc170557315)

[6.4 . Bezbednosni dizajn 14](#_Toc170557316)

[**7** **Demonstracija** 17](#_Toc170557317)

# Sažetak

Ovaj rad predstavlja sistem Prekršajnog suda, koji se bavi procedurom donošenja presuda na osnovu počinjenih prekršaja. Projekat obuhvata korišćenje UML dijagrama klasa i dijagrama slučajeva korišćenja za modeliranje. Programska rešenja uključuju korišćenje jezika Go za backend, Angular za frontend, dokerizaciju servisa, single sign-on za autentifikaciju korisnika, i mikroservisnu arhitekturu za komunikaciju između servisa. Implementacijom ovog sistema omogućena je efikasna obrada i pregled podataka vezanih za prekršaje, čime se poboljšava celokupni rad Prekršajnog suda.

# Ključne reči

* Prekršajni sud
* Presude
* Prekršaji
* UML dijagrami
* Use case dijagrami
* Programsko rešenje
* Autentifikacija korisnika
* Go (programski jezik)
* Angular
* Dokerizacija
* Single sign-on
* Mikroservisna arhitektura
* Komunikacija između servisa
* Efikasna obrada podataka

# **Uvod**

U današnjem digitalnom dobu, efikasnost i transparentnost pravosudnih sistema su od ključne važnosti za obezbeđivanje pravičnosti i brze obrade pravnih postupaka. U skladu s tim, implementacija modernih tehnoloških rešenja u radu Prekršajnog suda predstavlja vitalan korak ka unapređenju celokupnog pravnog sistema. Ovaj rad istražuje sistem **Prekršajnog suda** kroz primenu UML dijagrama klasa i dijagrama slučajeva korišćenja, koristeći napredne tehnologije kao što su mikroservisna arhitektura, jezik Go za backend, Angular za frontend, dokerizacija servisa, single sign-on za autentifikaciju korisnika i efikasna komunikacija između servisa. Fokus je na optimizaciji procesa donošenja presuda za prekršaje, što dovodi do brže i preciznije obrade pravnih predmeta. Ovaj rad pruža detaljan uvid u implementaciju tehnološki podržanog sistema koji transformiše način rada Prekršajnog suda, unapređujući efikasnost i transparentnost pravnog sistema.

Fokus istraživanja je na optimizaciji procesa donošenja presuda za prekršaje, što dovodi do brže i preciznije obrade pravnih predmeta. Ovaj rad pruža detaljan uvid u implementaciju tehnološki podržanog sistema koji transformiše način rada Prekršajnog suda, unapređujući efikasnost i transparentnost pravnog sistema u Srbiji. Kroz analizu UML dijagrama klasa i dijagrama slučajeva korišćenja, kao i primenu naprednih programskih rešenja, ovaj rad ilustruje kako digitalizacija može poboljšati procese evidencije, analize i donošenja odluka u pravosuđu.

# **Slična istraživanja**

 **Implementacija elektronskog vođenja predmeta u prekršajnom postupku**  
Ovo istraživanje istražuje implementaciju elektronskog vođenja predmeta u prekršajnom sudstvu, analizirajući benefite u efikasnosti, transparentnosti i smanjenju administrativnih troškova.

 **Primena UML dijagrama i dijagrama slučajeva korišćenja u modeliranju prekršajnih sudova**  
Istraživanje koje proučava kako se UML dijagrami klasa i dijagrami slučajeva mogu primeniti za modeliranje procesa u prekršajnom sudstvu, sa ciljem optimizacije rada i donošenja odluka.

 **Autentifikacija korisnika i sigurnost podataka u prekršajnim sudovima**  
Ova studija se fokusira na implementaciju sigurnosnih rešenja, uključujući single sign-on (SSO) i druge metode autentifikacije, radi zaštite podataka u informacionim sistemima prekršajnih sudova.

 **Dokerizacija servisa u prekršajnom pravosuđu: primeri dobre prakse**  
Istraživanje koje identifikuje primere dobre prakse u korišćenju dokerizacije servisa za optimizaciju infrastrukture i resursa u prekršajnim sudovima.

 **Korišćenje naprednih analitičkih alata za efikasniju obradu prekršajnih predmeta**  
Analiza primene naprednih analitičkih alata poput veštačke inteligencije i analize velikih podataka za identifikaciju uzoraka i optimizaciju procesa u prekršajnom pravosuđu.

## Korišćene tehnologije

 **Angular** :

Angular je popularan framework za izgradnju korisničkih interfejsa u web aplikacijama. U prekršajnom pravosuđu, Angular se često koristi za razvoj frontend aplikacija koje omogućavaju sudijama, advokatima i drugim korisnicima pristup informacijama o sudskim predmetima, pregled presuda, upravljanje dokumentacijom i komunikaciju sa strankama.

 **Go**:

Go je programski jezik koji je poznat po svojoj jednostavnosti, efikasnosti i brzini izvršavanja. U prekršajnom pravosuđu, Go se često koristi za backend aplikacija zbog svoje sposobnosti da efikasno upravlja velikim brojem zahteva, što je ključno za aplikacije koje zahtevaju brzu i pouzdanu obradu podataka, poput evidencije prekršaja, vođenja sudskih predmeta i generisanja izveštaja.

 **Mikroservisna arhitektura**:  
Mikroservisna arhitektura je pristup u razvoju softvera koji se fokusira na izgradnju aplikacija kao skupa manjih, samostalnih servisa, koji rade zajedno preko mreže. U prekršajnom pravosuđu, ova arhitektura omogućava modularnost i skalabilnost, što je ključno za efikasno upravljanje velikim brojem sudskih predmeta i brzo donošenje odluka.

 **Single sign-on (SSO)**:  
SSO je tehnika autentifikacije koja omogućava korisnicima da pristupe različitim aplikacijama sa samo jednim setom pristupnih podataka. U kontekstu prekršajnog pravosuđa, SSO poboljšava korisničko iskustvo smanjujući potrebu za višestrukim prijavljivanjem, dok istovremeno pruža visok nivo sigurnosti i kontrole pristupa informacionim sistemima.

 **UML dijagrami (Unified Modeling Language)**:  
UML je standardizovani jezik za vizualizaciju, specifikaciju, konstrukciju i dokumentovanje softverskih sistema. U prekršajnom pravosuđu, UML dijagrami se često koriste za modeliranje sudskih procesa, određivanje odnosa između entiteta i definisanje funkcionalnih zahteva, što olakšava razumevanje i implementaciju informacionih sistema.

 **Docker**:

Docker je platforma koja omogućava programerima da pakiraju aplikacije i sve njihove zavisnosti u kontejnere, čime olakšava njihovu distribuciju i pokretanje na različitim računarskim okruženjima, što je korisno za efikasno upravljanje aplikacijama u prekršajnom pravosuđu.

# **Specifikacija zahteva**

Specifikacija zahteva je ključni dokument koji detaljno opisuje funkcionalne i nefunkcionalne zahteve koje softverski sistem ili aplikacija treba da ispuni. Evo nekoliko ključnih elemenata specifikacije zahteva:

## Slučajevi korišćenja:

|  |  |
| --- | --- |
| Naziv | Zakazivanje ročista |
| Preduslov | Sudija je pristupio sistemu |
| Koraci | 1.Sudija popunjava podatke o ročištu  2. Sudija potvrđuje unos  3.Sistem proverava podatke o ročištu  4. Sistem smešta podatek o ročištu |
| Rezultat | Ročište je zakazano |

Tabela 1 – Opis slučaja korišćenja “Zakazivanje ročišta”

|  |  |
| --- | --- |
| Naziv | Odluka o prekršaju |
| Preduslov | Sudija je pristupio sistemu. Postupak prekrsaja j ezavršen, dokazi su prikupljeni I dostavljeni sudu. |
| Koraci | 1.Sudija analizira predmet  2. Sudija donosi odluku o prekršaju  3.Sudija izadje odluku I sankcije  4. Sudija zapisuje odluku |
| Rezultat | Sudija donosi pravednu odluku o prekršaju na temelju analize samog prekršaja I primenjenog prava |

Tabela 2 – Opis slučaja korišćenja “Odluka o prekršaju”

|  |  |
| --- | --- |
| Naziv | Promena statusa predmeta |
| Preduslov | Sudija je pristupio sistemu. |
| Koraci | 1.Sudija pronalazi predmet koji želi da ažurira  2. Sudija pristupa detaljima predmeta  3. Sudija odabira opciju da izmeni status prekrsaja |
| Rezultat | Status predmeta je uspešno promenjen |

Tabela 3 – Opis slučaja korišćenja “Promena statusa predmeta”

|  |  |
| --- | --- |
| Naziv | Pregled obavestenja o zakazanom ročištu |
| Preduslov | Korisnik je pristupio sistemu. |
| Koraci | 1.Korisnik odlazi na svoj email nalog  2. Korisnik vrši pregled detalja zakazanog ročišta |
| Rezultat | Korisnik je upućen o vremenu održavanja zakazanog ročišta |

Tabela 4 – Opis slučaja korišćenja “Pregled obaveštenja o zakazanom ročištu”

|  |  |
| --- | --- |
| Naziv | Obaveštavanje sudije o zakazanom ročištu I njegovim detaljima |
| Preduslov | Sistem je funkcionalan |
| Koraci | 1.Primanje I obrada svih podataka o zakazanom ročištu  2. Slanje obaveštenja sudiji koji je zaveden kao sudija na tom ročištu |
| Rezultat | Uspesno poslato obaveštenje o zakazanom ročištu, kao I njegovim detaljima |

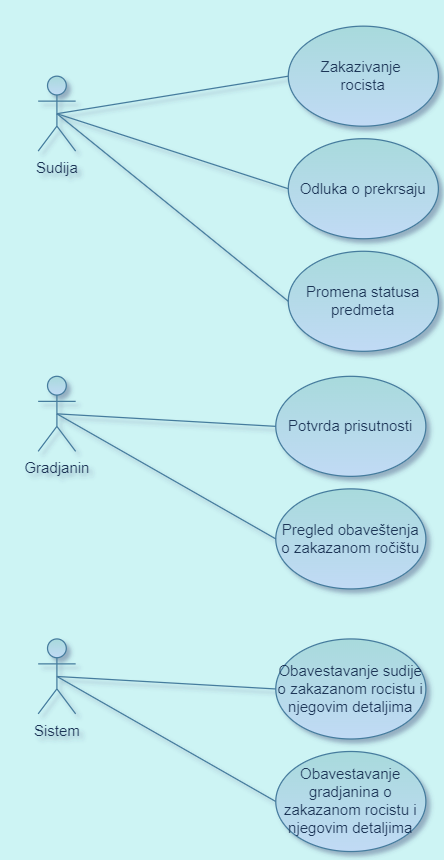
Tabela 5 – Opis slučaja korišćenja “Obaveštavanje sudije o zakazanom ročištu I njegovim detaljima”

|  |  |
| --- | --- |
| Naziv | Obaveštavanje gradjanina o zakazanom ročištu I njegovim detaljima |
| Preduslov | Sistem je funkcionalan |
| Koraci | 1.Primanje I obrada svih podataka o zakazanom ročištu  2. Slanje obaveštenja gradjaninu koji je zaveden kao optuženi na tom ročištu |
| Rezultat | Uspesno poslato obaveštenje o zakazanom ročištu, kao I njegovim detaljima |

Tabela 6 – Opis slučaja korišćenja “Obaveštavanje korisnika o zakazanom ročištu I njegovim detaljima”

|  |  |
| --- | --- |
| Naziv | Potvrda prisutnosti |
| Preduslov | Gradjanin je pristupio sistemu I dobio obaveštenje o zakazanom ročištu |
| Koraci | 1.Korisnik ulazi u obaveštenje I detalje o ročištu  2. Korisnik potvrđuje da će biti prisutan na zakazanom ročištu |
| Rezultat | Uspešna potvrda prisustva na zakazanom ročištu |

Tabela 7 – Opis slučaja korišćenja “Potvrda prisutnosti”



Slika 1 – UML dijagram slučaja korišćenja

## **Specifikacija nefunkcionalnih zahteva**

Nefunkcionalni zahtevi su zahtevi koji definišu kako sistem treba da funkcioniše u smislu performansi, sigurnosti, pouzdanosti, upotrebljivosti i drugih aspekata, ali ne definišu specifične funkcionalnosti sistema. Ovi zahtevi su često vezani za kvalitet sistema ili usluge koji se pruža, a ne za specifične operacije ili funkcije koje sistem obavlja. Evo nekoliko ključnih nefunkcionalnih zahteva:

 **Kontejnerizacija**

* Svi mikroservisi i baze podataka treba da budu pokrenuti kao Docker kontejneri, koristeći Docker Compose alat za upravljanje njihovim konfiguracijama

 **Otpornost na parcijalne otkaze sistema**

* Sistem treba da bude dizajniran tako da u slučaju da jedan servis postane nedostupan, svi ostali servisi nastave da funkcionišu normalno i podrže funkcionalnosti koje nisu direktno zavisne od tog servisa.

 **Upotrebljivost**

* Sistem mora da obezbedi jednostavan i intuitivan korisnički interfejs koji olakšava korisnicima upotrebu bez potrebe za detaljnim obukama ili uputstvima.

# **Specifikacija dizajna**

## Arhitektura sistema

Sistem prekršajnog suda će biti dizajniran kao mikroservisna arhitektura radi bolje modularnosti, skalabilnosti i održavanja. Svaki mikroservis će obavljati specifične funkcionalnosti kao što su upravljanje predmetima, korisnicima, odlukama sudova, itd. Arhitektura će se sastojati od:

* **Backend servisi**: Implementirani u Java Spring Boot, koji će pružati logiku poslovne logike i interakciju sa bazom podataka.
* **Frontend aplikacija**: Razvijena u Angular-u, pružajući korisnicima interfejs za pregled, upravljanje i administraciju sudskih predmeta.
* **Baza podataka**: Relacijska baza podataka (npr. MySQL) za skladištenje informacija o predmetima, korisnicima, odlukama i drugim entitetima.

## Dizajn baze podataka

Struktura baze podataka će biti pažljivo dizajnirana kako bi podržala efikasno skladištenje i upravljanje podacima relevantnim za prekršajni sud. Uključuje:

* **Entiteti**: Poput sudskih predmeta, korisnika, odluka, itd.
* **Veze i ograničenja**: Definisane veze između entiteta i ograničenja integriteta podataka radi očuvanja konzistentnosti i tačnosti informacija.

## . Korisnički interfejs

Korisnički interfejs će biti razvijen sa fokusom na intuitivnost, jednostavnost korišćenja i efikasnost. Elementi korisničkog interfejsa će uključivati:

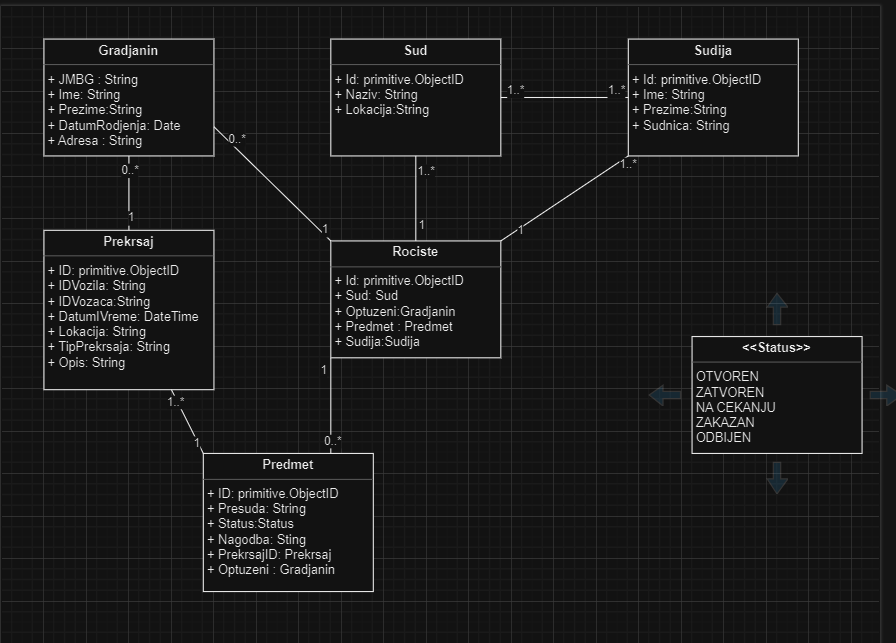
* **Pregled predmeta**: Mogućnost pregleda detalja o predmetima, njihovih statusa i relevantnih informacija.
* **Administracija korisnika**: Mogućnost upravljanja korisničkim nalozima, pravima pristupa i bezbednošću.

## . Bezbednosni dizajn

Bezbednost sistema će biti ključni aspekt dizajna, uključujući:

* **Autentifikacija i autorizacija**: Implementacija sigurnih mehanizama prijavljivanja korisnika i dodeljivanja pristupa na osnovu uloga.
* **Enkripcija podataka**: Zaštita podataka u pokretu i mirovanju radi sprečavanja neovlašćenog pristupa.

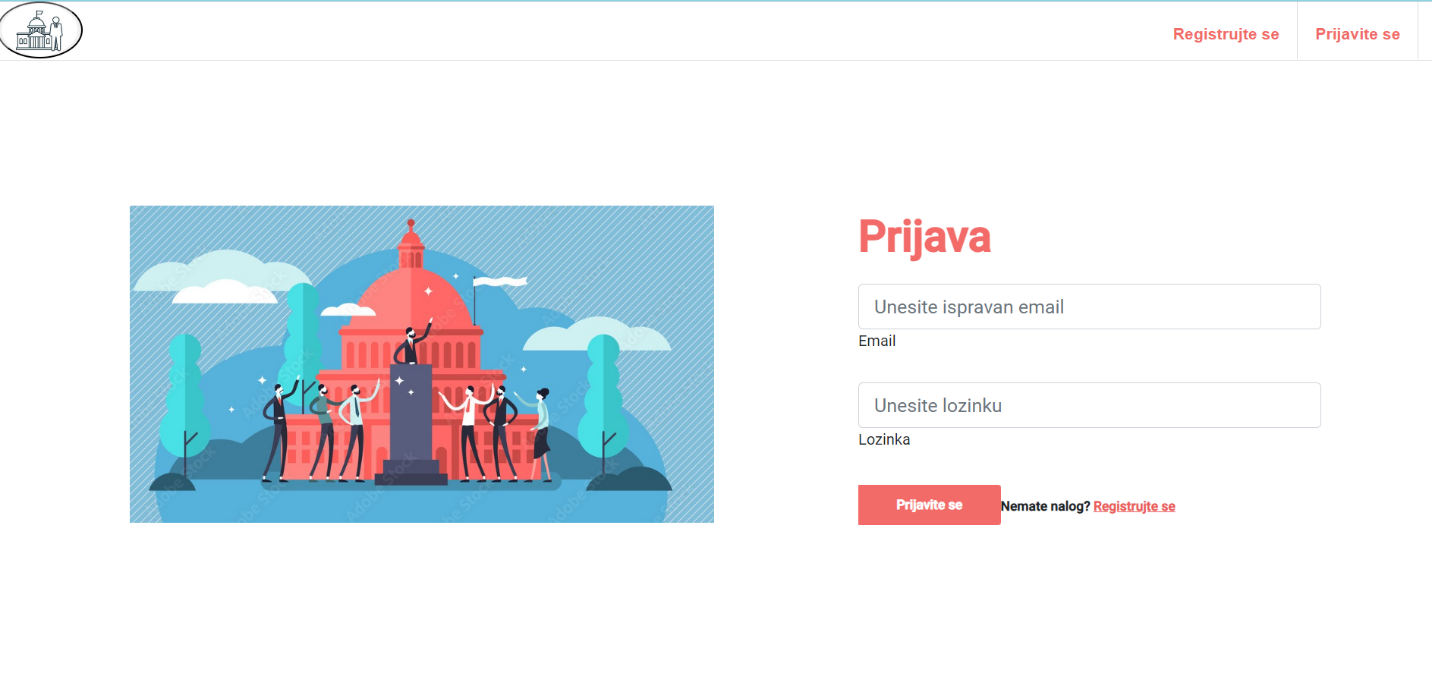
Dijagram klase sistema prekršajnog suda prikazuje strukturu entiteta i njihovih međusobnih veza, što omogućava detaljan uvid u podatke koji se čuvaju i upravljaju u sistemu.



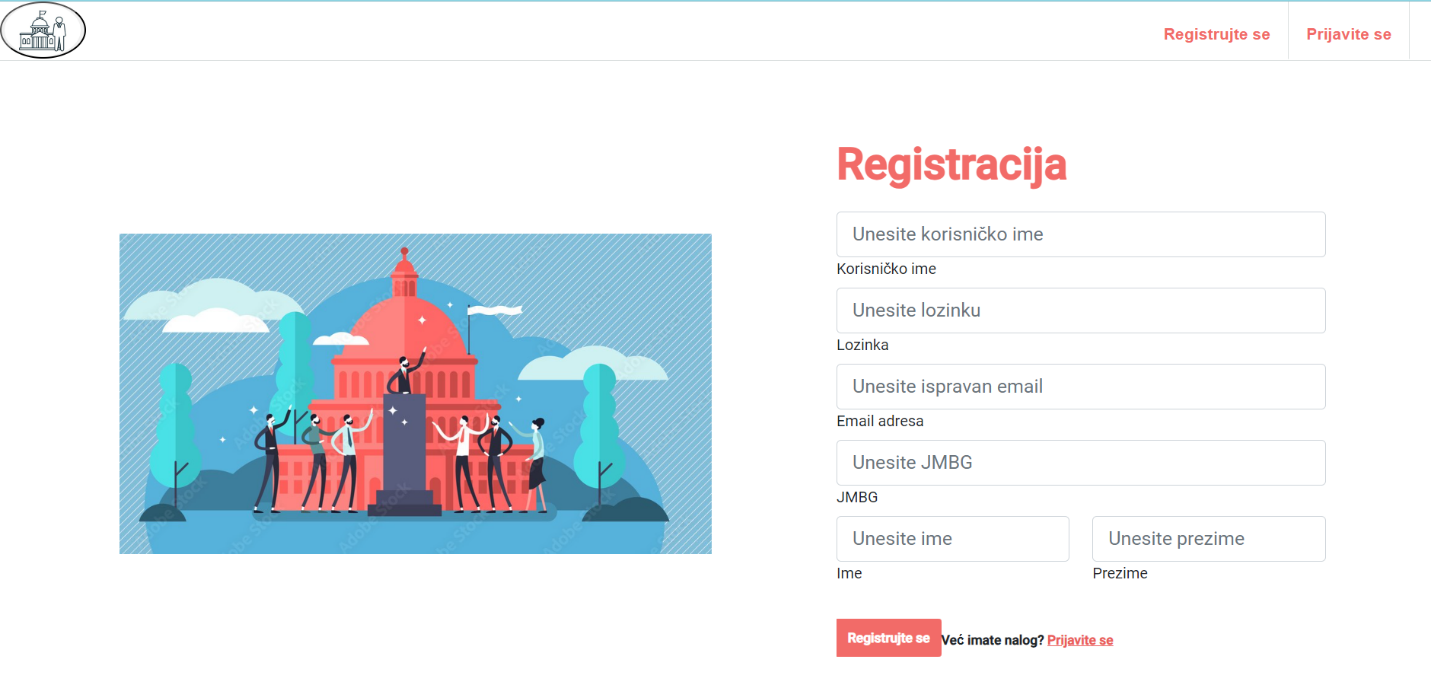
Slika 2 – UML dijagram klase

# **Demonstracija**

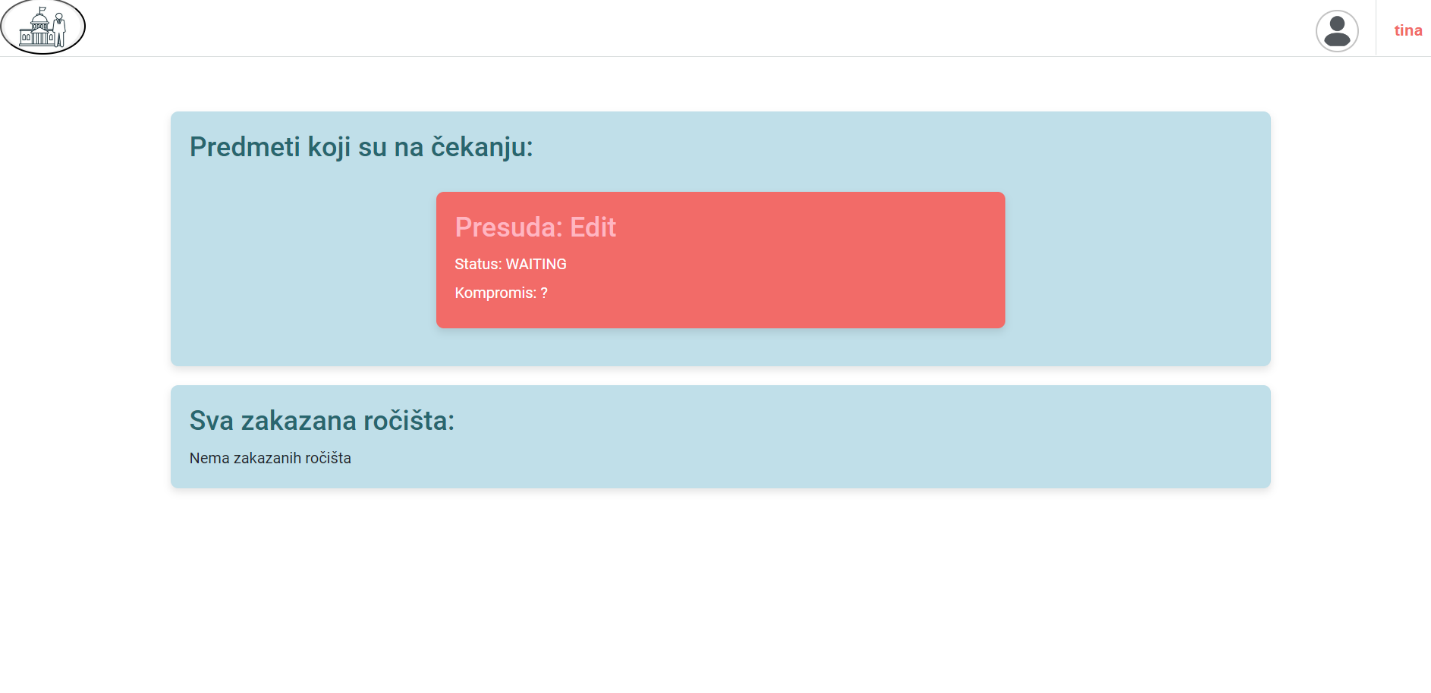
Sledi vizuelni prikaz sadžaja sistema prekršajnog suda :



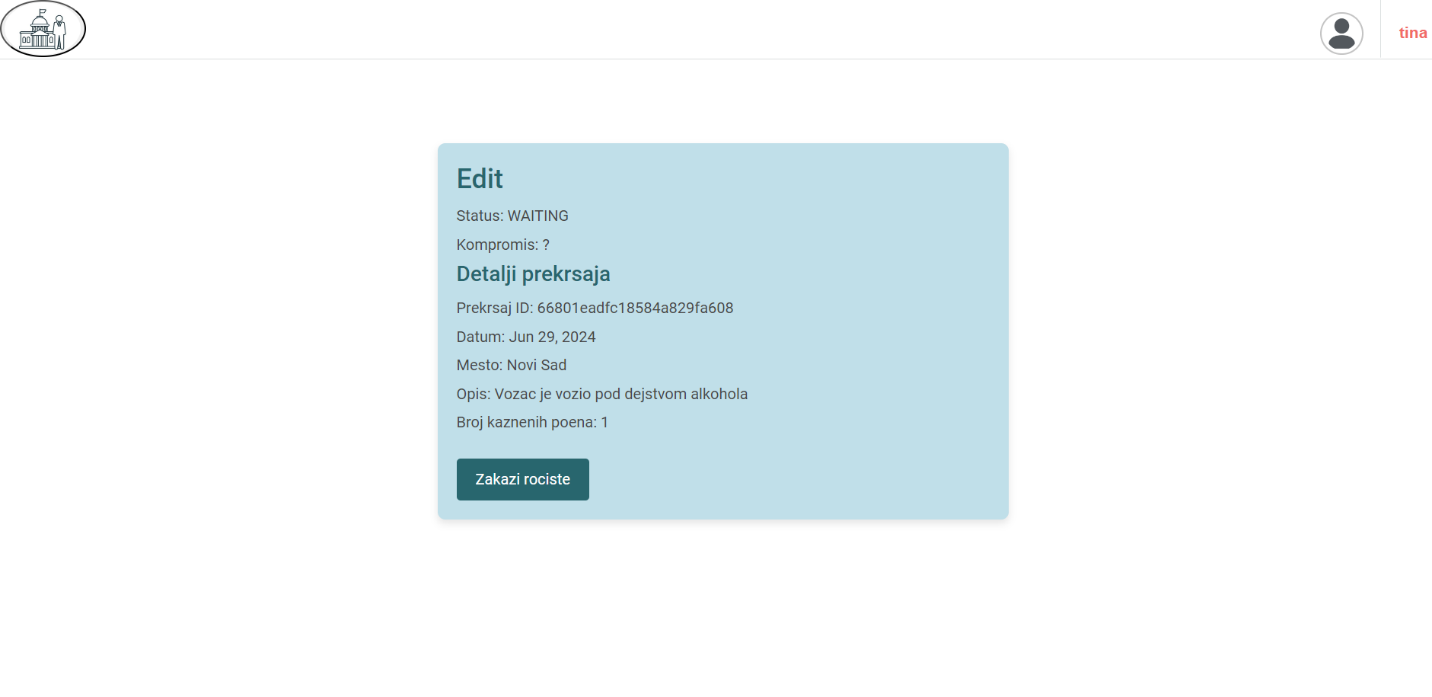
Slika 3 – Prijava na sistem



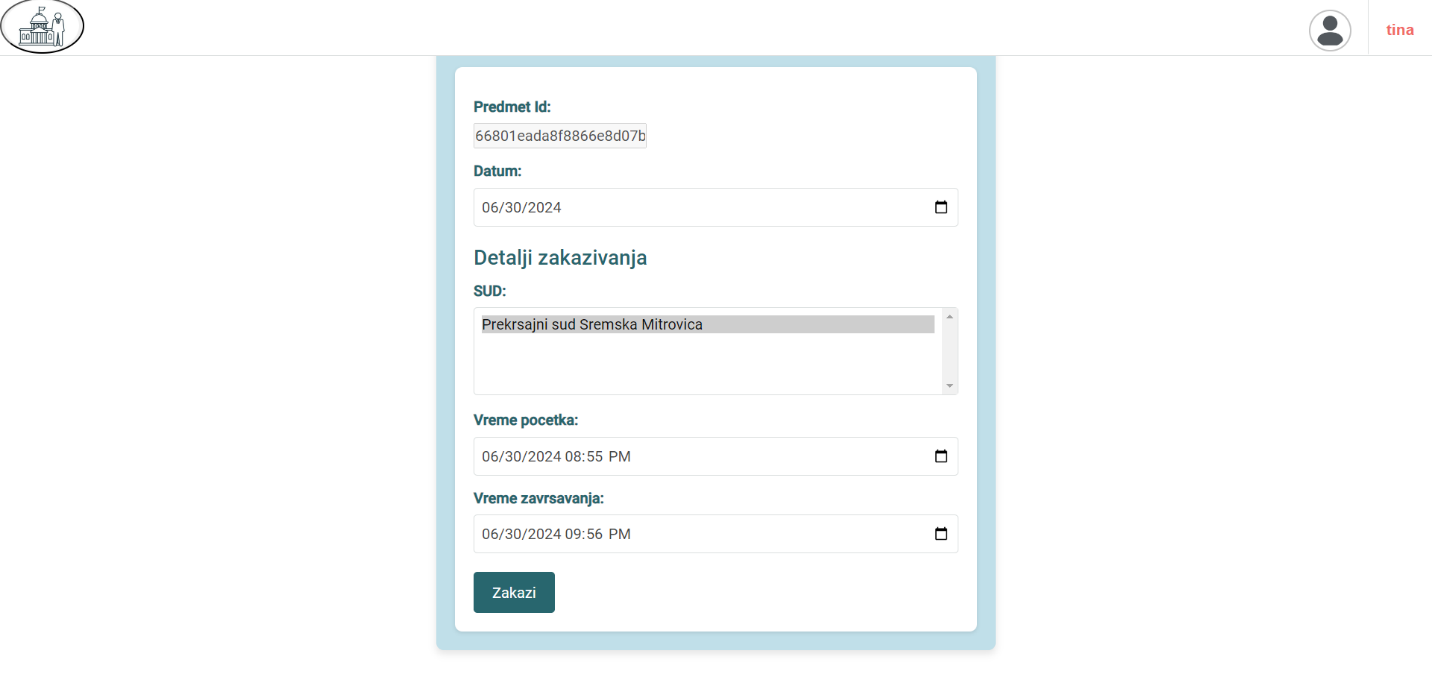
Slika 4 – Registracija korisnika na sistem



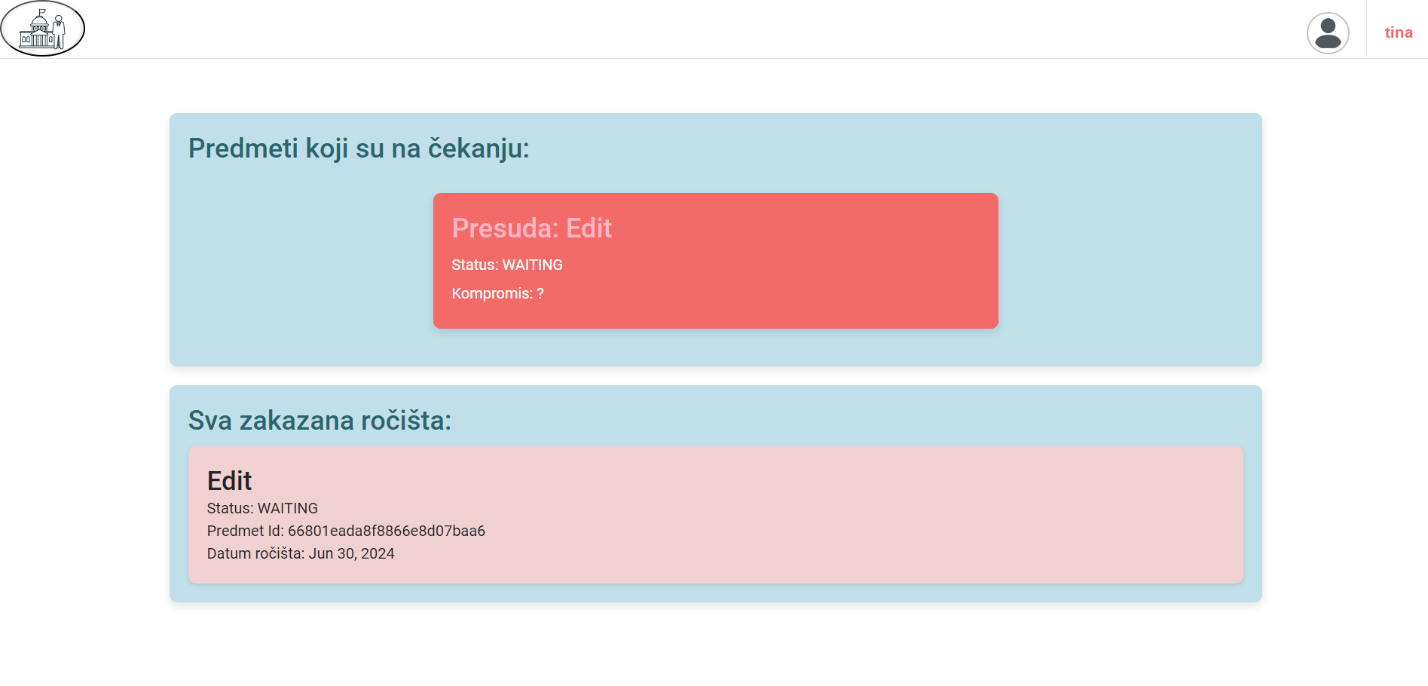
Slika 4 – Pilikom prijave sudije na sistem, prikaz postojecih predmeta



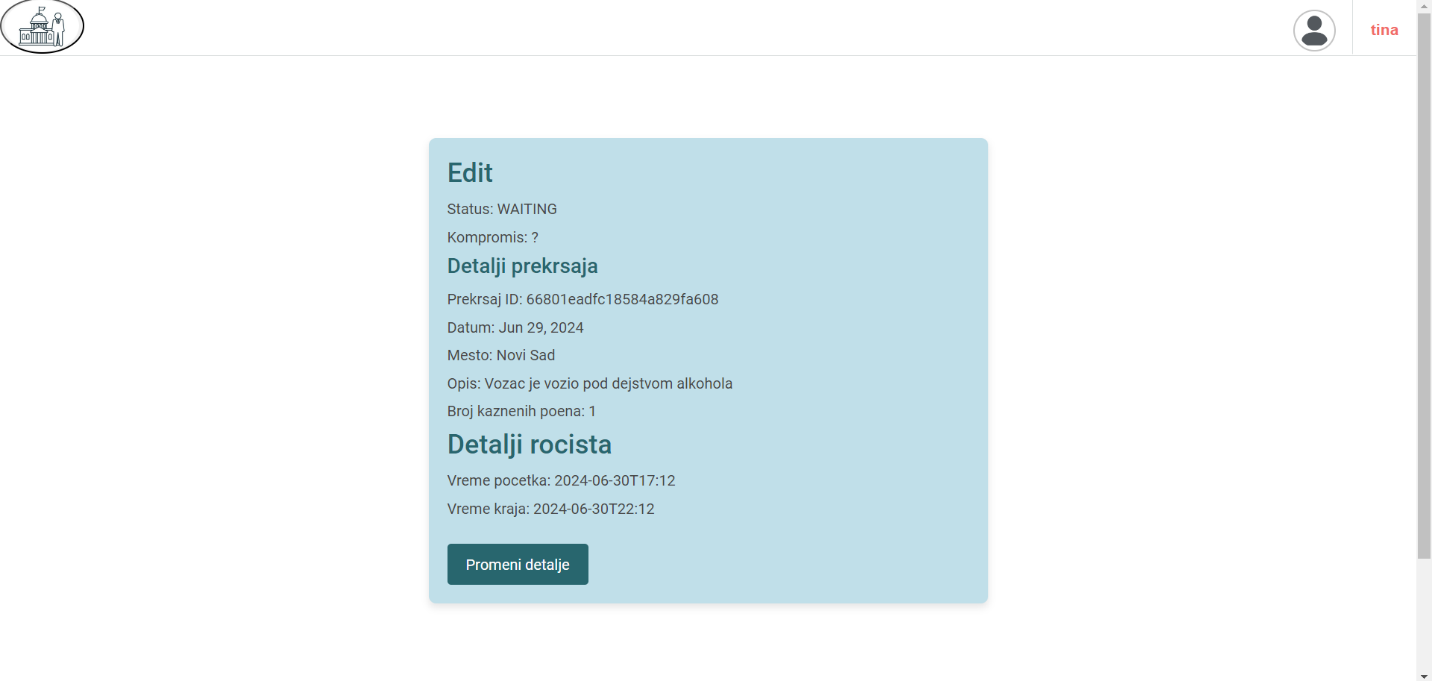
Slika 5 – Prikaz detalja predmeta



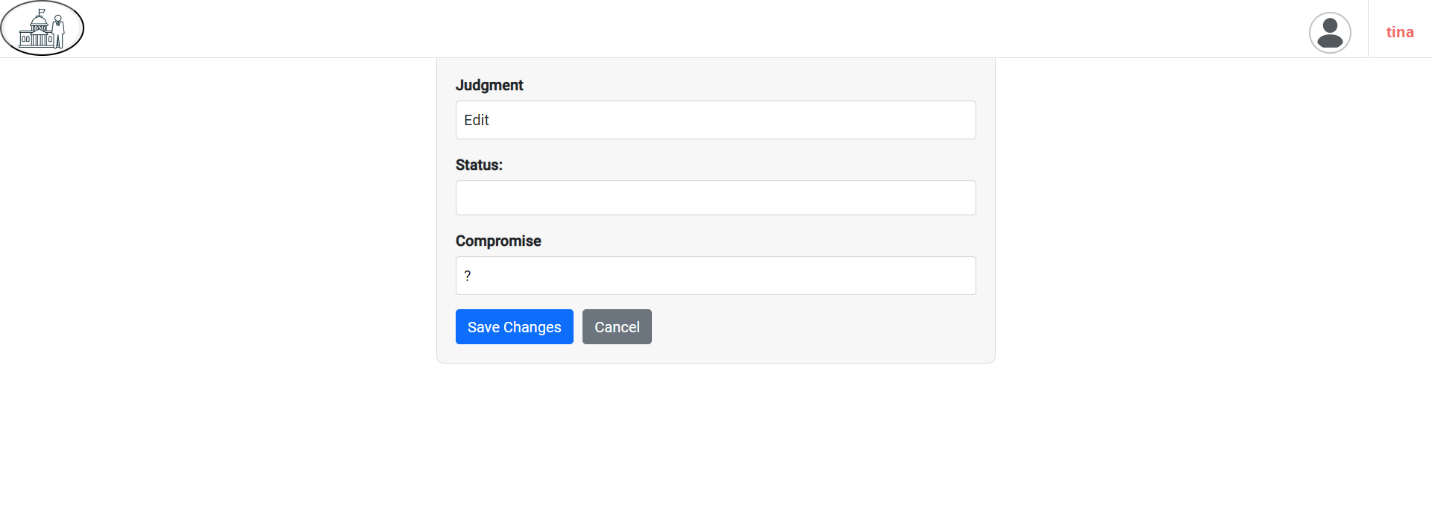
Slika 6 – Forma za zakazivanje ročišta za odabrani predmet



Slika 7 – Prikaz zakazanog ročišta na početnoj stranici



Slika 8 – Prikaz detalja zakazanog ročišta



Slika 8 – Forma za izmenu ročišta (donošenje odluke, promena statusa ili omogucavanje kompromisa)