Sistem za elektronsko podnošenje zahteva i upravljanje dokumentima u MUP-u

# Sažetak

U ovom radu prikazano je rešenje za elektronsko podnošenje i obradu zahteva građana u okviru Ministarstva unutrašnjih poslova (MUP). Sistem omogućava kreiranje korisničkog naloga, prijavu preko centralizovanog sistema autentikacije, podnošenje zahteva za lična dokumenta (lična karta, pasoš, državljanstvo, vozačka dozvola), zakazivanje termina i elektronsku uplatu taksi. Službenici MUP-a imaju mogućnost obrade podataka, dok sistem omogućava i razmenu statističkih podataka sa Zavodom za statistiku. Na ovaj način unapređuje se efikasnost administracije, smanjuju redovi u filijalama i povećava transparentnost procesa.

# Ključne reči

MUP, elektronska uprava, podnošenje zahteva, lična dokumenta, SSO, plaćanje taksi

# Uvod

Digitalizacija javne uprave podrazumeva sve veću dostupnost elektronskih servisa građanima. Jedna od oblasti u kojoj su takvi servisi od posebne važnosti jeste rad Ministarstva unutrašnjih poslova (MUP).  
  
MUP izdaje lična dokumenta, evidentira prekršaje i sarađuje sa drugim institucijama.   
Tradicionalno, ali I dalje u slučaju velikog broja država, proces podnošenje zahteva za dokumenta zahteva odlazak u filijalu, čekanje u redovima i manuelnu proveru dokumentacije.  
  
U ovom radu predstavljen je moderan informacioni sistem koji omogućava elektronsko podnošenje zahteva, zakazivanje termina, elektronsku uplatu taksi i praćenje statusa obrade.  
  
 Sistem omogućava I proširenja poput integracije sa Zavodom za statistiku u cilju razmene podataka koji mogu kasnije služiti za razne analitike koje bi koristile narodu I institucijama.

# Srodna rešenja i korišćene tehnologije

Jedno od najpoznatijih srodnih rešenja u Republici Srbiji jeste portal eUprava, koji građanima omogućava zakazivanje termina u MUP-u za ličnu kartu, pasoš ili vozačku dozvolu, kao i plaćanje taksi putem interneta. Prednost portala eUprava je široka dostupnost i centralizovan pristup različitim uslugama. Međutim, mana je složenost sistema, kao i to što deo procesa i dalje zahteva odlazak u filijalu i manuelnu proveru podataka od strane službenika.

U zemljama Evropske unije mogu se pronaći naprednija rešenja, poput Estonskog eID sistema, koji građanima omogućava potpunu elektronsku identifikaciju, digitalni potpis i pristup svim državnim servisima bez potrebe za fizičkim dolaskom. Prednost ovog sistema je visoka bezbednost i potpun digitalni pristup, dok je mana zahtevna infrastruktura i visoki troškovi implementacije.

Za razliku od navedenih rešenja, razvijeni MUP sistem je fokusiran upravo na procese vezane za Ministarstvo unutrašnjih poslova – podnošenje zahteva, zakazivanje termina, plaćanje taksi i razmenu podataka sa Zavodom za statistiku. Na ovaj način se postiže manja složenost, brža implementacija i prilagođenost lokalnim potrebama.

Tehnologije koje su korišćene:

**NestJS [1]** je okvir za razvoj serverskih aplikacija zasnovan na Node.js platformi. Pogodan je za mikroservisnu arhitekturu i nudi modularan pristup izgradnji aplikacija.

**Prisma ORM [2]** je alat koji omogućava tipizovan i jednostavan rad sa relacionim bazama podataka. Omogućava migracije i generisanje koda što olakšava rad programerima.

**PostgreSQL [3]** je relaciona baza podataka otvorenog koda, poznata po stabilnosti i pouzdanosti u radu sa velikim količinama podataka.

**React [4]** je biblioteka za razvoj korisničkih interfejsa, koja omogućava kreiranje interaktivnih i efikasnih web aplikacija.

**API Gateway [5]** služi za centralizovano usmeravanje zahteva u mikroservisnim arhitekturama, pruža sigurnost i olakšava integraciju različitih servisa.

**Docker [6]** omogućava kontejnerizaciju aplikacija, što olakšava prenosivost i održavanje sistema u različitim okruženjima.

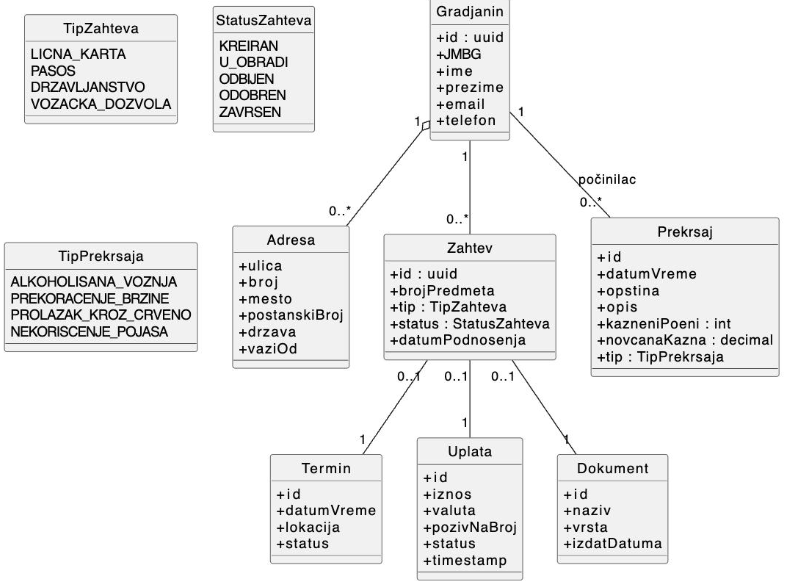
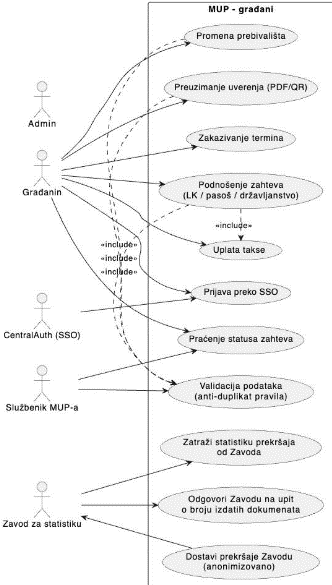
Specifikacija zahteva

**Organizacija podataka u okviru sistema prikazana je na UML dijagramu klasa (slika 1).** Na dijagramu su predstavljene glavne klase poput Građanin, Zahtev, Dokument, Prekršaj i Uplata, kao i njihove međusobne veze. Ovako modelovan sistem omogućava strukturirano čuvanje informacija o podnetim zahtevima, izdavanju dokumenata i evidenciji prekršaja.

Slika 1 – Dijagram klasa

**Funkcionalni zahtevi sistema prikazani su UML dijagramom slučajeva korišćenja (slika 2).** Na dijagramu su predstavljeni osnovni akteri sistema: građanin, administrator, službenik MUP-a, CentralAuth (SSO) i Zavod za statistiku. Građanin ima mogućnost kreiranja naloga, prijave, podnošenja zahteva i praćenja statusa, administrator upravlja sistemom i procesima, službenik MUP-a obrađuje podnete zahteve i vrši validaciju, dok Zavod za statistiku ostvaruje razmenu podataka u vidu statistike i agregiranih informacija.

Slika 2 – UML Use case dijagram

  
  
  
Slika 1: Dijagram klasa   
  
Slika 2: Use case dijagram s

|  |  |
| --- | --- |
| Naziv | Podnošeje zahteva za lični dokument |
| Učesnici | Građanin, Službenik MUP-a |
| Preduslovi | Građanin popunjavava potrebne podatke I prilaže informacije za registraciju |
| Koraci | Građanin potvrđuje zahtev I vrši elektronsku uplatu takse |
| Rezultat | Sistem evidentira zahtev |
| Izuzeci | Ako nedostaju validacioni podaci, onemogućava se podnošenje |

Implementacija  
  
Ovo poglavlje prikazuje način na koji su implementirane osnovne funkcionalnosti sistema Ministarstva unutrašnjih poslova. Fokus je na ključnim delovima implementacije koji se odnose na podnošenje zahteva za izdavanje dokumenata, praćenje statusa i generisanje izveštaja.

### Podnošenje zahteva za dokument

U okviru MUP servisa implementirana je funkcionalnost elektronskog podnošenja zahteva za lična dokumenta. Građanin preko React forme bira tip dokumenta (lična karta, pasoš, državljanstvo, vozačka dozvola), unosi potrebne podatke i podnosi zahtev.  
Na slici 3 prikazana je struktura kontrolera Request Controller koji preko API gateway-a prima zahtev i prosleđuje ga servisu.  
  
Slika 3. 

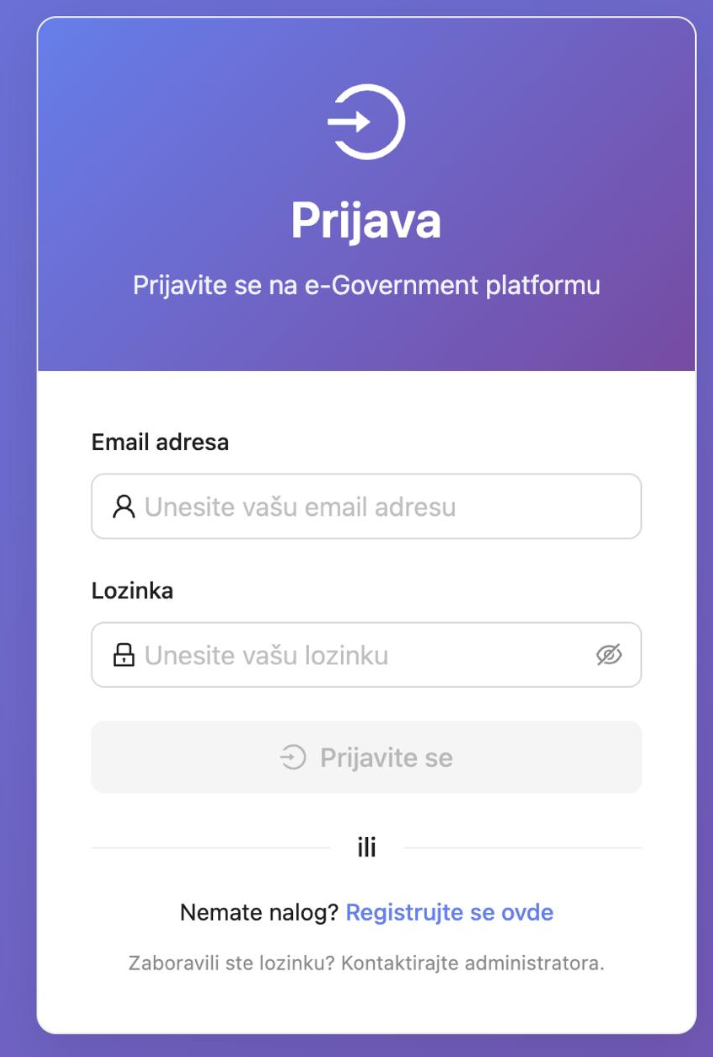
Nakon podnošenja, zahtev se beleži u bazi podataka, gde dobija status KREIRAN.

Na slici 4 prikazana je SQL tabela u kojoj se čuvaju podaci o zahtevima, sa atributima poput tipa dokumenta, datuma podnošenja i statusa.  
Takođe, u tabelu je kao strani ključ naveden građanin na koga se ovaj zahtev odnosi.

### Slika 4.

Demonstracija

U ovom poglavlju prikazan je primer korišćenja sistema Ministarstva unutrašnjih poslova kroz tipičan scenario podnošenja zahteva za lični dokument.

1. Građanin otvara aplikaciju i prijavljuje se putem **SSO login** forme (slika 5).
2. Nakon uspešne prijave, građanin bira opciju za podnošenje zahteva za ličnu kartu, pasoš, državljanstvo ili vozačku dozvolu.
3. Građanin popunjava formu sa potrebnim podacima i potvrđuje unos.
4. Sistem beleži zahtev u bazi podataka, dodeljuje mu jedinstveni broj predmeta.
5. Građanin preko aplikacije može da izvrši uplatu takse i da prati status svog zahteva (u obradi, odobren, odbijen, završen).
6. Nakon obrade od strane službenika MUP-a, građanin dobija obaveštenje i može da preuzme potvrdu ili uverenje.
7. Podaci o izdatim dokumentima postaju deo zbirnog seta i dostupni su za razmenu sa Zavodom za statistiku radi daljih analiza.   
     
   Slika 5 – Forma za prijavu ili registraciju korisnika

# Zaključak

U ovom radu prikazano je softversko rešenje za elektronsko podnošenje zahteva za lične dokumente, zakazivanje termina i praćenje statusa zahteva u okviru MUP servisa. Sistem omogućava građanima da jednostavno pristupe uslugama putem SSO prijave, čime se povećava dostupnost i transparentnost procesa. Pored toga, servis obezbeđuje automatizovanu razmenu podataka sa Zavodom za statistiku, što doprinosi tačnosti i efikasnosti državnih evidencija.

Moguća unapređenja sistema uključuju dodatne mehanizme za elektronsku verifikaciju korisnika, unapređenu bezbednost I sigurnost podataka, kao i proširene funkcionalnosti za vizuelizaciju statistike o izdavanju dokumenata i registrovanim prekršajima kako bi korisnici imali jedinstveni portal na kom mogu da pronađu korisne informacije vezane za njih.

# Literatura

[1] NestJS. 2023. Documentation. Preuzeto sa: <https://nestjs.com>  
[2] Prisma. 2023. Docs. Preuzeto sa: <https://prisma.io>  
[3] PostgreSQL Global Development Group. 2023. PostgreSQL Documentation. Preuzeto sa: <https://www.postgresql.org/docs>  
[4] React. 2023. Official Documentation. Preuzeto sa: <https://react.dev>  
[5] API Gateway. 2023. Concepts and Documentation. Preuzeto sa: https://microservices.io/patterns/apigateway.html  
[6] Docker. 2023. Documentation. Preuzeto sa: <https://docs.docker.com>