

Aplikacija za zakazivanje lekarskih pregleda i izdavanje lekarskih uverenja

Maja Birmančević

Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Trg Dositeja Obradovića 6, 21000 Novi Sad

birmancevic.sr8.2020@uns.ac.rs

Sažetak: Procedura useljenja studenata u studentske domove sastoji se iz nekoliko koraka. Od studenata se očekuje da ispune sve predviđene korake. Jedan od takvih koraka jesu i njihove posete studentskim ambulantama radi lekarskog pregleda. Kako bi svaki od studenata i stekao svoje uverenje o uspešno urađenom lekarskom pregledu, neophodno je da poseti doktore, u zavisnosti od pola, različitih specijalnosti pomenutih studentskih ambulanti, odnosno poliklinika. Vreme provedeno tokom čekanja studenata u redu je često znatno duže od trajanja celokupnih pregleda, naročito u gradovima gde je samo jedan objekat predviđen za ovu namenu. Sa ovim problemom susreću se studenti koji su ostvarili pravo na korišćenje studentskih domova u Novom Sadu.

Ovaj rad pruža predlog rešenja koje omogućava studentima da zakažu termin po izboru kod izabranog specijaliste, dok bi se doktorima olakšalo izdavanje lekarskih uverenja. Upotrebom ovog rešenja smanjile bi se gužve u studentskim ambulantama pa bi tako studenti provodili manje vremena čekajući u redu.

Ključne reči: veb aplikacija; zakazivanje; studentska ambulanta; lekarsko uverenje; studentski dom;

1. Uvod

Zbog obaveza na fakultetu, studenti su često primorani da zbog dugog čekanja napuste red i dođu ponovo drugi dan. Razlog za dugo čekanje se prevashodno zasniva na činjenici da, do danas, informaciono-komunikacioni sistem prisutan u studentskoj ambulanti, ne nudi mogućnost rezervacije zasebnih vremenskih intervala koji bi se dodelili svakom pojedinačnom studentu. Ipak, poželjno je studentima obezbediti najkraće moguće zadržavanje u studentskoj ambulanti. Prema tome, zadržavanje studenata ne treba da traje duže od trajanja svih predviđenih pregleda, a da bi se to moglo omogućiti, potreban je informacioni sistem koji bi rešio problem viščasovnog čekanja u redu.

U ovom radu je prikazan predlog rešenja koji bi u znatnoj meri olakšao i ubrzao proceduru lekarskih pregleda zakazivanjem i drugim načinima evidentiranja pregleda, kao i izdavanjem lekarskih uverenja.

Ostatak rada je organizovan kao što je objašnjeno u nastavku.

U drugom poglavlju je dat pregled srodnih istraživanja i tehnologije koje su korišćene u aplikaciji.

U trećem poglavlju su navedeni funkcionalni zahtevi koje ispunjava ovo rešenje, dok je u četvrtom poglavlju predstavljen njegov dizajn.

U petom poglavlju prikazan je način implementacije rešenja, dok je u šestom poglavlju objašnjeno njegovo korišćenje.

Sedmo poglavlje donosi zaključna razmatranja i pravce budućih istraživanja.

2. Srodna istraživanja

U ovom poglavlju je dat pregled postojećih rešenja za zakazivanje termina i navedene su tehnologije koje omogućavaju da se reši problem stvaranja dugih redova u čekaonici.

2.1. Srodna rešenja

Prisutnost informacionih sistema koji omogućavaju zakazivanje termina je sve češća u mnogim oblastima rada, te se kao rezultat njihove primene može primetiti smanjeno vreme čekanja u redu. Na portalu e-Uprave Republike Srbije [1] nalazi se usluga zakazivanja termina za izdavanje lične karte ili pasoša, gde je potrebno izabrati jedan od ponuđenih termina, kao što je prikazano na slici 1.

Назив услуге: еЗаказивање - лична карта и/или пасош - Град Шабац		ПУ Шабац 1 - Личне карте и пасоши							ПУ Шабац 2 - Личне карте и пасоши						
Подносилац захтева: МАЈА БИРМАНЧЕВИЋ		19.	20.	21.	22.	23.	24.	25.	19.	20.	21.	22.	23.	24.	25.
Изабрани термин 22.06.2023 Од: 08:45 До: 09:00		8 00	8:00- 8:30 Заузе терми	8:00- 8:45 Заузе терми	8:00- 8:30 Заузе терми	8:00- 8:15 Заузе	8:00- 8:15 Заузе		8:00- 8:30 Заузе терми	8:00- 8:15 Заузе		8:00- 8:30 Заузе терми			
		15			8:15- 8:30 Заузе		8:15- 8:30 Заузе		8:15- 8:45 Заузе терми	8:15- 8:45 Заузе терми					
		30	8:30- 8:45 Заузе		8:30- 8:00 Заузе	8:30- 8:00 Заузе	8:30- 8:45 Заузе		8:30- 8:45 Заузе				8:30- 8:45 Заузе		
		45	8:45- 8:00 Заузе	8:45- 8:15 Заузе терми	8:45- 8:00 Заузе терми	8:45- 8:00 Заузе			8:45- 8:15 Заузе терми	8:45- 8:00 Заузе терми	8:45- 8:15 Заузе терми				
		9 00	9:00- 8:30 Заузе терми		9:00- 8:15 Заузе	9:00- 8:30 Заузе	9:00- 8:15 Заузе		9:00- 8:30 Заузе терми	9:00- 8:30 Заузе терми					
		15		9:15- 8:30 Заузе	9:15- 8:45 Заузе		9:15- 8:30 Заузе				9:15- 8:45 Заузе терми				
		30	9:30- 10:00 Заузе терми	9:30- 10:00 Заузе		9:30- 8:45 Заузе	9:30- 10:00 Заузе		9:30- 10:00 Заузе	9:30- 8:45 Заузе					
		45		9:45- 10:00 Заузе	9:45- 10:00 Заузе										

Slika 1: Zakazivanje termina za izdavanje lične karte ili pasoša na portalu e-Uprave

2.2. Korišćene tehnologije

Za izradu projekta korišćene su sledeće tehnologije:

- Klijentska aplikacija je implementirana pomoću okruženja (eng. framework) Angular [2]
- Serverska aplikacija je implementirana korišćenjem programskog jezika Java [4], okruženja Spring [8] i Hibernate-a [3], biblioteke za objektno-relaciono mapiranje.
- Za skladištenje podataka korišćena je MySQL [7] baza podataka.

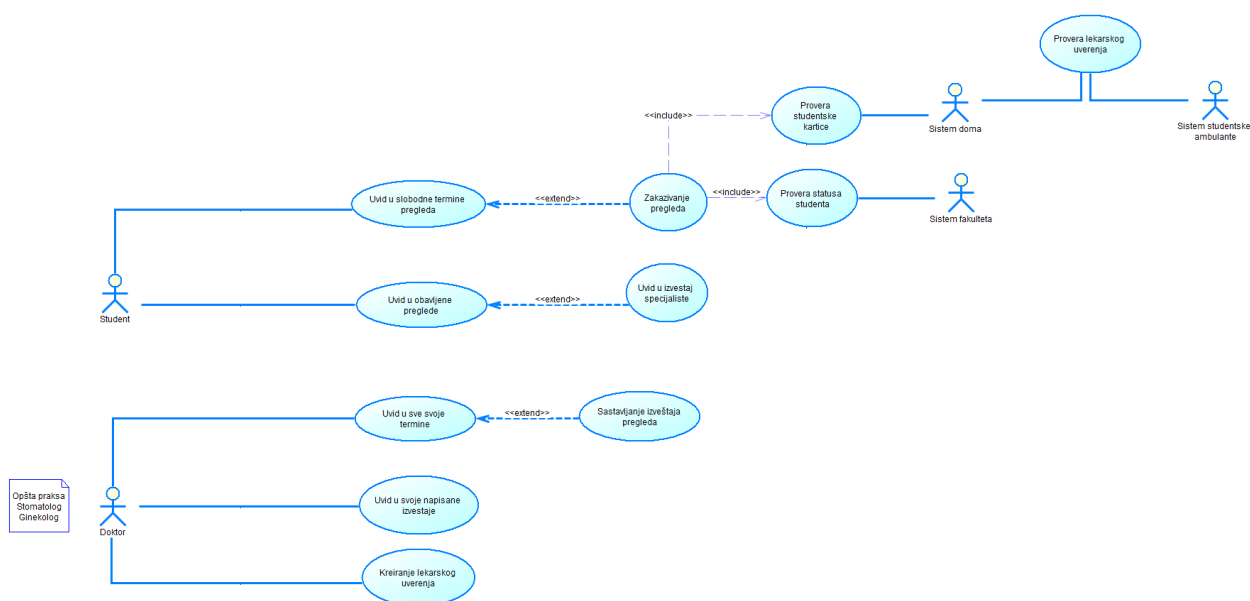
3. Specifikacija zahteva

U ovom odeljku su opisani funkcionalni zahtevi koje je potrebno da ispunjava softversko rešenje za zakazivanje termina lekarskog pregleda.

U sistemu su prisutna dva tipa korisnika: student i doktor. Studenti imaju mogućnost zakazivanja termina za pojedinačne preglede kod specijalista, uvida u obavljene preglede kao i prikaz izveštaja doktora za svaki pregled. Prilikom zakazivanja termina, sistem proverava da li student ispunjava potrebne uslove za obavljanje lekarskog pregleda, odnosno, da li je budžetski student fakulteta i da li ima aktivnu studentsku karticu.

Doktor ima uvid u sve svoje termine, gde prilikom izbora zakazanog termina ima mogućnost kreiranja izveštaja. Ima mogućnost i da pregleda svoje napisane izveštaje kao i da vrši filtriranje nad njima. Doktor opšte prakse izdaje lekarsko uverenje studentima koji su obavili sve potrebne preglede kod specijalista.

Funkcionalni zahtevi ovog softverskog rešenja su predstavljeni UML dijagramom slučajeva korišćenja, kao što je prikazano na slici 2, dok se opisi, u zavisnosti od uloga u sistemu, nalaze na slikama: 3,4 i 5.



Slika 2: Dijagram slučajeva korišćenja

STUDENT

<p>UVID U SLOBODNE TERMINE PREGLEDA</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Student bira lekara(užu specijalnost) čije slobodne termine želi da vidi,kao i datum2. Studentu se prikazuje lista slobodnih termina sa opcijom za zakazivanje. <p>REZULTAT: Student ima uvid u listu slobodnih termina izabranog lekara</p> <p>IZUZECI: -</p>	<p>ZAKAZIVANJE PREGLEDA</p> <p>PREDUSLOV: Student je na budžetu i ima aktivnu studentsku karticu</p> <p>KORACI:</p> <ol style="list-style-type: none">1.Student bira slobodan termin odabranog lekara2. Student potvrđuje zakazivanje termina (npr. klikom na dugme "Zakaži")3. Terminu se menja status iz "slobodan" u "zakazan" <p>REZULTAT: Student je zakazao termin kod izabranog lekara</p> <p>IZUZECI: -</p>	<p>PROVERA STUDENTSKE KARTICE</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Sistem studentske ambulante šalje zahtev sistemu studentskog doma da proverí da li je kartica aktivna2. Sistem studentske ambulante dobije odgovor od sistema fakulteta u vidu "boolean" vrednosti. <p>REZULTAT: Izvršena provera studentske kartice</p> <p>IZUZECI: -</p>
<p>UVID U OBAVLJENE PREGLEDE</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Student bira opciju za uvid u istoriju obavljenih pregleda2. Studentu se prikazuje lista pregleda koje je obavio <p>REZULTAT: Student ima uvid u listu obavljenih pregleda</p> <p>IZUZECI: -</p>	<p>UVID U IZVEŠTAJ SPECIJALISTE</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Student bira opciju za uvid u izveštaj pregleda2. Studentu se prikazuju podaci iz izveštaja obavljenog pregleda <p>REZULTAT: Student ima uvid u izveštaj specijaliste</p> <p>IZUZECI: -</p>	<p>PROVERA STATUSA STUDENTA</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Sistem studentske ambulante šalje zahtev sistemu fakulteta da proverí da li je student na budžetu.2. Sistem studentske ambulante dobije odgovor od sistema fakulteta u vidu "boolean" vrednosti. <p>REZULTAT: Izvršena provera statusa studenta.</p> <p>IZUZECI: -</p>

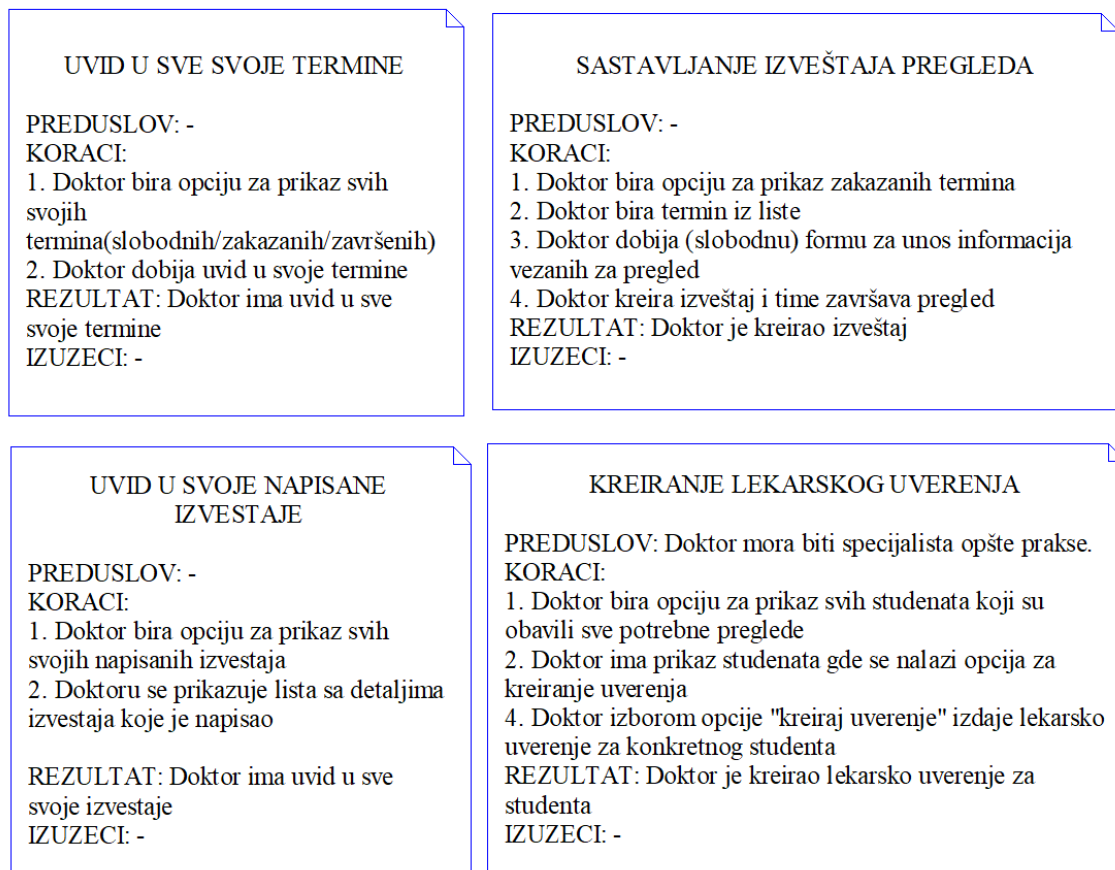
Slika 3: Opis slučajeva korišćenja za ulogu studenta

STUDENSKI DOM I AMBULANTA

<p>PROVERA LEKARSKOG UVERENJA</p> <p>PREDUSLOV: -</p> <p>KORACI:</p> <ol style="list-style-type: none">1. Sistem doma šalje zahtev sistemu studentske ambulate da proverí da li postoji lekarsko uverenje za određenog studenta.2. Sistem studentske ambulante proverava da li se u bazi nalazi uverenje za traženog studenta.3. Sistem vraća odgovor u vidu boolean vrednosti, 'true' ako postoji uverenje, 'false' ako ne postoji. <p>REZULTAT: Sistem studentske ambulante proverio da li student ima lekarsko uverenje.</p> <p>IZUZECI: -</p>

Slika 4: Opis slučajeva korišćenja za uloge sistema studentskog doma i ambulante

DOKTOR



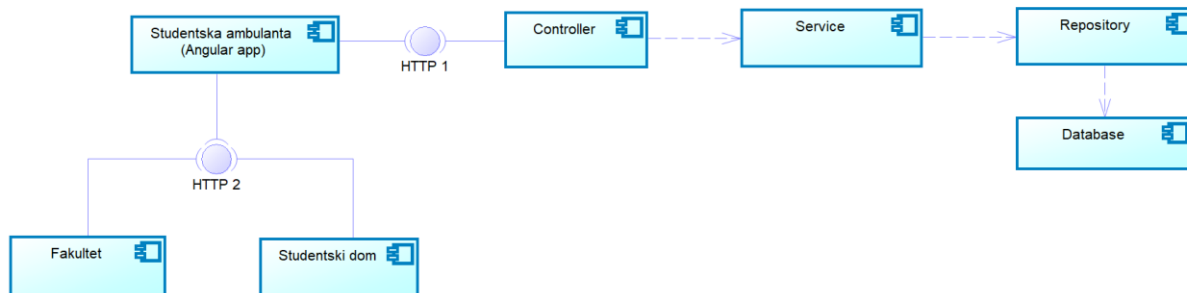
Slika 5: Opis slučajeva korišćenja za ulogu doktora

4. Specifikacija dizajna

Ovo poglavlje objašnjava dizajn softverskog rešenja za zakazivanje termina lekarskog pregleda.

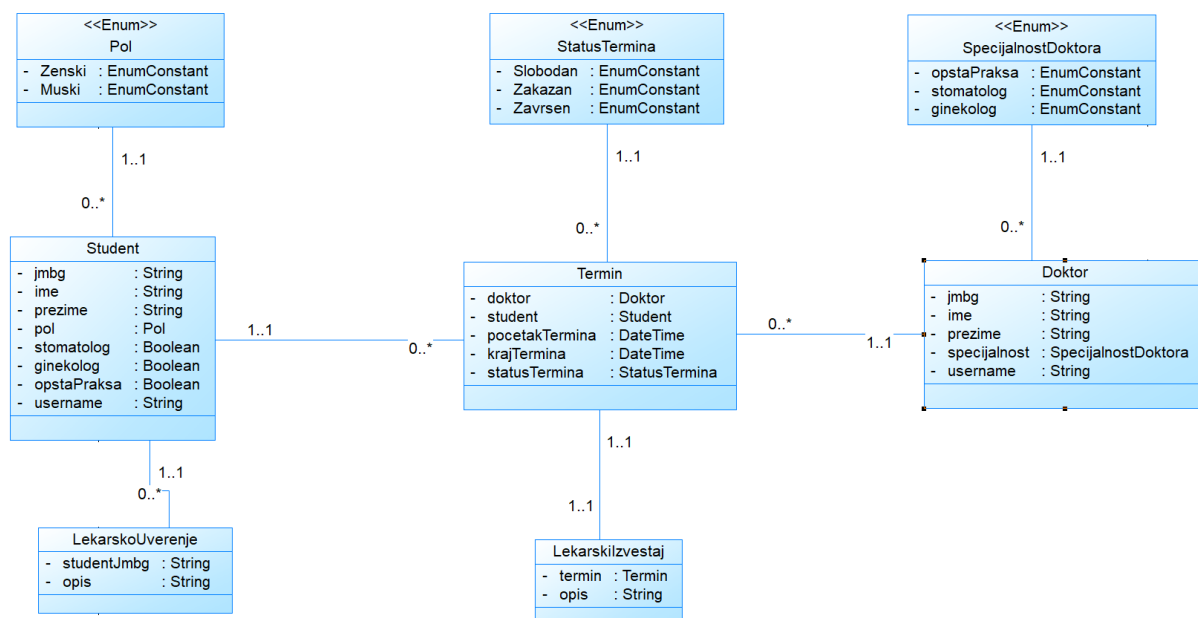
Rešenje predstavlja veb aplikaciju gde klijentska strana komunicira sa serverskom (koju čine Controller, Service i Repository), pomoću HTTP API-ja koji je dizajniran u skladu sa REST [6] arhitektonskim stilom. Takođe, na isti način se vrši komunikacija sa sistemima fakulteta i studentskog doma, gde se prosleđuje JMBG studenta i na osnovu njega vrši provera da li je student na budžetu (sistem fakulteta) i da li ima aktivnu studentsku karticu (sistem studentskog doma). U zavisnosti od uloge koju korisnik ima, na osnovu JWT (JSON Web Token) [5] koji se šalje u zaglavlju zahteva, moguć je pristup različitim API endpoint-ima u Controller-u (Controller komponenta). Komponenta Controller prosleđuje zahtev odgovarajućoj metodi servisnog sloja (Service komponenta) u kojoj se nalazi logika sistema i neke od provera koje utiču na dalji rad sa bazom. Komponenta Repository sadrži metode koje predstavljaju zasebne upite ka bazi.

Na slici 6 prikazana je arhitektura sistema.



Slika 6: Arhitektura sistema

Objektni model podataka prikazan na slici 7 predstavljen je dijagramom klasa.



Slika 7: Dijagram klasa

Klase „Student“ i „Doktor“ predstavljaju korisnike u sistemu i sadrže njihove lične podatke.

Termin za zakazivanje modelovan je klasom „Termin“. Sadrži sve potrebne informacije koje su važne za jedan termin, kao što su vreme početka i završetka termina, koji doktor pregleda i koji student je zakazao termin.

Izveštaj koji doktori sastavljaju tokom pregleda je modelovan klasom „Lekarskilzvestaj“, dok klasa „LekarskoUverenje“ predstavlja potvrdu koju doktor opšte prakse izdaje studentima koji su obavili sve potrebne preglede.

Enumeracija „StatusTermina“ je status koji termin može da ima, u zavisnosti od toga da li je zakazan, slobodan ili završen, dok je pol korisnika predstavljen enumeracijom „Pol“.

Specijalnost doktora je modelovana enumeracijom „SpecijalnostDoktora“.

5. Implementacija

Ovo poglavlje prikazuje način na koji su implementirane neke od funkcija sistema za zakazivanje termina lekarskog pregleda.

5.1. Implementacija klijentske aplikacije

Za kreiranje izveštaja lekarskog pregleda na klijentskom aplikaciji implementirana metoda prikazana je na listingu 1.

```
export class DodavanjeIzvestajaComponent implements OnInit {  
    idTermina!: number;  
    dodavanjeIzvestaja! : FormGroup;  
    request!:IzvestajRequest;  
  
    constructor(private router:Router,private activeRouter:ActivatedRoute,  
        private izvestajService:IzvestajiService) {}  
  
    this.idTermina = this.activeRouter.snapshot.params['id'];  
  
    this.dodavanjeIzvestaja = new FormGroup({  
        opis : new FormControl('')  
    })  
  
    this.request = {  
        opis : '',  
        terminId : this.idTermina  
    }  
  
    ngOnInit(): void {}  
  
    kreirajIzvestaj(){  
        this.request.opis = this.dodavanjeIzvestaja.get('opis')?.value;  
        this.request.terminId = this.idTermina;  
        this.izvestajService.postIzvestaj(this.request).subscribe((data) =>{  
            this.router.navigate(['istorija-izvestaja']);  
        })  
    }  
}
```

Listing 1: Kreiranja izveštaja za lekarski pregled na klijentskoj strani

U ovoj metodi se vrši dodela vrednosti atributa klase „IzvestajRequest“, gde se za atribut „opis“ postavlja vrednost iz input polja forme, kao što je označeno crvenom i narandžastom bojom.

Vrednost koja se dodeljuje za „terminId“ uzima se iz putanje pomoću klase „ActivatedRoute“, kao što je označeno zelenom bojom u konstruktoru.

Tako kreiran objekat klase „IzvestajRequest“ se prosleđuje kao parametar metode „postIzvestaj“, u kojoj se poziva odgovarajuća metoda serverske aplikacije za kreiranje izveštaja.

5.2. Implementacija serverske aplikacije

Za kreiranje izveštaja lekarskog pregleda na serverskoj aplikaciji implementirana metoda prikazana je na listingu 2.

```

public Boolean addIzvestaj(LekarskiIzvestajRequest request){

    Termin termin = terminService.findById(request.getTerminId());

    if(termin != null && !termin.getStatusTermina().equals(StatusTermina.ZAVRSEN)) {

        termin.setStatusTermina(StatusTermina.ZAVRSEN);
        setSpecijalnost(termin);
        terminService.save(termin);

        LekarskiIzvestaj izvestaj = new LekarskiIzvestaj();
        izvestaj.setTermin(termin);
        izvestaj.setOpis(request.getOpis());

        LekarskiIzvestaj savedReport = lekarskiIzvestajRepository.save(izvestaj);
        return true;
    }
    return false;
}

```

Listing 2: Kreiranja izveštaja za lekarski pregled na serverskoj strani

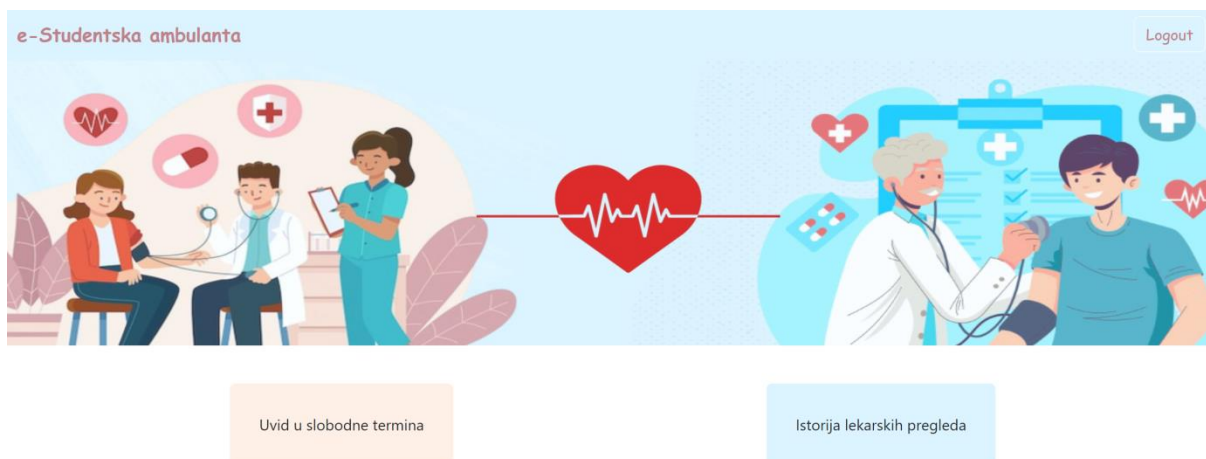
Ovoj metodi se prosleđuje referenca na objekat koji reprezentuje zahtev za kreiranje lekarskog izveštaja. Ukoliko postoji termin na osnovu identifikatora, koji se nalazi u zahtevu, menjaju se njegove vrednosti atributa i kreira se objekat koji predstavlja lekarski izveštaj. Termin i opis iz zahteva se dodeljuju kao vrednosti atributa objekta lekarskog izveštaja. U zavisnosti od datog uslova, metoda će imati povratnu vrednost Boolean, odnosno “true” ako je uslov ispunjen, u suprotnom “false”.

6. Demonstracija

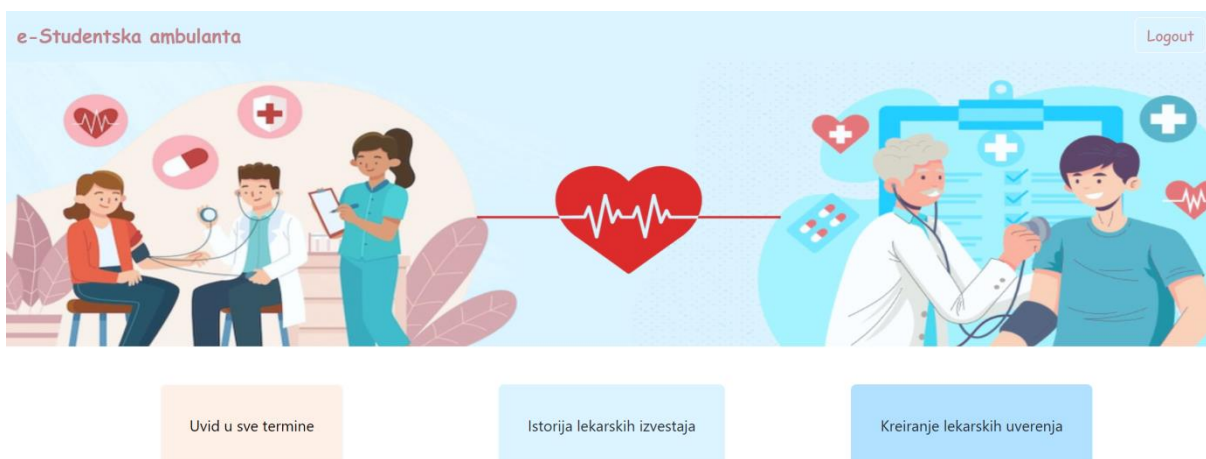
Ovo poglavlje prikazuje način korišćenja aplikacije za zakazivanje termina lekarskih pregleda.

Nakon prijavljivanja na sistem, u zavisnosti od uloge korisnika, prikazuje se početna strana sa funkcionalnostima koje su dostupne za tu ulogu.

Početna strana koja se prikazuje studentu predstavljena je na slici 8, dok je za ulogu doktora početna strana prikazana na slici 9. Doktori opšte prakse imaju dodatnu funkcionalnost za razliku od ostalih specijalista, te se kartica za kreiranje lekarskih uverenja prikazuje ukoliko prijavljen korisnik ima ulogu doktora opšte prakse.



Slika 8: Izgled početne strane za studente



Slika 9: Izgled početne strane za doktore

Studentu se izborom kartice „Uvid u slobodne termine“ prikazuje panel za zakazivanje gde je potrebno izabrati datum i specijalistu, kao što je prikazano na slici 10 dok uvid u obavljene preglede dobija izborom opcije „Istorija lekarskih pregleda“ kao što je prikazano na slici 11.

e-Studentska ambulanta
Logout

Izaberi datum
Izaberi specijalistu

2023-06-24

Stomatolog
Stomatolog
Opsta Praksa
Ginekolog

Pretrazi

Pocetak termina	Kraj termina	Specijalnost
24-06-2023 16:00	24-06-2023 17:45	STOMATOLOG
24-06-2023 17:30	24-06-2023 17:45	STOMATOLOG

Zakazi
Nazad

Slika 10: Panel za zakazivanje termina

e-Studentska ambulanta				Logout
Istorija obavljenih pregleda				
Pocetak termina	Kraj termina	Ime doktora	Specijalnost	
24-06-2023 16:00	24-06-2023 16:15	Natasa Djurica	STOMATOLOG	Pregled izvestaja
← Nazad				

Slika 11: Prikaz istorije obavljenih pregleda studenta

Doktor ima uvid u sve svoje termine kao na slici 12. Izborom zakazanog termina ima mogućnost kreiranja izveštaja.

e-Studentska ambulanta				Logout
Pregled svih termina				
Pretražite termine..				
Student	Jmbg studenta	Pocetak termina	Kraj termina	Status termina
-	-	19-06-2023 16:30	19-06-2023 16:45	SLOBODAN
Maja Birmancevic	3322114433441	24-06-2023 16:00	24-06-2023 16:15	ZAKAZAN
Dodaj izvestaj				
-	-	24-06-2023 17:30	24-06-2023 17:45	SLOBODAN

Slika 12: Prikaz svih termina doktora

Doktor opšte prakse ima mogućnost kreiranja lekarskih uverenja za studente koji su obavili sve potrebne preglede izborom opcije „Kreiraj uverenje“, kao što se može videti na slici 13.

e-Studentska ambulanta				Logout
Pregled studenata koji su obavili preglede				
Student	Jmbg studenta	Sva uverenja		
Maja Birmancevic	3322114433441	Kreiraj uverenje		

Slika 13: Prikaz studenata koji su obavili sve potrebne preglede za izdavanje uverenja

7. Zaključak

U radu je predstavljeno softversko rešenje za zakazivanje termina lekarskih pregleda kao i izdavanje lekarskih uverenja za studente koji su ostvarili pravo na korišćenje studentskih domova. Takođe, demonstriran je rad prototipske aplikacije.

Prikazano rešenje olakšava studentima obavljanje lekarskih pregleda koji predstavljaju preduslov za dobijanje lekarskog uverenja, neophodnog za useljenje u studentski dom. Ovo rešenje omogućava studentima da zakažu termine kod specijalista studentske ambulante i tako ubrzaju proces obavljanja lekarskog pregleda, dok je doktorima opšte prakse na raspolaganju mogućnost izdavanja lekarskog uverenja.

Moguća proširenja za ovaj sistem bi bila da se on primenjuje i u svrhu lekarskih pregleda povodom overe semestra na fakultetu, kao i da se omogući ocenjivanje rada medicinskih radnika.

8. Reference

- [1] *Portal eUprava Republike Srbije*, <https://euprava.gov.rs/>
- [2] *Google*. 2016. *Angular*, <https://angular.io/>
- [3] *Red Hat*. 2001. *Hibernate*, <https://hibernate.org/>
- [4] *Sun Microsystems*. 1996. *Java*, <https://www.java.com/>
- [5] *Michael B. Jones, John Bradley, Nat Sakimura*. 2010. *JSON Web Token (JWT)*, <https://datatracker.ietf.org/doc/html/rfc7519>
- [6] *Roy Thomas Fielding*. 2000. *Architectural Styles and the Design of Network-based Software*, <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [7] *David Axmark, Michael "Monty" Widenius*. Oracle. 1995. *MySQL*, <https://www.mysql.com/>
- [8] *VMware*. 2002. *Spring Framework*, <https://spring.io/>