

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5455

# **WEB APLIKACIJA ZA NARUČIVANJE PACIJENATA**

Toni Martinčić

Zagreb, lipanj 2018.

Zagreb, 12. ožujka 2018.

## ZAVRŠNI ZADATAK br. 5455

Pristupnik: **Toni Martinčić (0036492423)**  
Studij: Računarstvo  
Modul: Računarska znanost

Zadatak: **Web aplikacija za naručivanje pacijenata**

### Opis zadatka:


U suvremenim zdravstvenim sustavima, procesi naručivanja pacijenata za preglede oslanjaju se na Web aplikacije čiji su korisnici pacijenti, liječnici opće medicine i liječnici specijalisti. Pacijentima aplikacija treba omogućiti uvid u termine za pregled te naručivanje za pregled. Liječnici opće prakse moraju moći izdavati uputnice, dok specijalistima sustav treba omogućiti pregled postojećih zahtjeva te dodjeljivanje termina za preglede. Također, poželjno je da se pacijentima automatski pošalje email s dodijeljenim terminom za pregled.

Vaš je zadatak da projektirate i implementirate Web aplikaciju i pripadnu bazu podataka. Za implementaciju ćete koristiti Spring tehnologiju i programski jezik Java. Aplikacija treba podržavati izvođenje svih navedenih funkcija. Programsko rješenje i način upotrebe je potrebno dokumentirati.

Zadatak uručen pristupniku: 16. ožujka 2018.

Rok za predaju rada: 15. lipnja 2018.

Mentor:

  
Izv. prof. dr. sc. Mirko Randić

Djelovoda:

  
Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za  
završni rad modula:

  
Prof. dr. sc. Siniša Srblić



# Sadržaj

1.	Uvod.....	1
2.	Arhitektura i dizajn sustava .....	2
2.1.	Zahtjevi na programski sustav .....	2
2.2.	Skica sustava.....	4
2.3.	Dijagram razreda s opisom.....	7
2.3.1.	Paket controller .....	7
2.3.2.	Paket service.....	12
2.3.3.	Paket repository .....	16
2.3.4.	Paket domain .....	17
2.3.5.	Paket enumeration .....	21
3.	Implementacija i korisničko sučelje .....	23
3.1.	Korištene tehnologije i alati.....	23
3.2.	Isječci programskog koda vezani za temeljnu funkcionalnost...	26
3.3.	Korisničko sučelje .....	30
3.3.1.	Početna stranica .....	30
3.3.2.	Korisničko sučelje administratora.....	31
3.3.3.	Korisničko sučelje specijalista obiteljske medicine .....	35
3.3.4.	Korisničko sučelje liječnika specijalista .....	38
3.3.5.	Korisničko sučelje pacijenta .....	40
3.3.6.	Postavke sustava .....	44
4.	Zaključak .....	46
5.	Literatura .....	47
6.	Popis slika .....	48
7.	Naslov, sažetak i ključne riječi na hrvatskom jeziku .....	49

7.1.	Naslov.....	49
7.2.	Sažetak.....	49
7.3.	Ključne riječi.....	49
8.	Naslov, sažetak i ključne riječi na engleskom jeziku .....	50
8.1.	Naslov.....	50
8.2.	Sažetak.....	50
8.3.	Ključne riječi.....	50

# 1. Uvod

Suvremeno društvo vrijeme cijeni kao jedno od najvrjednijih bogatstva. Upravo zbog toga se ušteda vremena koje trošimo na različitim aktivnostima nameće kao potreba. Odlazak liječniku, a posebice naručivanje na preglede pacijentima između ostalog predstavlja utrošak dragocjenog vremena koji bi se pravilno realiziranim idejama mogao smanjiti na minimum. Cilj ovog projekta je modernizirati postupak naručivanja pacijenata na preglede, omogućiti pacijentima i liječnicima pristup jedinstvenom sustavu koji će im to omogućiti na jednostavan i pristupačan način te u konačnici dovesti do smanjenog broja pacijenata u čekaonicama i do uštede svima toliko dragocjenog vremena.

U drugom poglavlju opisani su arhitektura i dizajn sustava. Prikazani su dijagrami razreda s opisima. U trećem poglavlju opisani su implementacija i korisničko sučelje. Navedene i opisane su korištene tehnologije i alati. Prikazani su isječki programskog koda vezani za temeljnu funkcionalnost. U četvrtom poglavlju nalazi se zaključak, u petom popis literature, a u šestom popis slika. Na kraju rada nalaze se naslov, zaključak i ključne riječi na hrvatskom i engleskom jeziku.

## **2. Arhitektura i dizajn sustava**

### **2.1. Zahtjevi na programski sustav**

Cilj projekta je oblikovati i implementirati programski sustav koji će olakšati izdavanje uputnica i upravljanje liječničkim pregledima. Sustav treba biti jednostavan za korištenje i intuitivan. Predviđeno je da sustav koriste četiri tipa korisnika (slika 1): administrator, specijalist obiteljske medicine, liječnik specijalist i pacijent.

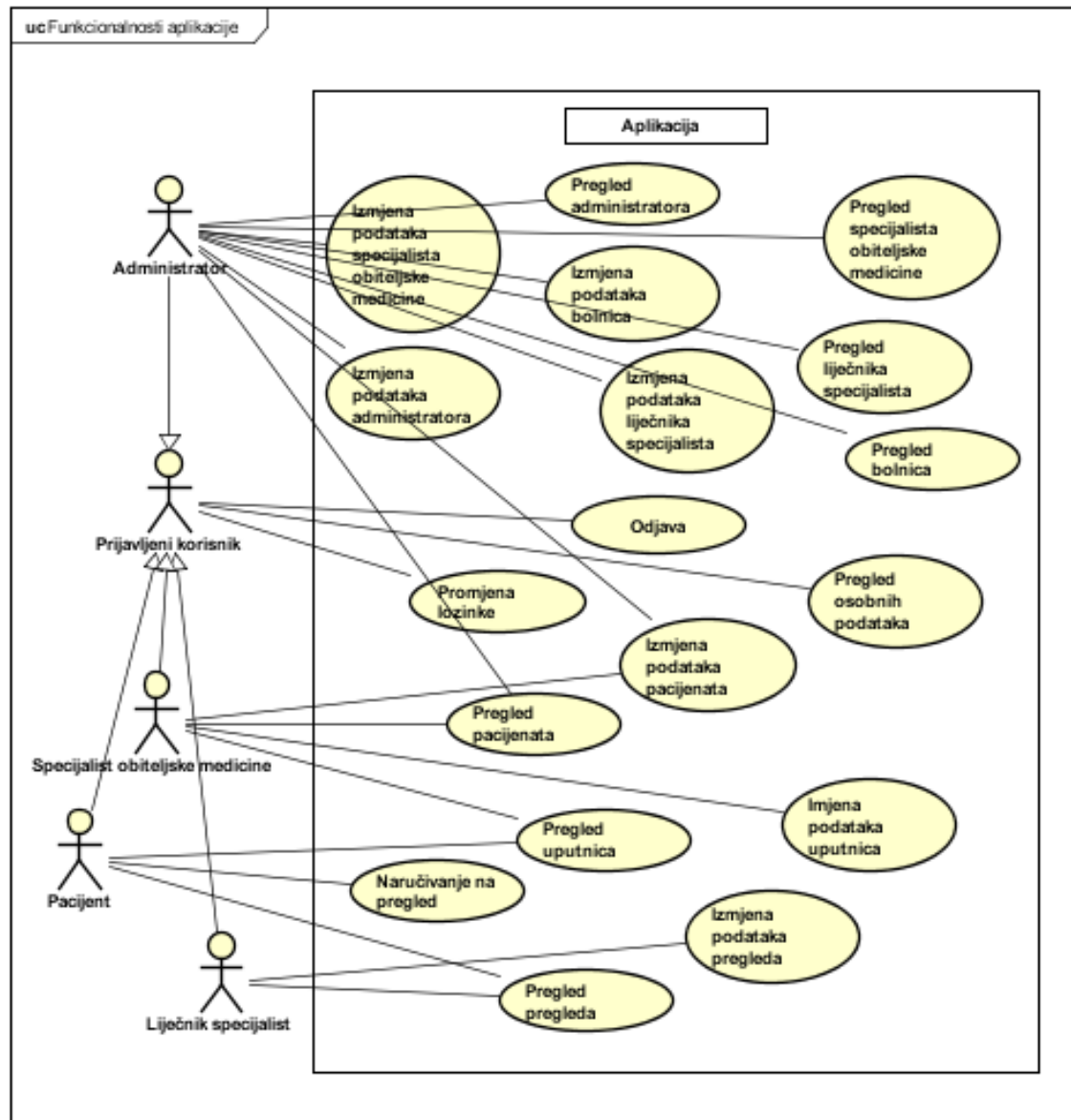
Administrator je zadužen za upravljanje korisnicima i bolnicama. Ima mogućnost dodavanja novih korisnika u sustav, uređivanja korisnika i brisanja korisnika. Također, ima mogućnost dodavanja, uređivanja i brisanja bolnica.

Specijalist obiteljske medicine ima uvid u podatke o svim svojim pacijentima. Ima uvid u sve uputnice koje je izdao svojim pacijentima. Omogućeno mu je uređivanje podataka za pacijente i uputnice.

Liječnik specijalist ima uvid u sve svoje preglede. Određuje termine za buduće preglede.

Pacijent ima uvid u sve svoje uputnice i preglede. Ima mogućnost naručivanja na preglede.

Svaki korisnik treba imati svoje korisničko ime i lozinku pomoću kojih se prijavljuje u sustav.

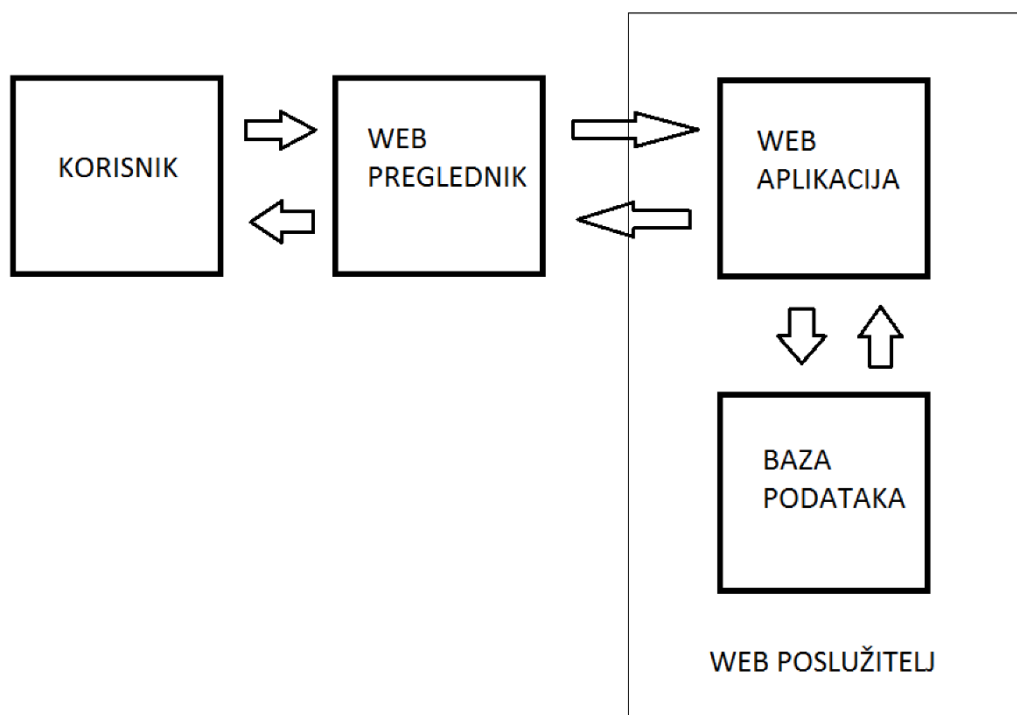


Slika 1. Dijagram obrazaca uporabe



## 2.2. Skica sustava

Sustav predstavlja trorednu Web aplikaciju koja sadrži tri uobičajena podsustava (slika 2): klijentski dio aplikacije, poslužiteljski dio Web aplikacije i bazu podataka.



Slika 2. Skica sustava

### Web poslužitelj

Web poslužitelj prihvaća i obrađuje sve zahtjeve koje dobiva od strane korisnika.. Zahtjevi se šalju HTTP protokolom. Nakon obrade zahtjeva poslužitelj šalje korisniku odgovor HTTP protokolom. Web aplikacija se izvodi na *Apache Tomcat* web poslužitelju.

## Poslužiteljski dio Web aplikacije

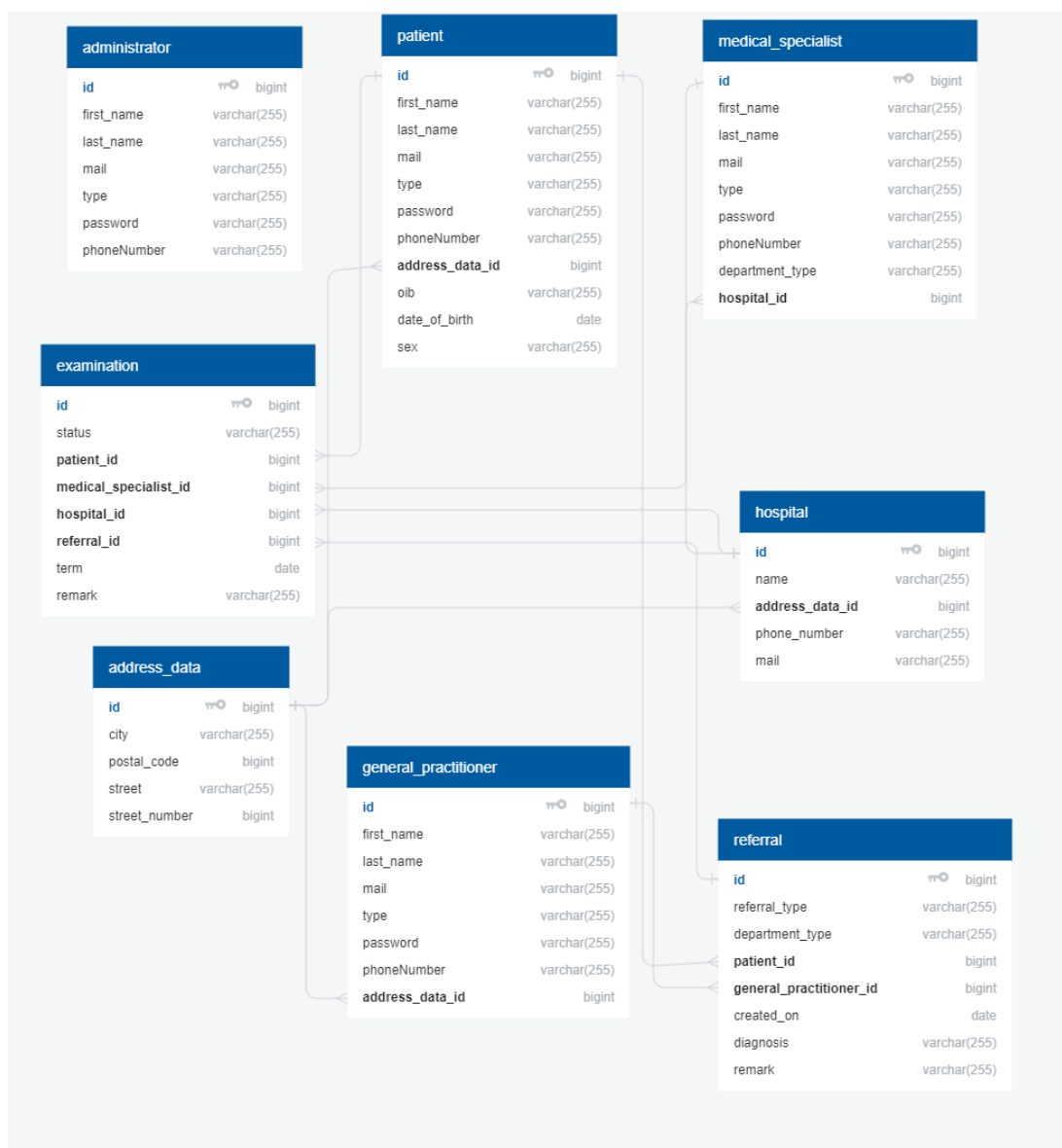
Koristeći web preglednik korisnik pristupa web aplikaciji preko koje ima pristup svim funkcionalnostima sustava ovisno o pridijeljenim ovlastima. Korisnik može pokrenuti različite akcije kojima se šalju zahtjevi na poslužitelj i nakon što poslužitelj obradi zahtjev, korisnik dobiva odgovarajući odgovor na akciju.

Web aplikacija prikazuje sadržaj u obliku HTML dokumenta koje preglednik prikazuje u obliku web stranica.

## Baza podataka

Baza podataka pohranjuje sve podatke o korisnicima i ostalim entitetima u sustavu. Zadaća baze podataka je da brzo i po potrebi dohvaća podatke koje je korisnik zatražio, osvježava podatke kada to korisnik inicira i sprječava dupliciranje podataka tj. onemogućava unos podataka koji se već nalaze u bazi podataka. U projektu je korištena H2 baza podataka koja podatke pohranjuje u radnoj memoriji računala. U nastavku opisujemo entitete baze podataka (Slika 3). **Administrator** - sadrži sve informacije o administratoru aplikacije. To su automatski generirani id, ime, prezime, mail, tip korisnika, lozinka i telefonski broj. Entitet nema vezu prema nijednom drugom entitetu. **GeneralPractitioner** - sadrži informacije o specijalistu obiteljske medicine. To su automatski generirani id, ime, prezime, mail, tip korisnika, lozinka i telefonski broj. Entitet je povezan jednostranom vezom jedan na jedan s entitetom **AddressData**. **MedicalSpecialist** - sadrži informacije o liječniku specijalistu. To su automatski generirani id, ime, prezime, mail, tip korisnika, lozinka i telefonski broj. Entitet je povezan jednostranom vezom više na jedan s entitetom **Hospital**. **Patient** - sadrži informacije o pacijentu. To su automatski generirani id, ime, prezime, mail, tip korisnika, lozinka, telefonski broj, oib, datum rođenja i spol. Entitet je povezan jednostranom vezom jedan na jedan s entitetom **AddressData**. **AddressData** - sadrži informacije o adresi. To su automatski generirani id, grad, poštanski broj, ulica i kućni broj. Entitet nema vezu prema nijednom drugom entitetu. **Examination** - sadrži informacije o pregledu. To su automatski generirani id, termin i bilješke. Entitet ima jednostrane jedan na jedan veze prema entitetima: **Patient**, **MedicalSpecialist**, **Hospital**, **Referral**. **Referral** - sadrži informacije o uputnici. To su automatski generirani id, tip uputnice, bolnički odjel, datum kreiranja, dijagnoza i

napomena. Entitet ima jednostrane jedan na jedan veze prema entitetima **Patient** i **GeneralPractitioner**.



Slika 3. Model baze podataka

## 2.3. Dijagram razreda s opisom

U ovom odlomku opisani su razredi i priloženi su pripadajući dijagrami razreda. Razredi su raspodijeljeni u pet paketa: *controller*, *service*, *repository*, *domain* i *enumeration*. U paketu *controller* nalaze se razredi koji zaprimaju web zahtjeve i pozivaju odgovarajuće razrede iz paketa *service* da obrade zahtjeve, dobiveni odgovor vraćaju korisniku. U paketu *service* nalaze razredi koji su odgovorni za poslovnu logiku. Po potrebi pristupaju bazi podataka preko sučelja u paketu *repository*. U paketu *repository* nalaze se sučelja preko kojih se pristupa tablicama u bazi podataka. U paketu *domain* definirani su svi entiteti aplikacije. U paketu *enumeration* nalaze se sve enumeracije aplikacije.

### 2.3.1. Paket controller

#### ExaminationController

Zaprima korisničke zahtjeve za upravljanje podacima razreda **Examination**. Sadrži primjerak sučelja **ExaminationService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih pregleda za zadani id i tip korisnika, metodu za dohvat pregleda za zadani id, metodu za dodavanje novog pregleda, metodu za uređivanje pregleda, metodu za brisanje pregleda za zadani id i metodu koja dohvaća sve statuse pregleda.

#### HospitalController

Zaprima korisničke zahtjeve za upravljanje podacima razreda **Hospital**. Sadrži primjerak sučelja **HospitalService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih bolnica, metodu za dohvat bolnice za zadani id, metodu za dodavanje nove bolnice, metodu za uređivanje bolnice i metodu za brisanje bolnice za zadani id.

### **ReferralController**

Zaprima korisničke zahtjeve za upravljanje podacima razreda **Referral**. Sadrži primjerak sučelja **ReferralService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih uputnica za zadani id i tip korisnika, metodu za dohvat uputnice za zadani id, metodu za dodavanje nove uputnice, metodu za uređivanje uputnice, metodu za brisanje uputnice za zadani id i metodu za dohvaćanje svih tipova uputnica.

### **UserController**

Sadrži metodu za dohvaćanje korisnika za zadani id i tip korisnika i metodu za promjenu lozinke korisnika.

Sadrži primjerak sučelja **UserService**.

### **LoginController**

Sadrži metodu koja prima korisničko ime i lozinku i provjerava postoji li za primljene podatke korisnik, ako postoji vraća odgovarajućeg korisnika.

Sadrži primjerak sučelja **UserService**.

### **AdministratorController**

Zaprima korisničke zahtjeve za upravljanje podacima razreda **Administrator**. Sadrži primjerak sučelja **AdministratorService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih administratora, metodu za dohvat administratora za zadani id, metodu za dodavanje novog administratora, metodu za uređivanje administratora i metodu za brisanje administratora za zadani id.

### **MedicalSpecialistController**

Zaprima korisničke zahtjeve za upravljanje podacima razreda **MedicalSpecialist**. Sadrži primjerak sučelja **MedicalSpecialistService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih liječnika specijalista, metodu za dohvat liječnika specijalista za zadani id, metodu za dodavanje novog liječnika specijalista, metodu za uređivanje liječnika specijalista i metodu za brisanje liječnika specijalista za zadani id.

### **PatientController**

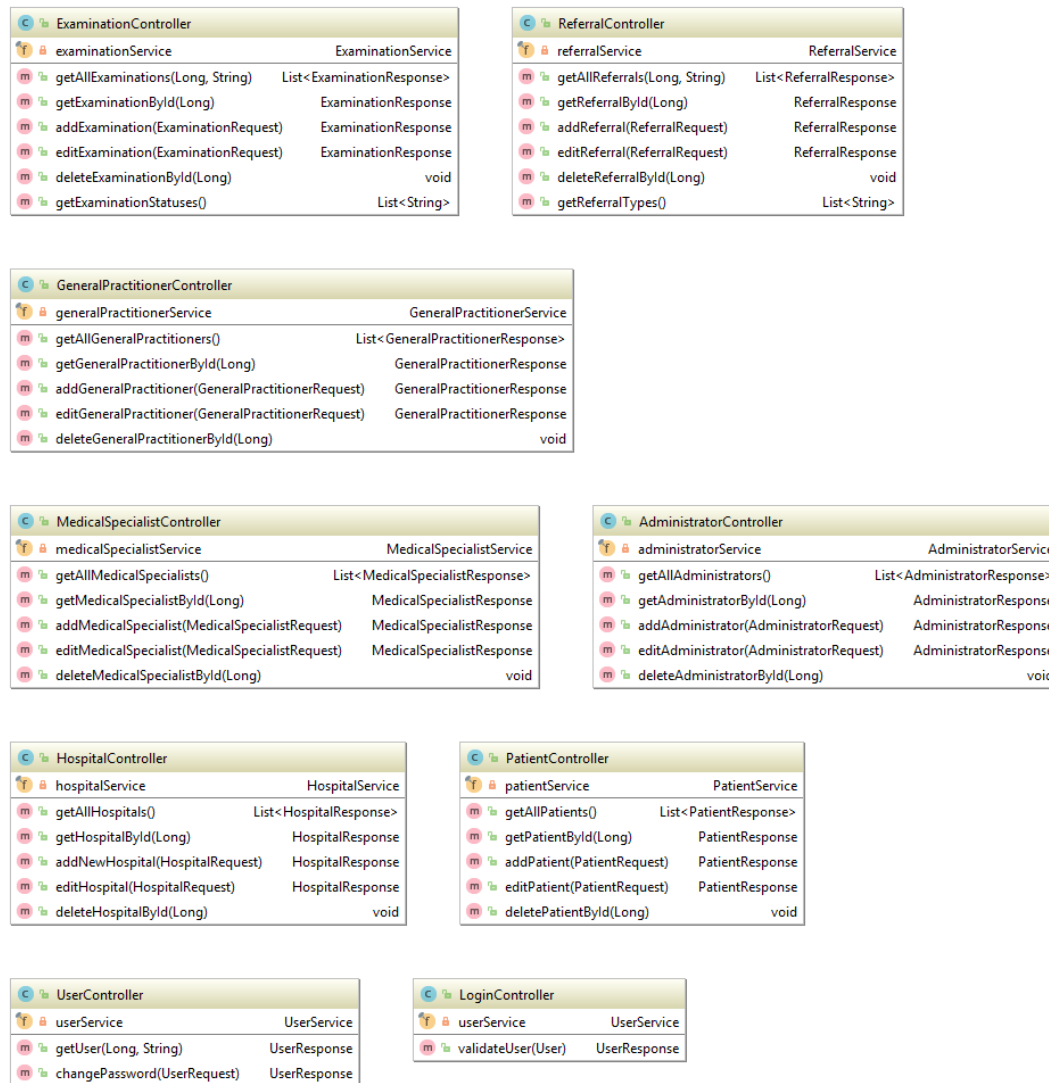
Zaprima korisničke zahtjeve za upravljanje podacima razreda **Patient**. Sadrži primjerak sučelja **PatientService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih pacijenata, metodu za dohvat pacijenta za zadani id, metodu za dodavanje novog pacijenta, metodu za uređivanje pacijenta i metodu za brisanje pacijenta za zadani id.

## **GeneralPractitionerController**

Zaprima korisničke zahtjeve za upravljanje podacima razreda **GeneralPractitioner**. Sadrži primjerak sučelja **GeneralPractitionerService** čije metode poziva i dobivene odgovore prosljeđuje korisniku.

Sadrži metodu za dohvat svih specijalista obiteljske medicine, metodu za dohvat specijalista obiteljske medicine za zadani id, metodu za dodavanje novog specijalista obiteljske medicine, metodu za uređivanje specijalista obiteljske medicine i metodu za brisanje specijalista obiteljske medicine za zadani id.



Powered by yFiles

Slika 4. Dijagram razreda paketa *controller*



### 2.3.2. Paket service

Sadrži sučelja i implementacije svih servisa.

#### **AddressDataService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **AddressData**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id.

Implementacija sučelja **AddressDataService** sadrži primjerak sučelja **AddressDataRepository** preko kojega pristupa tablici u *address\_data* bazi podataka.

#### **ExaminationService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **Examination**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvat svih statusa pregleda.

Implementacija sučelja **ExaminationService** sadrži primjerak sučelja **ExaminationRepository** preko kojega pristupa tablici *examination* u bazi podataka.

#### **HospitalService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **Hospital**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id.

Implementacija sučelja **HospitalService** sadrži primjerak sučelja **HospitalRepository** preko kojega pristupa tablici *hospital* u bazi podataka.

### **ReferralService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **Referral**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvat svih tipova uputnica.

Implementacija sučelja **ReferralService** sadrži primjerak sučelja **ReferralRepository** preko kojega pristupa tablici *referral* u bazi podataka.

### **AdministratorService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **Administrator**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvaćanje administratora za zadano korisničko ime i lozinku.

Implementacija sučelja **AdministratorService** sadrži primjerak sučelja **AdministratorRepository** preko kojega pristupa tablici *administrator* u bazi podataka.

### **GeneralPractitionerService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **GeneralPractitioner**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvaćanje specijalista obiteljske medicine za zadano korisničko ime i lozinku. Implementacija sučelja **GeneralPractitionerService** sadrži primjerak sučelja **GeneralPractitionerRepository** preko kojega pristupa tablici *general\_practitioner* u bazi podataka.

### **MedicalSpecialistService**

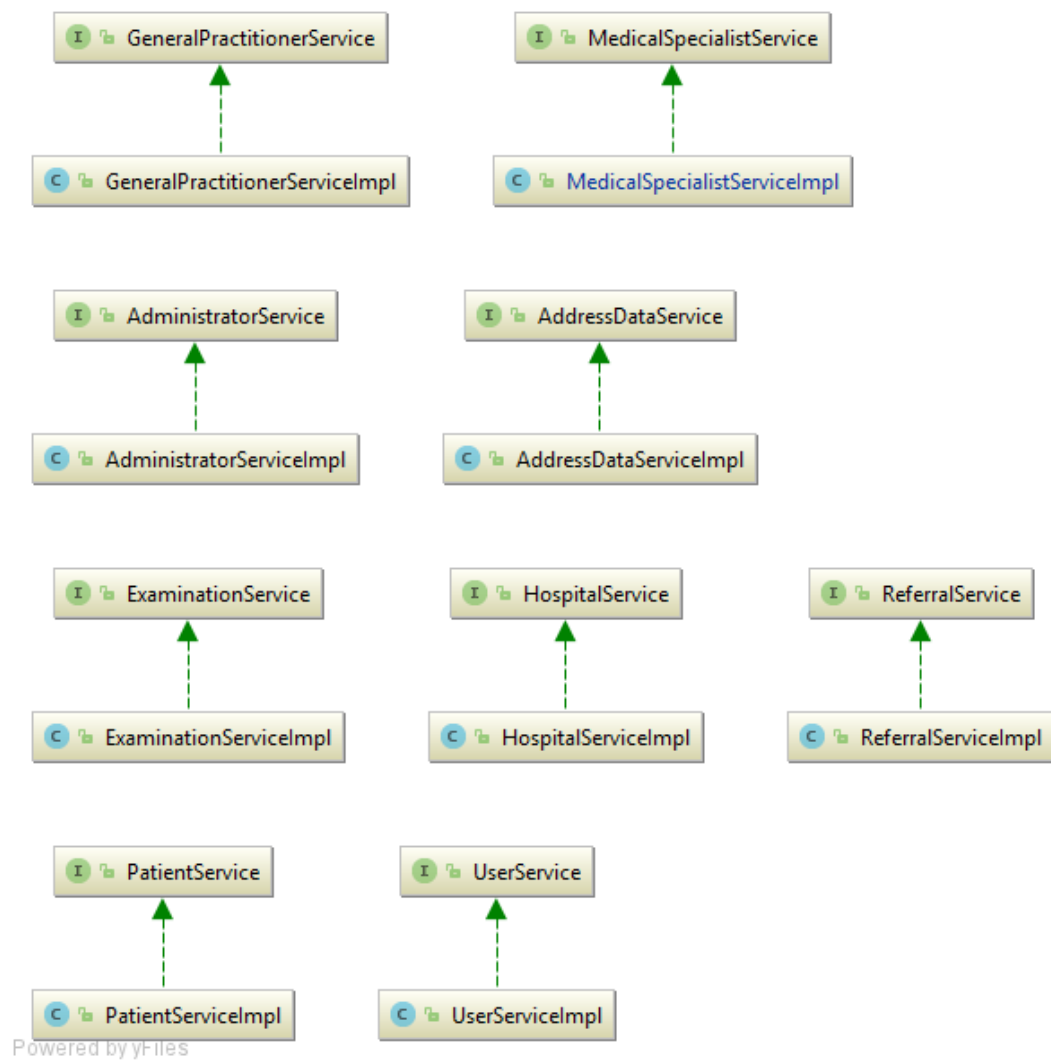
Sučelje sadrži metodu za dohvat svih zapisa entiteta **MedicalSpecialist**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvaćanje liječnika specijalista za zadano korisničko ime i lozinku. Implementacija sučelja **MedicalSpecialistService** sadrži primjerak sučelja **MedicalSpecialistRepository** preko kojega pristupa tablici *medical\_specialist* u bazi podataka.

### **PatientService**

Sučelje sadrži metodu za dohvat svih zapisa entiteta **Patient**, metodu za dohvat zapisa za zadani id, metodu za dodavanje novog zapisa, metodu za uređivanje zapisa i metodu za brisanje zapisa za zadani id. Sadrži i metodu za dohvaćanje pacijenta za zadano korisničko ime i lozinku. Implementacija sučelja **PatientService** sadrži primjerak sučelja **PatientRepository** preko kojega pristupa tablici *patient* u bazi podataka.

### **UserService**

Sučelje sadrži metodu koja za zadano korisničko ime i lozinku provjerava postoji li korisnik u bazi, metodu koja za zadani id i tip korisnika vraća odgovarajućeg korisnika i metodu za promjenu lozinke korisnika. Implementacija sučelja **UserService** poziva metode svih drugih servisa koji obavljaju manipulaciju podacima korisnika.

Slika 5. Dijagram razreda paketa *service*

### 2.3.3. Paket repository

**AddressDataRepository, ExaminationRepository, HospitalRepository, ReferralRepository, AdministratorRepository, GeneralPractitionerRepository, MedicalSpecialistRepository i PatientRepository** jesu sučelja prema pripadnim tablicama u bazi podataka koje pohranjuje zapise za entitete **AddressData, Examination, Hospital, Referral, Administrator, GeneralPractitioner, MedicalSpecialist i Patient**.



Slika 6. Dijagram razreda paketa *repository*

### 2.3.4. Paket domain

#### AddressData

Entitet koji sadrži sve podatke o adresi. Sadrži attribute: *id*, *city*, *postalCode*, *street* i *streetNumber*. Atribut **id** je tipa *Long* i primarni je ključ entiteta. Automatski se i inkrementalno generira prilikom svakog dodavanja novog zapisa u bazu podataka. Atribut **city** je tipa *String* i predstavlja naziv grada. Atribut **postalCode** je tipa *Long* i predstavlja poštanski broj. Atribut **street** je tipa *String* i predstavlja naziv ulice. Atribut **streetNumber** je tipa *Long* i predstavlja kućni broj.

#### Examination

Entitet koji sadrži sve podatke o pregledu. Sadrži attribute: *id*, *status*, *patient*, *medicalSpecialist*, *hospital*, *referral*, *term* i *remark*. Atribut **id** je tipa *Long* i primarni je ključ entiteta. Automatski se i inkrementalno generira prilikom svakog dodavanja novog zapisa u bazu podataka. Atribut **status** ima vrijednost iz enumeracije *ExaminationStatus* i predstavlja trenutni status pregleda. Atribut **patient** je tipa *Patient* i predstavlja pacijenta na koji se pregled odnosi. Atribut **medicalSpecialist** je tipa *MedicalSpecialist* i predstavlja liječnika specijalista koji obavlja pregled. Atribut **hospital** je tipa *Hospital* i predstavlja bolnicu u kojoj se pregled obavlja. Atribut **referral** je tipa *Referral* i predstavlja uputnicu kojom je pacijent naručen na pregled. Atribut **term** je tipa *LocalDate* i predstavlja termin pregleda. Atribut **remark** je tipa *String* i predstavlja bilješke za pripadajući pregled.

#### Hospital

Entitet koji sadrži sve podatke o bolnici. Sadrži attribute: *id*, *name*, *addressData*, *phoneNumber* i *mail*. Atribut **id** je tipa *Long* i primarni je ključ entiteta. Automatski se i inkrementalno generira prilikom svakog dodavanja novog zapisa u bazu podataka. Atribut **name** je tipa *String* i predstavlja naziv bolnice. Atribut **addressData** je tipa *AddressData* i sadrži podatke o adresi bolnice. Atribut

**phoneNumber** je tipa *String* i predstavlja telefonski broj bolnice. Atribut **mail** je tipa *String* i predstavlja e-mail adresu bolnice.

## Referral

Entitet koji sadrži sve podatke o uputnici. Sadrži attribute: *id*, *referralType*, *departmentType*, *patient*, *generalPractitioner*, *createdOn*, *diagnosis* i *remark*. Atribut **id** je tipa *Long* i primarni je ključ entiteta. Automatski se i inkrementalno generira prilikom svakog dodavanja novog zapisa u bazu podataka. Atribut **referralType** ima vrijednost iz enumeracije *ReferralType* i predstavlja vrstu uputnice. Atribut **departmentType** ima vrijednost iz enumeracije *DepartmentType* i predstavlja bolnički odjel za koji je izdana uputnica. Atribut **patient** je tipa *Patient* i predstavlja pacijenta za kojega je izdana uputnica. Atribut **generalPractitioner** je tipa *GeneralPractitioner* i predstavlja specijalista obiteljske medicine koji je izdao uputnicu. Atribut **createdOn** je tipa *LocalDate* i predstavlja datum kreiranja uputnice. Atribut **diagnosis** je tipa *String* i predstavlja dijagnozu koju upisuje specijalist obiteljske medicine. Atribut **remark** je tipa *String* i predstavlja dodatne napomene na uputnici.

## User

Razred koji sadrži podatke o korisniku. Nasljeđuju ga razredi koji predstavljaju specifične korisnike. Sadrži attribute: *id*, *firstName*, *lastName*, *mail*, *type*, *password* i *phoneNumber*. Atribut **id** je tipa *Long* i primarni je ključ entiteta. Automatski se i inkrementalno generira prilikom svakog dodavanja novog zapisa u bazu podataka. Atribut **firstName** je tipa *String* i predstavlja ime korisnika. Atribut **lastName** je tipa *String* i predstavlja prezime korisnika. Atribut **mail** je tipa *String* i predstavlja e-mail adresu korisnika. Koristi se kao korisničko ime prilikom prijave u aplikaciju. Atribut **type** poprima vrijednost iz enumeracije *UserType* i predstavlja tip korisnika. Atribut **password** je tipa *String* i predstavlja lozinku. Atribut **phoneNumber** je tipa *String* i predstavlja telefonski broj.

### Administrator

Entitet koji sadrži sve podatke o administratoru. Nasljeđuje razred *User*. Sadrži samo atribute koje nasljeđuje od razreda *User*.

### GeneralPractitioner

Entitet koji sadrži sve podatke o specijalistu obiteljske medicine. Nasljeđuje razred *User*. Uz atribute koje nasljeđuje od razreda *User* sadrži atribut *addressData*. Atribut **addressData** je tipa *AddressData* i sadrži sve podatke o adresi specijalista obiteljske medicine.

### MedicalSpecialist

Entitet koji sadrži sve podatke o liječniku specijalistu. Nasljeđuje razred *User*. Uz atribute koje nasljeđuje od razreda *User* sadrži atribute *departmentType* i *hospital*. Atribut **departmentType** poprima vrijednost iz enumeracije *DepartmentType* i predstavlja odjel na kojemu liječnik specijalist radi. Atribut **hospital** je tipa *Hospital* i predstavlja bolnicu u kojoj liječnik specijalist radi.

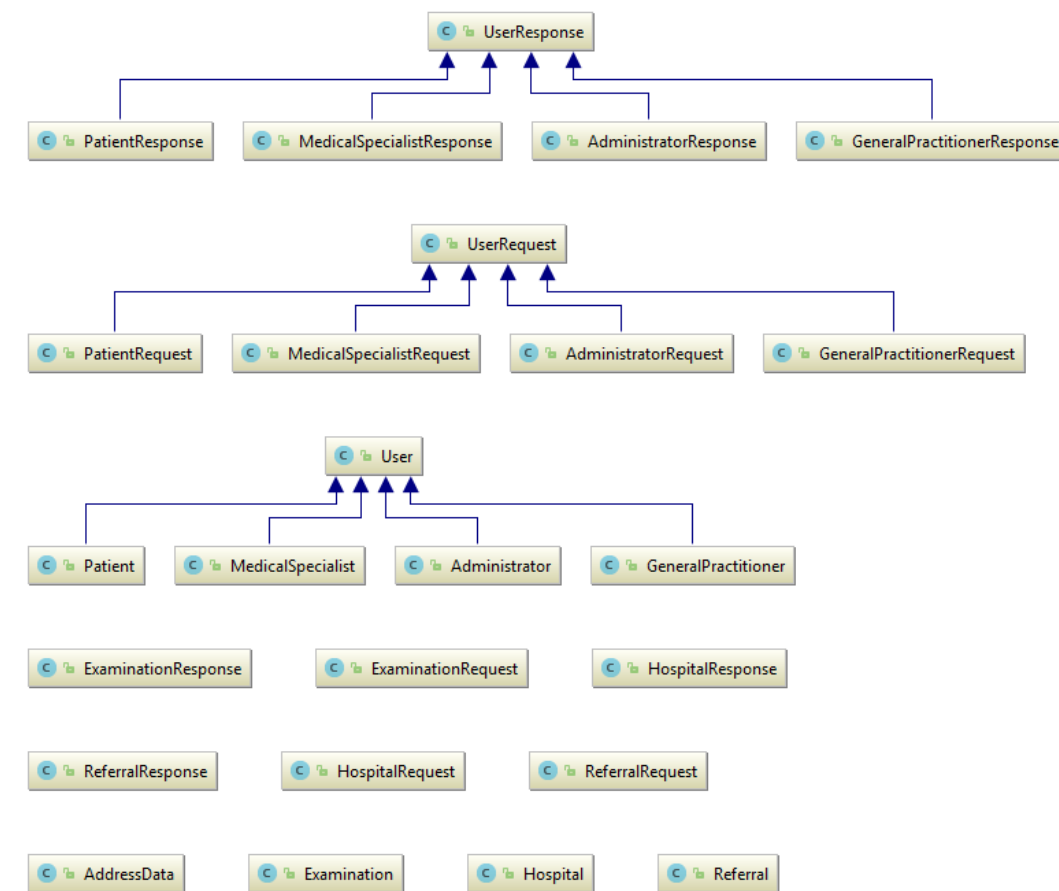
### Patient

Entitet koji sadrži sve podatke o pacijentu. Nasljeđuje razred *User*. Uz atribute koje nasljeđuje od razreda *User* sadrži atribute; *addressData*, *OIB*, *dateOfBirth* i *sex*. Atribut **addressData** je tipa *AddressData* i sadrži sve podatke o adresi pacijenta. Atribut **OIB** je tipa *String* i predstavlja oib pacijenta. Atribut **dateOfBirth** je tipa *LocalDate* i predstavlja datum rođenja pacijenta. Atribut **sex** je tipa *String* i predstavlja spol pacijenta.



## Ostali razredi

Uz navedene razrede u paketu *domain* nalaze se i razredi koji se koriste pri slanju odgovora korisniku ili za primanje podataka u web zahtjevima. To su razredi sa sufiksima *Response* i *Request*.



Slika 7. Dijagram razreda paketa *domain*

### 2.3.5. Paket enumeration

#### DepartmentType

Sadrži pobrojane sve bolničke odjele. Svaki odjel ima svoj jedinstveni identifikator koji je tipa *Long* i naziv koji je tipa *String*.

#### ExaminationStatus

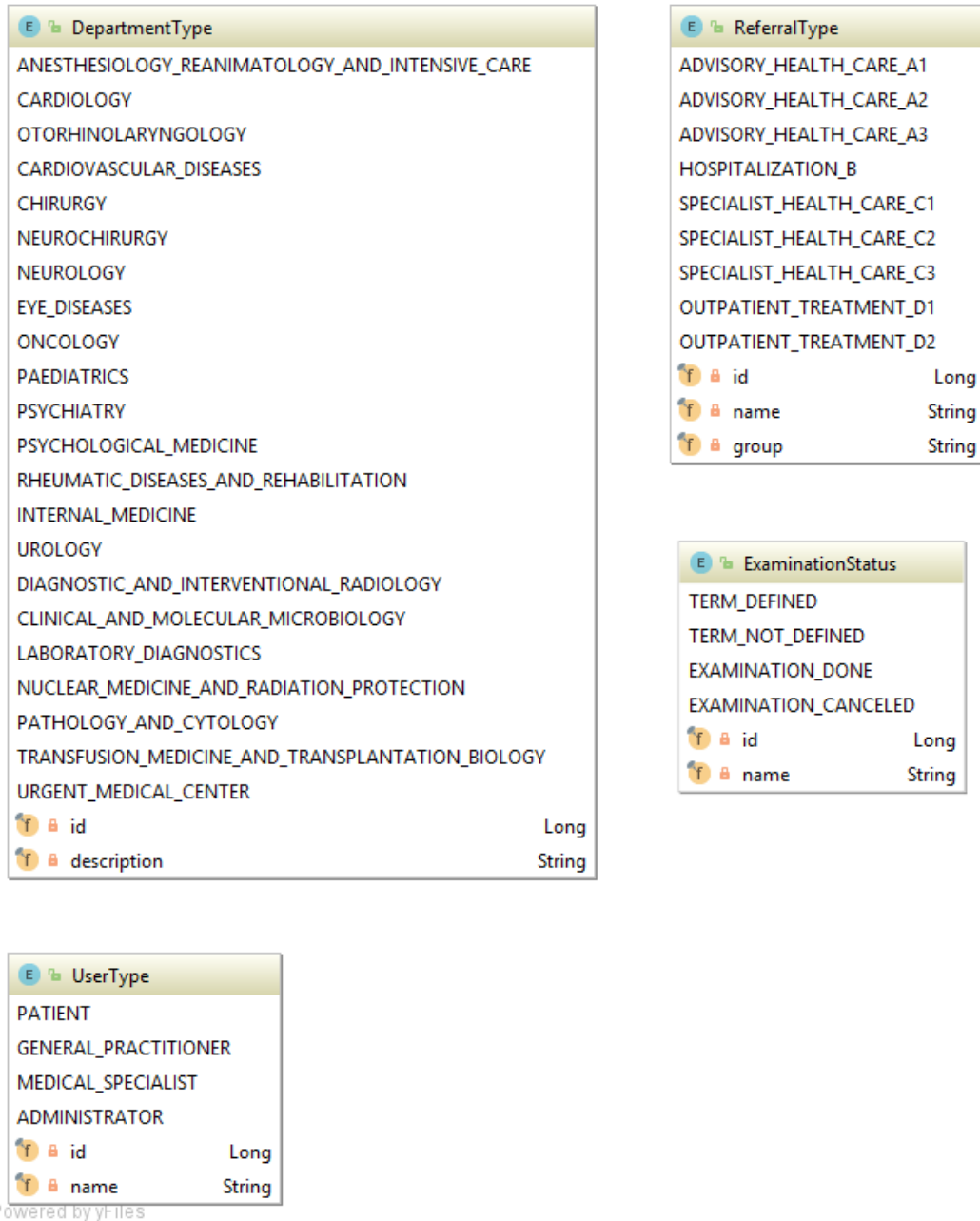
Sadrži pobrojane sve moguće statuse pregleda. Svaki pregled ima svoj jedinstveni identifikator koji je tipa *Long* i naziv koji je tipa *String*. Postoje četiri različita statusa koje pregled može imati. Prvi je da termin pregleda još nije definiran, drugi status je kada je termin pregleda definiran, ali pregled još nije obavljen, treći status je da je pregled obavljen i posljednji je da je pregled otkazan.

#### ReferralType

Sadrži pobrojane sve vrste uputnica. Svaka uputnica ima svoj jedinstveni identifikator koji je tipa *Long*, vrstu koja je tipa *String* i podvrstu koja je tipa *String*.

#### UserType

Sadrži pobrojane sve vrste korisnika. Svaki korisnik ima svoj jedinstveni identifikator koji je tipa *Long* i naziv koji je tipa *String*. Postoje četiri vrste korisnika, a to su: administrator, specijalist obiteljske medicine, liječnik specijalist i pacijent.

Slika 8. Dijagram razreda paketa *enumeration*

## 3. Implementacija i korisničko sučelje

### 3.1. Korištene tehnologije i alati

Za izradu programskog sustava odabrani su niže navedeni alati. Izabrani su navedeni alati jer omogućavaju brz i jednostavan razvoj web aplikacije.

#### Intelij IDEA

Razvojno okruženje za razvoj programske podrške integrirano s Javom. Omogućuje brz i učinkovit razvoj projekata.

#### Git

Besplatan sustav kontrole verzija dizajniran za upravljanje projektima velikom brzinom i učinkovitosti.

#### Java

Objektno orijentirani programski jezik. Java je statički tipizirani programski jezik. Jedan je od najkorištenijih programskih jezika. Velika prednost Jave je ta što se programi napisani u Javi mogu izvoditi bez preinaka na svim operacijskim sustavima za koje postoji Java virtualni stroj. Java virtualni stroj je aplikacija u računalu koja interpretira i prevodi Java *bytecode*.

#### React

React je biblioteka JavaScript programskog jezika koja služi za izradu korisničkog sučelja. Omogućuje razvoj velikih web-aplikacija koje koriste podatke koji se mogu mijenjati kroz vrijeme bez učitavanja stranice. Prvenstveno nastoji osigurati brzinu, jednostavnost i skalabilnost.

## Redux

Redux je spremnik koji sadrži stanje JavaScript aplikacije. Primarno (ali ne i nužno) se koristi s React-om za izgradnju korisničkih sučelja.

## Spring Boot

Spring Boot olakšava razvoj web aplikacija pisanih u programskom jeziku Java. Smanjuje vrijeme koje je potrebno uložiti u razvoj i povećava produktivnost. Nije potrebno pisati puno koda, anotacija i konfiguracijskih datoteka. Spring Boot ima postavljene podrazumijevane konfiguracije koje je moguće mijenjati po potrebi. Pruža ugrađene HTTP servere koji omogućuju lagan razvoj i testiranje aplikacije. Pruža sučelje kroz konzolu za razvoj i testiranje aplikacije. Pruža mnogo dodataka koji olakšavaju rad s bazama podataka.

## H2

Ugrađena baza podataka koju koristi Spring Boot i podatke pohranjuje u radnoj memoriji računala.

## CSS

CSS je stilski jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću HTML-a. CSS-om se uređuje sam izgled stranice.

## HTML

Prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML-a. HTML-om oblikuje se sadržaj i stvaraju se hiperveze *hipertext* dokumenta. Prikaz *hipertext* dokumenta omogućuje web preglednik. Temeljna zadaća HTML-a jest uputiti web preglednik kako prikazati *hipertext* dokument.

## **Hibernate ORM**

Hibernate ORM je radni okvir koji mapira objektno-relacijski model baze podataka u programskom jeziku Java. Pruža okvir za mapiranje objektno orijentiranog modela u relacijsku bazu podataka. Njegova glavna značajka je mapiranje iz Java klase u tablice baze podataka. Također nudi sučelje za rad sa SQL upitima i obrađuje konverziju rezultata SQL upita u Java objekte.

## **JSON**

Java Script Object Notation je minimalan format paketa za prijenos podataka koji ima prednost u lakoći čitanja i pisanja za ljude, te se odlikuje lakoćom strojne obrade generiranjem i parsiranjem što ga čini pogodnim kao sredstvo za prijenos podataka neovisno o tome koja se platforma koristi.

## 3.2. Isječci programskog koda vezani za temeljnu funkcionalnost

### MedicalSpecialistController

Razred zadužen za zaprimanje zahtjeva za obradu podataka za liječnike specijaliste. Sadrži primjerak sučelja *MedicalSpecialistService* čije metode poziva i čije odgovore vraća pošiljatelju zahtjeva.

```
17 @RestController
18 public class MedicalSpecialistController {
19
20     private final MedicalSpecialistService medicalSpecialistService;
21
22     @Autowired
23     public MedicalSpecialistController(final MedicalSpecialistService medicalSpecialistService) {
24         this.medicalSpecialistService = medicalSpecialistService;
25     }
26
27     @GetMapping("/api/medical-specialists")
28     public List<MedicalSpecialistResponse> getAllMedicalSpecialists() { return this.medicalSpecialistService.getAll(); }
29
30     @GetMapping("/api/medical-specialists/{id}")
31     public MedicalSpecialistResponse getMedicalSpecialistById(@PathVariable final Long id) {
32         return this.medicalSpecialistService.getById(id);
33     }
34
35     @PostMapping("/api/medical-specialists")
36     public MedicalSpecialistResponse addMedicalSpecialist(@RequestBody final MedicalSpecialistRequest medicalSpecialistRequest) {
37         return this.medicalSpecialistService.add(medicalSpecialistRequest);
38     }
39
40     @PutMapping("/api/medical-specialists")
41     public MedicalSpecialistResponse editMedicalSpecialist(@RequestBody final MedicalSpecialistRequest medicalSpecialistRequest) {
42         return this.medicalSpecialistService.edit(medicalSpecialistRequest);
43     }
44
45     @DeleteMapping("/api/medical-specialists/{id}")
46     public void deleteMedicalSpecialistById(@PathVariable final Long id) { this.medicalSpecialistService.deleteById(id); }
47
48 }
```

Slika 9. Razred *MedicalSpecialistController*

## MedicalSpecialistService

Sučelje koje sadrži metode za upravljanje podataka za liječnike specijaliste. Razredi koji implementira ovo sučelje sadrži primjerak sučelja *MedicalSpecialistRepository* preko kojega pristupa tablici liječnika specijalista u bazi podataka.

```
1 package hr.fer.snarp.service.users.medicalSpecialist;
2
3 import hr.fer.snarp.domain.users.medicalSpecialist.MedicalSpecialist;
4 import hr.fer.snarp.domain.users.medicalSpecialist.MedicalSpecialistRequest;
5 import hr.fer.snarp.domain.users.medicalSpecialist.MedicalSpecialistResponse;
6
7 import java.util.List;
8
9 public interface MedicalSpecialistService {
10
11     List<MedicalSpecialistResponse> getAll();
12
13     MedicalSpecialistResponse getById(Long id);
14
15     MedicalSpecialistResponse add(MedicalSpecialistRequest medicalSpecialistRequest);
16
17     MedicalSpecialistResponse edit(MedicalSpecialistRequest medicalSpecialistRequest);
18
19     void deleteById(Long id);
20
21     MedicalSpecialist getByMailAndPassword(String mail, String password);
22
23     MedicalSpecialistResponse getMedicalSpecialistResponse(final MedicalSpecialist medicalSpecialist);
24 }
```

Slika 10. Razred *MedicalSpecialistService*



## MedicalSpecialistRepository

Sučelje preko kojega se pristupa tablici *medical\_specialist* koja sadrži zapise za liječnike specijaliste u bazi podataka. Nasljeđuje parametrizirano sučelje *CrudRepository* i parametrizira ga tipovima podataka *MedicalSpecialist* koji predstavlja entitet koji se pohranjuje u tablici i *Long* koji predstavlja primarni ključ entiteta. Sučelje *CrudRepository* pruža mnoge korisne metode. Neke od metoda su: metoda za dohvaćanje svih zapisa u tablici podataka, metoda za dohvaćanje zapisa iz tablice za zadani primarni ključ, metoda za dodavanje novog zapisa i metoda za brisanje zapisa iz tablice za zadani primarni ključ. Ako je potrebno moguće je definirati dodatne metode koje dohvaćaju zapise za zadane neke druge parametre. Sučelje *MedicalSpecialistRepository* pruža dvije dodatne metode. To su metoda za dohvaćanje zapisa za zadanu mail adresu i lozinku i metoda za dohvaćanje zapisa za zadanu bolnicu i odjel.

```

1 package hr.fer.snarp.repository.users;
2
3 import hr.fer.snarp.domain.hospital.Hospital;
4 import hr.fer.snarp.domain.users.medicalSpecialist.MedicalSpecialist;
5 import hr.fer.snarp.enumeration.DepartmentType;
6 import org.springframework.data.repository.CrudRepository;
7
8 public interface MedicalSpecialistRepository extends CrudRepository<MedicalSpecialist, Long> {
9
10     MedicalSpecialist findByMailAndPassword(String mail, String password);
11
12     MedicalSpecialist findByHospitalAndDepartmentType(Hospital hospital, DepartmentType departmentType);
13 }

```

Slika 11. Razred *MedicalSpecialistRepository*

## MedicalSpecialist

Razred predstavlja entitet koji sadrži podatke o jednom liječniku specijalistu. Nasljeđuje razred *User*.

```

1  package hr.fer.snarp.domain.users.medicalSpecialist;
2
3  import hr.fer.snarp.domain.hospital.Hospital;
4  import hr.fer.snarp.domain.users.user.User;
5  import hr.fer.snarp.enumeration.DepartmentType;
6  import hr.fer.snarp.enumeration.UserType;
7  import lombok.Data;
8
9  import javax.persistence.*;
10
11  @Data
12  @Entity
13  public class MedicalSpecialist extends User {
14
15      @Enumerated(EnumType.STRING)
16      private DepartmentType departmentType;
17
18      @ManyToOne
19      @JoinColumn(name = "hospital_id")
20      private Hospital hospital;
21
22      public MedicalSpecialist() {
23      }
24
25      public MedicalSpecialist(final MedicalSpecialistRequest medicalSpecialistRequest) {
26          super(medicalSpecialistRequest, UserType.MEDICAL_SPECIALIST);
27
28          this.departmentType = DepartmentType.getByDescription(medicalSpecialistRequest.getDepartmentType());
29      }

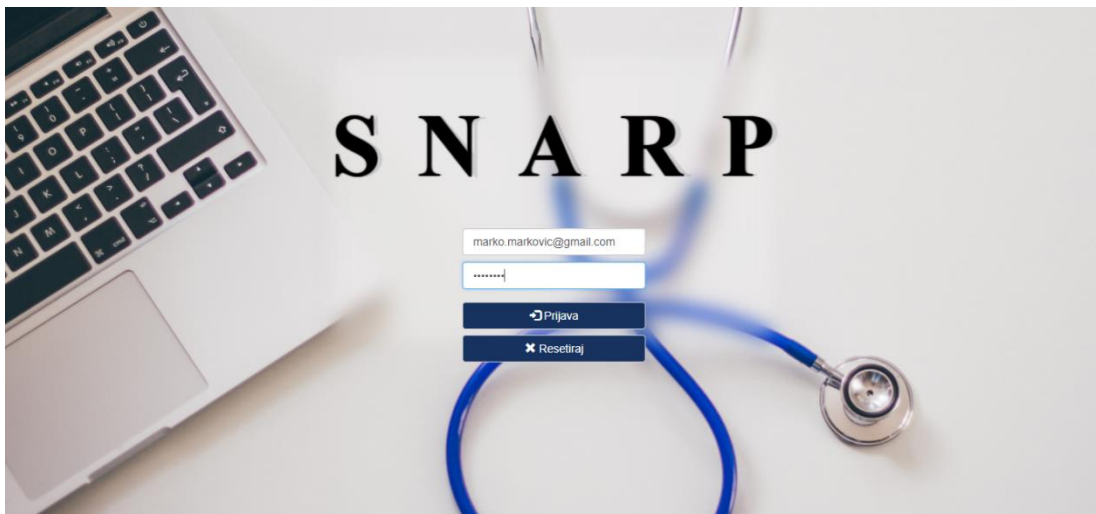
```

Slika 12. Razred *MedicalSpecialist*

### 3.3. Korisničko sučelje

#### 3.3.1. Početna stranica

Početna stranica aplikacije sadrži formu za prijavu korisnika kroz koju postojeći korisnici sustava unose podatke za prijavu i prijavljuju se u aplikaciju. Naslov **SNARP** je kratica od *Sustav za naručivanje pacijenata*. Podaci za prijavu su korisničko ime koje je e-mail adresa korisnika i lozinka. Kada se unesu potrebni podaci, prijava se obavlja pritiskom na gumb *Prijava*. Ako su uneseni podaci ispravni korisnik se prijavljuje uspješno i usmjerava ga se na neki od ekrana aplikacije, ovisno o ulozi koju ima. Ako uneseno korisničko i ime nisu ispravni korisniku se dojavljuje poruka o neispravnim podacima te se od njega traži da ponovi unos podataka. Resetiranje unesenih podataka obavlja se pritiskom na gumb *Resetiraj*.



Slika 13. Početna stranica

### 3.3.2. Korisničko sučelje administratora

#### Podaci koje vidi administrator

Administrator ima pristup tablicama svih korisnika, dakle ima pristup podacima o ostalim administratorima, podacima o specijalistima obiteljske medicine, podacima o liječnicima specijalistima i podacima o pacijentima. Administrator ima i pristup podacima o svim bolnicama. Podaci su prikazani u tablicama. Moguće je sortirati podatke po bilo kojem stupcu tablice silazno i uzlazno. Moguće je i mijenjati broj redaka tablice koji se odjednom prikazuju na ekranu. Može dodavati, uređivati i brisati korisnike iz sustava i može dodavati, uređivati i brisati bolnice iz sustava.

The screenshot shows the SNARP administrator interface. A dark blue header bar contains the logo 'SNARP' and navigation links: 'Korisnici', 'Bolnice', 'Postavke', and a user profile 'Toni Martinčić'. A dropdown menu is open under 'Korisnici', showing options: 'Administratori', 'Specijalisti obiteljske medicine', 'Liječnici specijalisti', and 'Pacijenti'. The 'Liječnici specijalisti' option is selected, and a table of specialists is displayed below. The table has columns for ID, Ime, Prezime, Mail, and Broj mobitela. Below the table are pagination controls (5, 0, 1, Naprijed) and three action buttons: '+ Dodaj', 'Uredi', and 'Obriši'.

ID	Ime	Prezime	Mail	Broj mobitela
1	Valentina	Kunić	valentina.kunic@gmail.com	095 160 0143
2	Petra	Petrić	petra.petric@gmail.com	095 162 3542
3	Mia	Franjčić	mia.franjic@gmail.com	095 330 0193
4	Iris	Martinčić	iris.martincic@gmail.com	095 999 5676
5	Dino	Šemsudin	dino.semsudin@gmail.com	091 202 1333

Slika 14. Korisničko sučelje administratora

## Dodavanje zapisa

Dodavanje zapisa obavlja se odabirom gumba *Dodaj* na dnu stranice. Nakon što se pritisne gumb za dodavanje otvara se forma za unos podataka. Spremanje podataka se pokreće pritiskom na gumb *Spremi*. Ako se želi odustati od spremanja potrebno je pritisnuti gumb *Odustani* i u tom slučaju se forma za unos zatvara i podaci uneseni u formu se gube. Prilikom dodavanja zapisa obavlja se validacija unesenih vrijednosti. Ako vrijednosti nisu unesene ispravno onda se podaci ne spremaju u sustav već se korisniku dojavljuje odgovarajuća poruka.

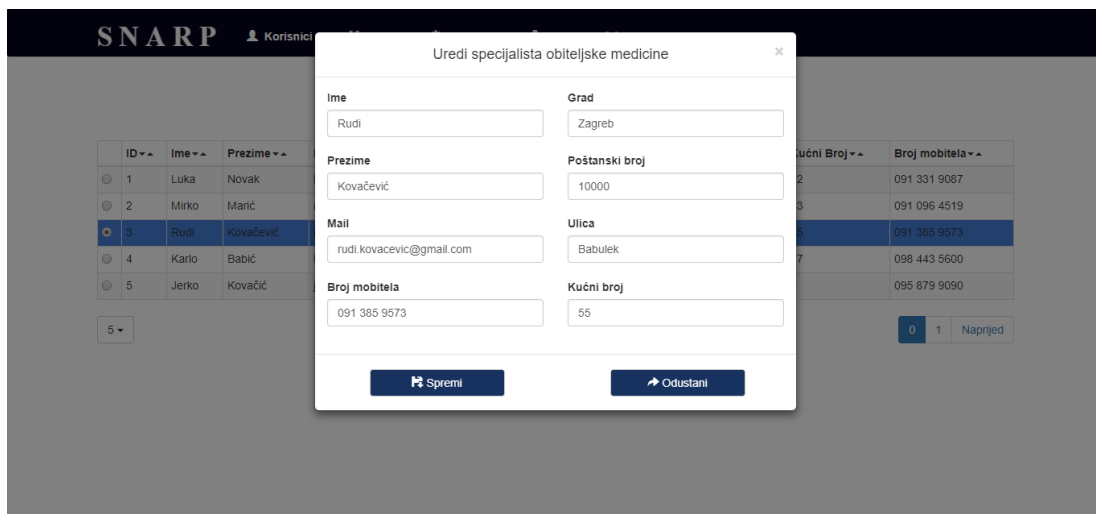
**Dodaj specijalista obiteljske medicine**

<b>Ime</b> <input type="text" value="Unesite ime"/> Morate unijeti ime.	<b>Grad</b> <input type="text" value="Unesite grad"/> Morate unijeti grad.
<b>Prezime</b> <input type="text" value="Unesite prezime"/> Morate unijeti prezime.	<b>Poštanski broj</b> <input type="text" value="Unesite poštanski broj"/> Morate unijeti poštanski broj.
<b>Mail</b> <input type="text" value="Unesite mail"/> Morate unijeti mail adresu.	<b>Ulica</b> <input type="text" value="Unesite ulicu"/> Morate unijeti ulicu.
<b>Broj mobitela</b> <input type="text" value="Unesite broj mobitela"/> Morate unijeti broj mobitela.	<b>Kućni broj</b> <input type="text" value="Unesite kućni broj"/> Morate unijeti kućni broj.

Slika 15. Dodavanje zapisa o specijalistu obiteljske medicine

## Uređivanje zapisa

Uređivanje se obavlja tako da se odabere redak u tablici koji se želi urediti i pritisne se gumb *Uredi* koji se nalazi na dnu stranice. Ako nije odabran nijedan zapis, a pritisne se gumb *Uredi* onda se korisniku dojavljuje poruka da mora odabrati zapis koji želi urediti. Ako je odabran zapis u tablici i pritisne se gumb za uređivanje otvara se forma za uređivanje podataka. Unesene promjene se pohranjuju tek nakon što se odabere gumb *Spremi*. Ako se želi odustati od uređivanja potrebno je pritisnuti gumb *Odustani* i u tom slučaju se forma za uređivanje zatvara i zapis ostaje nepromijenjen. Prilikom uređivanja zapisa obavlja se jednaka validacija unesenih vrijednosti kakva se obavlja i kod dodavanja novih zapisa. Ako vrijednosti nisu unesene ispravno onda se podaci ne spremaju u sustav već se korisniku dojavljuje odgovarajuća poruka.



The screenshot displays the SNARP application interface. In the background, there is a table with columns: ID, Ime (Name), and Prezime (Surname). The table contains five records, with the third record (ID 3, Name Rudi, Surname Kovačević) selected. A modal window titled "Uredi specijalista obiteljske medicine" (Edit family medicine specialist) is open in the foreground. The modal contains the following fields and values:

Ime	Grad
Rudi	Zagreb

Prezime	Poštanski broj
Kovačević	10000

Mail	Ulica
rudi.kovacevic@gmail.com	Babulek

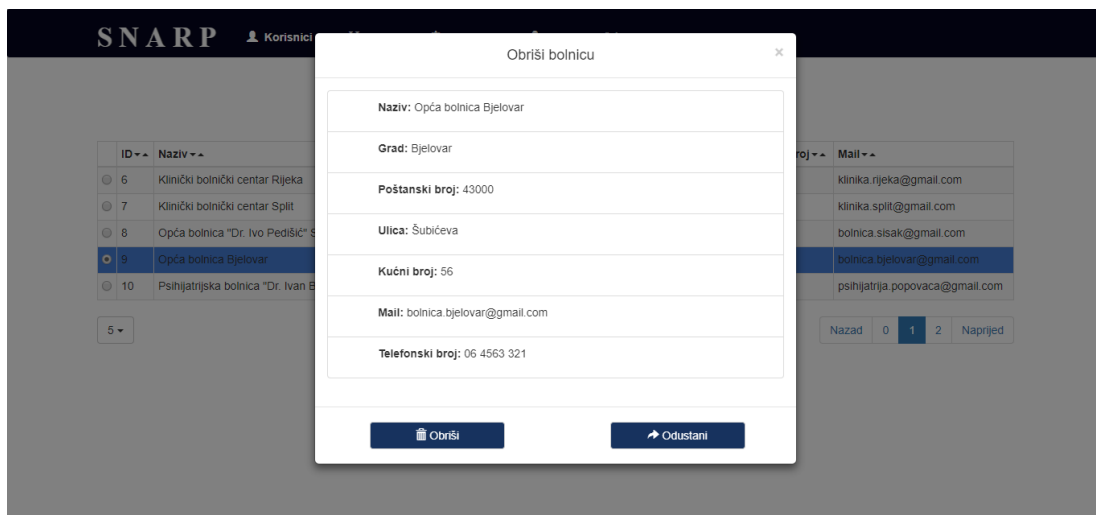
Broj mobitela	Kućni broj
091 385 9573	55

At the bottom of the modal, there are two buttons: "Spremi" (Save) and "Odustani" (Cancel).

Slika 16. Uređivanje podataka o specijalistu obiteljske medicine

## Brisanje zapisa

Brisanje se obavlja tako da se odabere redak u tablici koji se želi obrisati i pritisne se gumb *Obriši* koji se nalazi na dnu stranice. Ako nije odabran nijedan zapis, a pritisne se gumb *Obriši* onda se korisniku dojavljuje poruka da mora odabrati zapis koji želi obrisati. Ako je odabran zapis u tablici i pritisne se gumb za brisanje zapisa prikazuje se ekran s podacima o odabranom zapisu. Zapis se briše tek nakon odabira gumba *Obriši* na prikazanom ekranu. Za odustajanje od brisanja zapisa potrebno je odabrati gumb *Odustani* na prikazanom ekranu.



Slika 17. Brisanje zapisa o bolnici

### 3.3.3. Korisničko sučelje specijalista obiteljske medicine

#### Pregled pacijenata

Specijalist obiteljske medicine ima omogućen pregled svih svojih pacijenata. Pregled je jednak pregledu pacijenata koji ima administrator, jedina je razlika da administrator vidi sve pacijente u sustavu, a specijalist obiteljske medicine vidi samo svoje pacijente. Specijalist obiteljske medicine može dodavati nove pacijente, uređivati podatke o postojećim pacijentima i brisati pacijente iz sustava.

**SNARP**
Pacijenti
Uputnice
Postavke
Luka Novak

#### Pacijenti

ID	Ime	Prezime	Spol	OIB	Datum rođenja	Mail	Grad	Poštanski broj	Ulica	Kućni Broj	Broj mobitela
1	Marko	Marković	M	10525231193	09-11-1984	marko.markovic@gmail.com	Zagreb	10000	Ulica sreće	3	091 331 9087
2	Mirko	Marković	M	74180178817	19-03-1981	mirko.markovic@gmail.com	Zagreb	10000	Ulica sreće	3	091 096 4519
3	Petar	Zecimir	M	53569680271	11-06-1972	petar.zecimir@gmail.com	Zagreb	10000	Strojarska	8	091 385 9573
4	Miroslav	Puška	M	02851978197	01-08-2000	miroslav.puska@gmail.com	Zagreb	10000	Vukovarska	62	098 443 5600
5	Vladimir	Primorac	M	89362962652	25-03-1996	vladimir.primorac@gmail.com	Zagreb	10000	Ulica Nikole Tesle	15	095 879 9090

5
0
1
Naprijed

+ Dodaj
Uredi
Obrisi

Slika 18. Pregled zapisa o pacijentima



## Pregled uputnica

Specijalist obiteljske medicine ima omogućen pregled svih uputnica koje je izdao svojim pacijentima. Ima mogućnost dodavanja novih uputnica, uređivanja podataka postojećih uputnica i brisanja uputnica iz sustava.

**SNARP**
Pacijenti
Uputnice
Postavke
Luka Novak

### Uputnice

ID	Ime pacijenta	Prezime pacijenta	Vrsta uputnice	Podvrsta uputnice	Odjel	Datum kreiranja
1	Marko	Marković	A – Konzilijarna zdravstvena zaštita	A1 - Konzilijarni pregled	Anestezijologija, reanimatologija i intenzivno liječenje	04-06-2018
2	Mirko	Marković	A – Konzilijarna zdravstvena zaštita	A2 - Kontrolni konzilijarni pregled	Kardiologija	04-06-2018
3	Petar	Zecimir	A – Konzilijarna zdravstvena zaštita	A3 - Dijagnostička pretraga	Otorinolaringologija	04-06-2018

5
0

Pregled
Dodaj
Uredi
Obriši

Slika 19. Pregled uputnica

## Dodavanje uputnice

Za dodavanje uputnice potrebno je odabrati gumb *Dodaj* nakon čega se otvara forma za dodavanje. Kroz formu se unose podaci za uputnicu koja se dodaje. To su vrsta uputnice koja se odabire iz padajućeg izbornika, bolnički odjel koji se odabire iz padajućeg izbornika, pacijent koji se odabire iz padajućeg izbornika i dijagnoza i napomena koje se unose kao tekst. Svi podaci osim napomene su obavezni i ako nisu uneseni nije moguće spremiti uputnicu. Za spremanje uputnice potrebno je odabrati gumb *Spremi*. Za odustajanje od dodavanja uputnice potrebno je odabrati gumb *Odustani* i u tom slučaju se forma za dodavanje zatvara i uneseni podaci se gube.

**Dodaj uputnicu**

Odaberi vrstu uputnice  
A2 - Kontrolni konzilijarni pregled

Odaberi odjel  
Pedijatrija

Odaberi pacijenta  
Petar Zecimir (53569680271)

Dijagnoza  
Naevus naevocellularis pigmentosus

Napomena  
Molim hitno!

+ Spremi      ↗ Odustani

ID	Ime pacijenta	Prezime pacijenta
1	Marko	Marković
2	Mirko	Marković
3	Petar	Zecimir

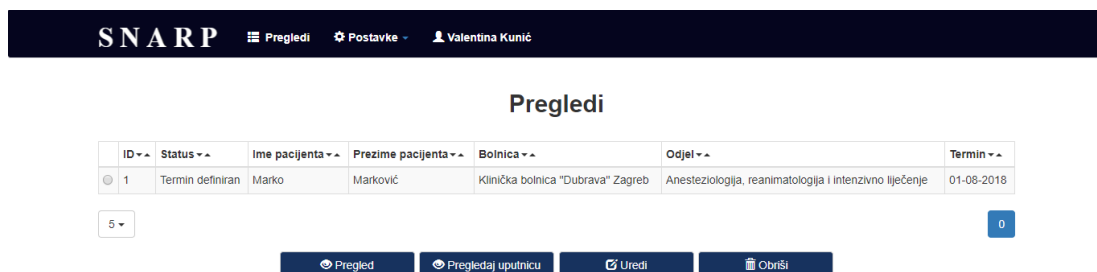
Patologija i intenzivno		Datum kreiranja
		04-06-2018
		04-06-2018
		04-06-2018

Slika 20. Dodavanje uputnice

### 3.3.4. Korisničko sučelje liječnika specijalista

#### Pregled zapisa o obavljenim pregledima

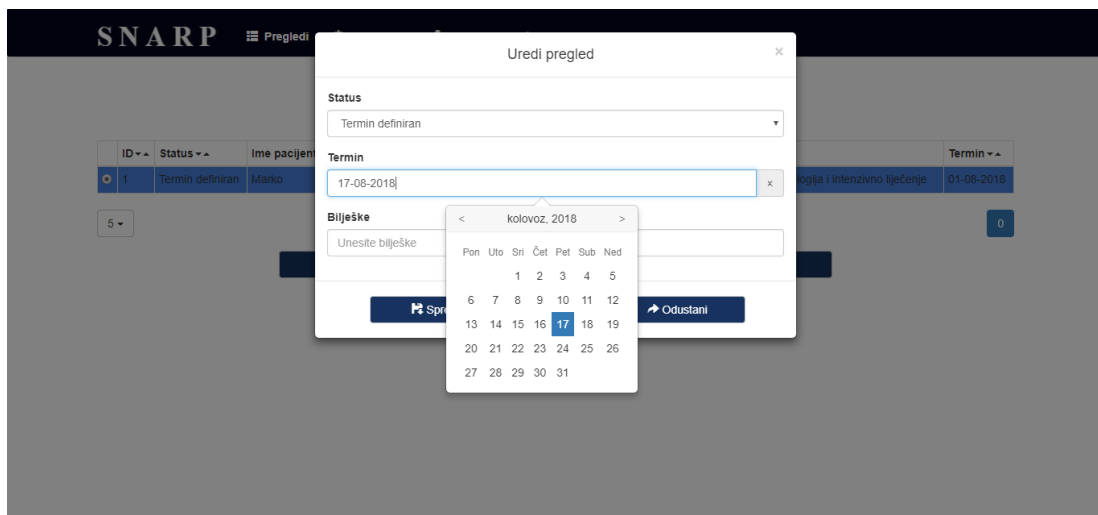
Liječnik specijalist ima omogućen prikaz svih pregleda koje je obavio. Za svaki pregled može otvoriti detaljan prikaz i može otvoriti detaljan prikaz uputnice kojom je pacijent naručen na pregled. Detaljan prikaz pregleda otvara se tako da se odabere zapis u tablici za koji se želi otvoriti detaljan prikaz i nakon toga se odabire gumb *Pregled*. Detaljan prikaz uputnice kojom je pacijent naručen na pregled se otvara tako da se odabere zapis u tablici i nakon toga se odabere gumb *Pregledaj uputnicu*. Liječniku specijalistu omogućeno je i uređivanje i brisanje pregleda. Uređivanje se radi tako da se odabere zapis i pritisne gumb *Uredi*, a brisanje tako da se odabere zapis i odabere gumb *Obriši*. Ako se odabere neki od navedenih gumba, a da prije toga nije odabran zapis u tablici korisniku se pojavljuje poruka da mora odabrati zapis.



Slika 21. Pregled zapisa o pregledima

## Određivanje termina pregleda

Liječnik specijalist je zadužen za određivanje termina pregleda. Termin pregleda određuje se tako da liječnik specijalist odabere pregled iz tablice za koji želi odrediti termin i odabere gumb *Uredi*. Nakon toga se otvara forma za uređivanje pregleda kroz koju liječnik specijalist može odrediti trenutni status pregleda i termin pregleda koji odabire kroz kalendar. Može upisati i bilješke vezane uz pregled. Promjene se pohranjuju tek nakon odabira gumba *Spremi*. Za odustajanje od promjena potrebno je odabrati gumb *Odustani*.



Slika 22. Određivanje termina pregleda

### 3.3.5. Korisničko sučelje pacijenta

#### Naručivanje na preglede

Pacijent ima mogućnost naručivanja na preglede. Naručivanje se obavlja tako da se u navigacijskoj traci odabere *Naručivanje* nakon čega se otvara forma kroz koju se naručuje na preglede. Za naručivanje potrebno je iz padajućeg izbornika odabrati bolnicu u koju se naručuje i iz padajućeg izbornika odabrati uputnicu kojom se naručuje. Oba podatka su obavezna i ako nisu odabrana nije se moguće naručiti. Nakon što su odabrani podaci potrebno je odabrati gumb *Naruči se*. Ako se naručivanje izvrši uspješno korisniku se pokazuje poruka koja to potvrđuje. Nakon što je narudžba poslana pacijent dobiva zapis u svojoj tablici pregleda koji može vidjeti ako ode na stranicu *Pregledi*. Taj isti pregled pojavljuje se i u tablici pregleda liječnika specijalista kojemu je narudžba poslana. To je liječnik specijalist koji radi u bolnici i na odjelu za koje je narudžba poslana. Kada liječnik specijalist odabere termin za taj pregled, taj termin se upisuje u tablicu koju vidi pacijent i tako pacijent zna kada treba doći na pregled.

The screenshot shows the 'Naručivanje' (Ordering) section of the SNARP application. At the top is a dark navigation bar with the logo 'SNARP' and menu items: '+ Naručivanje', 'Pregledi', 'Uputnice', 'Postavke', and a user profile 'Marko Marković'. Below the navigation bar, the title 'Naručivanje' is centered. The form contains two dropdown menus: 'Bolnica' (Hospital) with 'Opća bolnica Bjelovar' selected, and 'Uputnica' (Referral) with 'A – Konzilijarna zdravstvena zaštita / A1 – Konzilijarni pregled / Anesteziologija, re' selected. Below these is a blue button with a megaphone icon and the text 'Naruči se'. At the bottom, a green success message box states 'Naručivanje izvršeno uspješno.' (Ordering completed successfully.) with a close 'x' button.

Slika 23. Naručivanje na preglede

## Pregled naručenih pregleda

**SNARP**
+ Naručivanje
 Pregledi
 Uputnice
 Postavke
 Marko Marković

### Pregledi

ID	Status	Ime pacijenta	Prezime pacijenta	Bolnica	Odjel	Termin
1	Termin definiran	Marko	Marković	Klinička bolnica "Dubrava" Zagreb	Anesteziologija, reanimatologija i intenzivno liječenje	01-08-2018
9	Termin nije definiran	Marko	Marković	Opća bolnica Bjelovar	Anesteziologija, reanimatologija i intenzivno liječenje	

5
0

Pregled
Pregledaj uputnicu

Slika 24. Pregled naručenih pregleda kakve vide pacijenti

## Pregled uputnica

**SNARP**
+ Naručivanje
 Pregledi
 Uputnice
 Postavke
 Marko Marković

### Uputnice

ID	Ime pacijenta	Prezime pacijenta	Vrsta uputnice	Podvrsta uputnice	Odjel	Datum kreiranja
1	Marko	Marković	A – Konzilijarna zdravstvena zaštita	A1 - Konzilijarni pregled	Anesteziologija, reanimatologija i intenzivno liječenje	04-06-2018

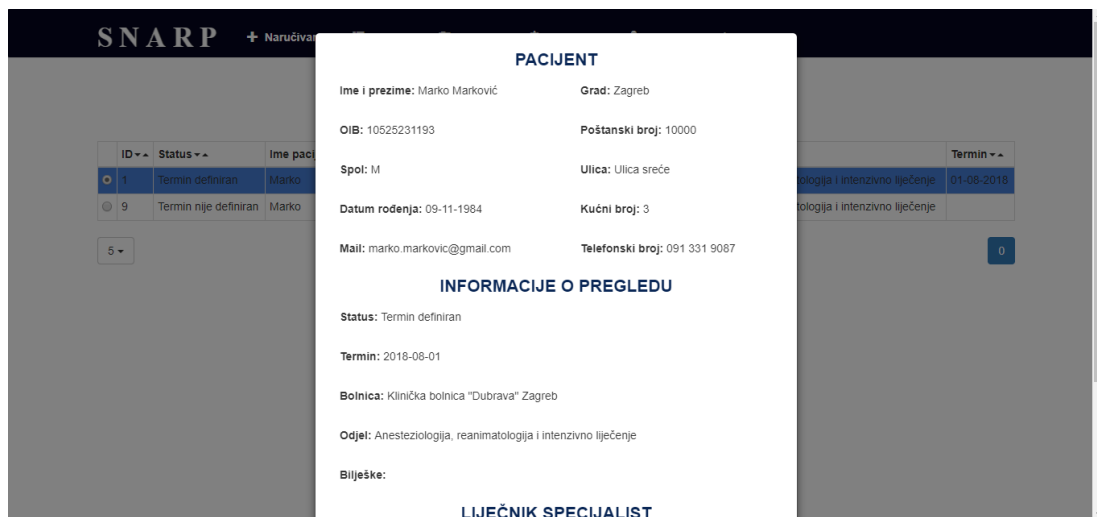
5
0

Pregled

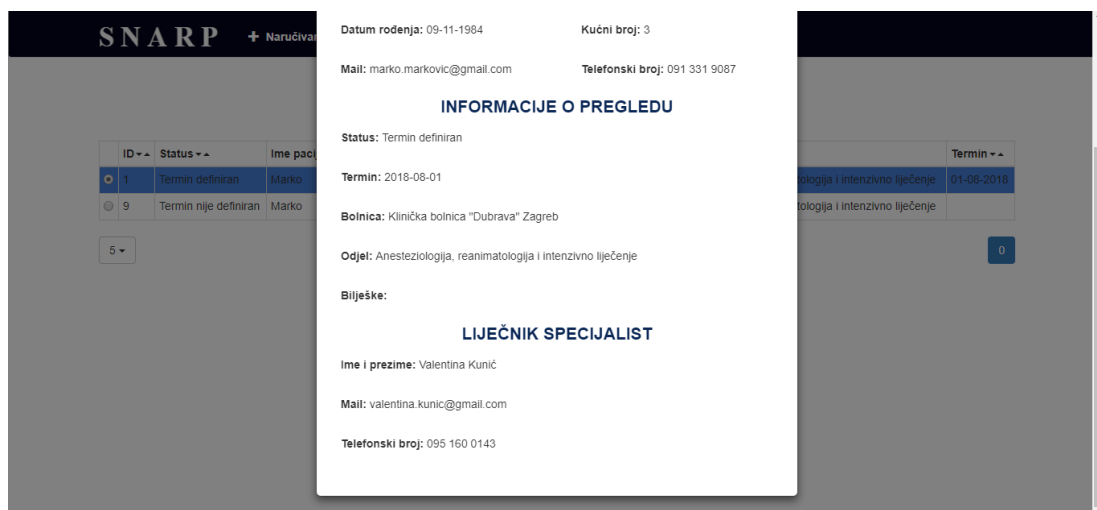
Slika 25. Pregled uputnica, pacijenti

## Detaljan uvid u informacije o pregledu

Pacijentima je omogućen i detaljan uvid u informacije vezane uz pregled do kojega se dolazi odlaskom na stranicu *Pregledi*, odabere se pregled u tablici i nakon toga odabere gumb *Pregled*.



Slika 26. Detaljan uvid u preglede, prvi dio



Slika 27. Detaljan uvid u preglede, drugi dio

## Detaljan uvid u informacije o uputnicama

Pacijentima je omogućen i detaljan uvid u informacije o uputnici do kojega se dolazi odlaskom na stranicu *Uputnice*, odabere se uputnica u tablici i nakon toga odabere gumb *Pregled*.

The screenshot shows a web application interface with a dark header containing the logo 'SNARP' and a '+ Naručiva' button. A table on the left lists patients, with 'Marko Marković' selected. A modal window titled 'PACIJENT' displays the following information:

Ime i prezime: Marko Marković	Grad: Zagreb
OIB: 10525231193	Poštanski broj: 10000
Spol: M	Ulica: Ulica sreće
Datum rođenja: 09-11-1984	Kućni broj: 3
Mail: marko.markovic@gmail.com	Telefonski broj: 091 331 9087

Below the patient information, the form is titled 'UPUTNICA ZA' and contains the following details:

- Vrsta uputnice: A – Konzilijarna zdravstvena zaštita
- Podvrsta uputnice: A1 - Konzilijarni pregled
- Odjel: Anesteziologija, reanimatologija i intenzivno liječenje
- Dijagnoza: Naevus naevocellularis pigmentosus
- Napomena: Molim hitno
- Datum kreiranja: 04-06-2018

Slika 28. Detaljan pregled uputnice, prvi dio

This screenshot shows the second part of the medical referral form. The modal window is titled 'UPUTNICA ZA' and contains the same information as the previous screenshot. Below this, a new section titled 'SPECIJALIST OBITELJSKE MEDICINE' displays the following information:

Ime i prezime: Luka Novak	Grad: Zagreb
Mail: luka.novak@gmail.com	Poštanski broj: 10000
Telefonski broj: 091 331 9087	Ulica: Abramovićeve
	Kućni broj: 12

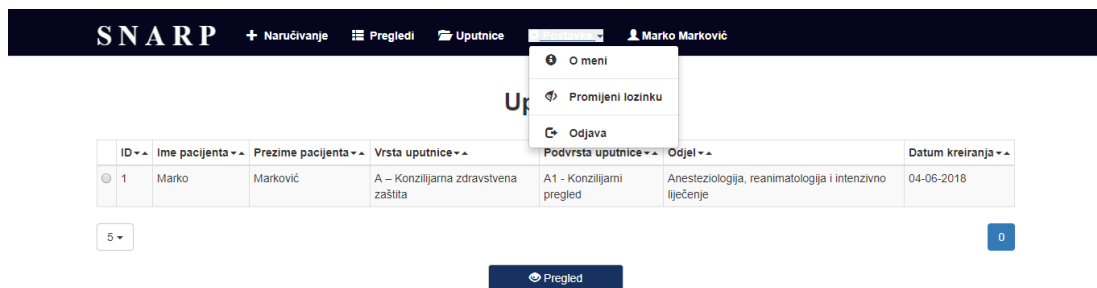
Slika 29. Detaljan pregled uputnice, drugi dio



### 3.3.6. Postavke sustava

#### Postavke

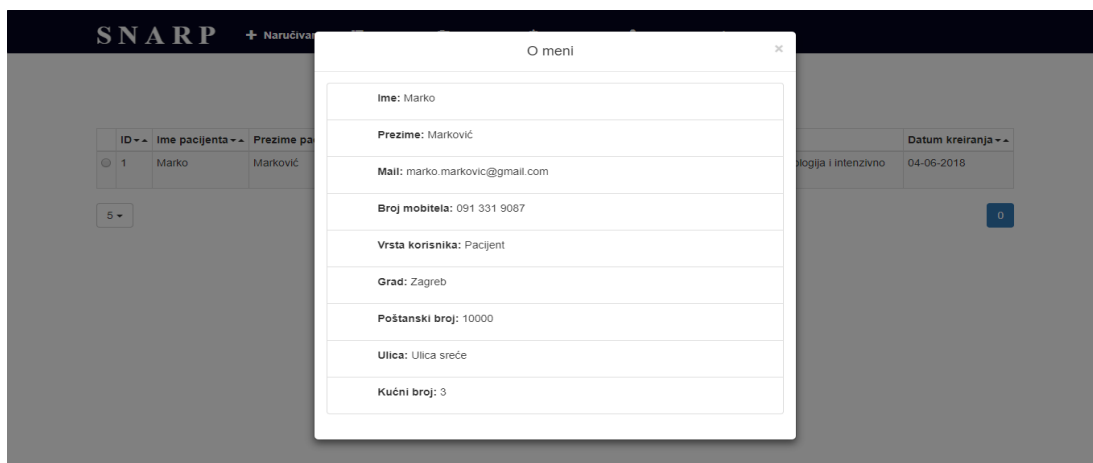
Odlaskom na *Postavke* u navigacijskoj traci otvara se izbornik koji nudi tri opcije. Prva opcija je *O meni* pomoću koje je moguće vidjeti osobne podatke. Druga opcija je *Promijeni lozinku* pomoću koje je moguće promijeniti trenutnu lozinku. Treća opcija je *Odjava* pomoću koje se korisnik odjavljuje iz aplikacije. Pristup postavkama imaju svi korisnici aplikacije.



Slika 30. Postavke

## Pregled osobnih podataka

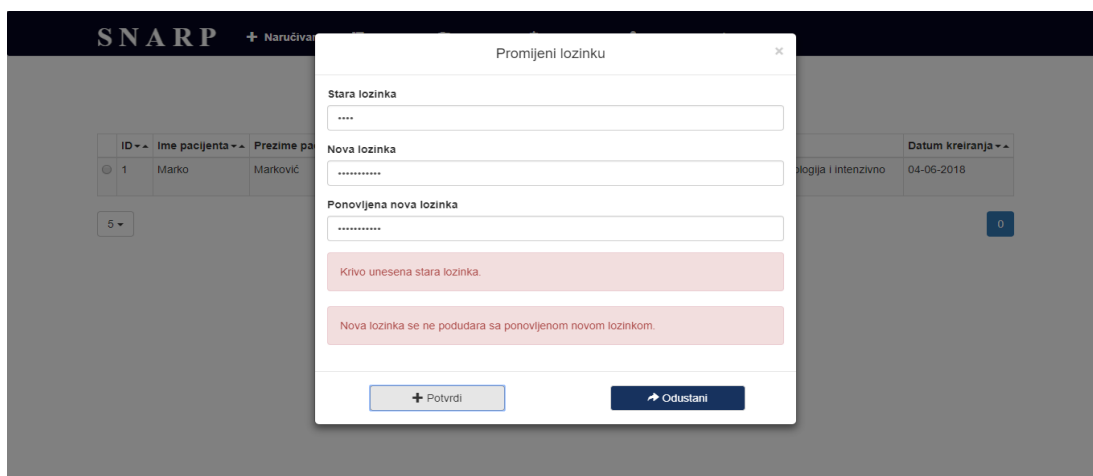
Odabirom opcije *O meni* iz izbornika postavki prikazuju se osobni podaci korisnika.



Slika 31. Pregled osobnih podataka

## Promjena lozinke

Odabirom opcije *Promjeni lozinku* iz izbornika postavki otvara se forma za promjenu lozinke. Za promjenu lozinke potrebno je upisati staru lozinku, novu lozinku i ponovljenu novu lozinku.



Slika 32. Promjena lozinke

## 4. Zaključak

Izdavanje uputnica i naručivanje za preglede putem web aplikacije značajno može olakšati i ubrzati proces naručivanja i organizaciju pregleda. Web aplikacija razvijena u sklopu ovog rada ispunjava većinu potrebne funkcionalnosti i može se efikasno koristiti za unapređenje spomenutog procesa. Idući korak rada na sustavu bila bi implementacija obavješćavanja pacijenata o terminima pregleda. Bilo bi korisno kad bi se pacijentu mogle slati poruke u trenutku kada je za njega određen termin za pregled, a isto tako bilo bi poželjno slati poruke nekoliko dana prije pregleda da ga podsjetite na pregled. Pacijentu bi se dala mogućnost odabira želi li primiti takve poruke. Dala bi mu se mogućnost odabira koliko dana unaprijed želi primiti podsjetnik za pregled. Također, mogao bi birati kojim komunikacijskim kanalom želi primiti poruke, želi li da mu se pošalje e-mail ili sms poruka.

## 5. Literatura

1. Walls, Craig: *Spring in Action, Fourth Edition*, New York: Manning Publications, 2014
2. Banks, Alex; Porcello, Eve: *Learning React*, Sebastopol, California: O'Reilly, 2017
3. Bugl, Daniel: *Learning Redux*, Birmingham, United Kingdom: Packt Publishing, 2017
4. Bloch, Joshua: *Effective Java 3rd Edition*, Boston, United States: Addison-Wesley, 2018
5. React, s Interneta, <https://reactjs.org/> , 2018
6. Spring, s Interneta, <https://spring.io/>, 2018
7. React-Bootstrap, s Interneta, <https://react-bootstrap.github.io/>, 2018
8. H2 Database Engine, s Interneta, <http://www.h2database.com/html/main.html>, 2018

## 6. Popis slika

Slika 1. Dijagram obrazaca uporabe .....	3
Slika 2. Skica sustava .....	4
Slika 3. Model baze podataka .....	6
Slika 4. Dijagram razreda paketa <i>controller</i> .....	11
Slika 5. Dijagram razreda paketa <i>service</i> .....	15
Slika 6. Dijagram razreda paketa <i>repository</i> .....	16
Slika 7. Dijagram razreda paketa <i>domain</i> .....	20
Slika 8. Dijagram razreda paketa <i>enumeration</i> .....	22
Slika 9. Razred <i>MedicalSpecialistController</i> .....	26
Slika 10. Razred <i>MedicalSpecialistService</i> .....	27
Slika 11. Razred <i>MedicalSpecialistRepository</i> .....	28
Slika 12. Razred <i>MedicalSpecialist</i> .....	29
Slika 13. Početna stranica .....	30
Slika 14. Korisničko sučelje administratora .....	31
Slika 15. Dodavanje zapisa o specijalistu obiteljske medicine .....	32
Slika 16. Uređivanje podataka o specijalistu obiteljske medicine .....	33
Slika 17. Brisanje zapisa o bolnici .....	34
Slika 18. Pregled zapisa o pacijentima .....	35
Slika 19. Pregled uputnica .....	36
Slika 20. Dodavanje uputnice .....	37
Slika 21. Pregled zapisa o pregledima .....	38
Slika 22. Određivanje termina pregleda .....	39
Slika 23. Naručivanje na preglede.....	40
Slika 24. Pregled naručenih pregleda kakve vide pacijenti.....	41
Slika 25. Pregled uputnica, pacijenti.....	41
Slika 26. Detaljan uvid u preglede, prvi dio .....	42
Slika 27. Detaljan uvid u preglede, drugi dio .....	42
Slika 28. Detaljan pregled uputnice, prvi dio.....	43
Slika 29. Detaljan pregled uputnice, drugi dio .....	43
Slika 30. Postavke .....	44
Slika 31. Pregled osobnih podataka .....	45
Slika 32. Promjena lozinke .....	45

## 7. Naslov, sažetak i ključne riječi na hrvatskom jeziku

### 7.1. Naslov

Web aplikacija za naručivanje pacijenata

### 7.2. Sažetak

U radu *Web aplikacija za naručivanje pacijenata* opisan je razvoj sustava za naručivanje pacijenata. Uvodni dio sadrži kratak opis problema i motivaciju za rješavanje problema. Nakon uvoda slijedi poglavlje u kojemu su opisani arhitektura i dizajn sustava. U ovom poglavlju nalaze se zahtjevi na programski sustav, skica sustava i dijagrami razreda s opisom. U idućem poglavlju opisana je implementacija i korisničko sučelje. Navedene i opisane su korištene tehnologije i alati. Priloženi su isječci programskog koda vezani za temeljnu funkcionalnost. Na kraju rada nalaze se zaključak, literatura i popis slika.

### 7.3. Ključne riječi

Java, Spring Boot, React.js, Redux, H2, web aplikacija, administrator, pacijent, liječnik

## **8. Naslov, sažetak i ključne riječi na engleskom jeziku**

### **8.1. Naslov**

Web application for patients appointment booking

### **8.2. Sažetak**

This thesis *Web application for patients appointment booking* describes the construction of a patient ordering system. The introductory section contains a brief description of problem and motivation to solve the problem. In the next chapter architecture and design of the system are described. This chapter also contains requirements for a program system, system sketch and class diagrams with description. The next chapter describes the implementation and the user interface. Technologies and tools used in project are described too. Programming code snippets of the basic functionality are attached. At the end of the paper there is a conclusion, literature and a list of images.

### **8.3. Ključne riječi**

Java, Spring Boot, React.js, Redux, H2, web application, administrator, patient, doctor