**Veb aplikacija za zubarsku ordinaciju**

Ovo je projekat za ispit iz predmeta Praktikum – Internet i veb tehnologije.

Broj indeksa: 2018203764

Ime i prezime: Svetozar Stanković

Školska godina: 2021/2022

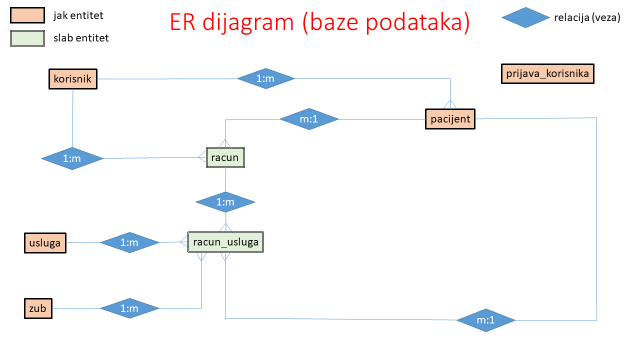
# Projektni zahtev

Realizovati veb baziranu aplikaciju za zubarsku ordinaciju. Aplikaciji može da pristupi samo prijavljeni korisnik sa korisničkim imenom i lozinkom. Svaki korisnik sistema ima svoje pristupne parametre. Aplikacija treba da vodi evidenciju o svim prijava korisnika, kao i o svim neuspelim prijavama na sistem (pogrešno uneta lozinka naloga) ili pogrešno uneto korisničko ime u formularu za prijavu. Kada se korisnik prijavi na aplikaciju, u panelu za uređivanje sadržaja može da doda, izmeni ili sakrije određenu uslugu zubarske ordinacije. Svaka usluga ima naziv, opis, kataloški broj, kategoriju usluge (preventivna intervencija, redovna intervencija, hirurška intervencija, itd), cenu za izdvojeno pružanje usluge, cenu za pružanje usluge u paketu, cenu za pružanje usluge za decu i cenu za pružanje usluge za penzionere. U sistemu korisnik ima pristup podacima o pacijentima. Može da doda podatke o pacijentu ili da menja već unete podatke, kao i da odjavi pacijenta iz spiska aktivnih pacijenata ordinacije. Svakom pacijentu se otvara elektronski karton u kojem administrator ima prikaz zuba u vilici u obliku šeme (raspored). Korisnik (zubar) po završenoj intervenciji u aplikaciju unosi spisak pruženih usluga i obeležava na šemi vilice nad kojim zubom ili zubima je vršio intervenciju i obeležava koja je usluga u pitanju, a informacioni sistem na kraju računa automatski cenu na osnovu toga da li je pružena jedna usluga ili više, pa ih računa na osnovu cene u paketu i da li je pacijent dete, penzioner ili ne spada u jednu od te dve kategorije pacijenata. U kartonu pacijenta se mogu videti sve intervencije i pružene usluge koje su pružane od kako je pacijent upisan u informacioni sistem veb aplikacije. Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.

# Tehnička ograničenja

* + Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Express biblioteke. Aplikacija mora da bude podeljena u dve nezavisne celine: back-end veb servis (API) i front-end (GUI aplikacija). Sav kôd aplikacije treba da bude organizovan u dva Git spremišta u okviru istog korisničkog naloga za ovaj projekat.
  + Baza podataka mora da bude relaciona i treba koristiti MySQL ili MariaDB sistem za upravljanje bazama podataka (RDBMS) i u spremištu back-end dela aplikacije mora da bude dostupan SQL dump strukture baze podataka, eventualno sa inicijalnim podacima, potrebnim za demonstraciju rada projekta.
  + Back-end i front-end delovi projekta moraju da budi pisani na TypeScript jeziku, prevedeni TypeScript prevodiocem na adekvatan JavaScript. Back-end deo aplikacije, preveden na JavaScript iz izvornog TypeScript koda se pokreće kao Node.js aplikacija, a front-end deo se statički servira sa rute statičkih resursa back-end dela aplikacije i izvršava se na strani klijenta. Za postupak provere identiteta korisnika koji upućuje zahteve back-end delu aplikacije može da se koristi mehanizam sesija ili JWT (JSON Web Tokena), po slobodnom izboru.
  + Sav generisani HTML kôd koji proizvodi front-end deo aplikacije mora da bude 100% validan, tj. da prođe proveru W3C Validatorom (dopuštena su upozorenja - Warning, ali ne i greške - Error). Grafički korisnički interfejs se generiše na strani klijenta (client side rendering), korišćenjem React biblioteke, dok podatke doprema asinhrono iz back-end dela aplikacije (iz API-ja). Nije neophodno baviti se izradom posebnog dizajna grafičkog interfejsa aplikacije, već je moguće koristiti CSS biblioteke kao što je Bootstrap CSS biblioteka. Front-end deo aplikacije treba da bude realizovan tako da se prilagođava različitim veličinama ekrana (responsive design).
  + Potrebno je obezbediti proveru podataka koji se od korisnika iz front-end dela upućuju back-end delu aplikacije. Moguća su tri sloja zaštite i to: (1) JavaScript validacija vrednosti na front-end-u; (2) Provera korišćenjem adekvatnih testova ili regularnih izraza na strani servera u back-end-u (moguće je i korišćenjem izričitih šema - Schema za validaciju ili drugim pristupima) i (3) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.
  + Neophodno je napisati prateću projektnu dokumentaciju o izradi aplikacije koja sadrži (1) model baze podataka sa detaljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram baze podataka; (3) dijagram organizacije delova sistema, gde se vidi veza između baze, back-end, front-end i korisnika sa opisom smera kretanja informacija; (4) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije po ISO standardu.
  + Izrada oba dela aplikacije (projekata) i promene kodova datoteka tih projekata moraju da bude praćene korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git spremištu, npr. na besplatnim GitHub ili Bitbucket servisima, jedno spremište za back-end projekat i jedno za front-end projekat. Ne može ceo projekat da bude otpremljen u samo nekoliko masovnih Git commit-a, već mora da bude pokazano da je projekat realizovan u kontinuitetu, da su korišćene grane (branching), da je bilo paralelnog rada u više grana koje su spojene (merging) sa ili bez konflikata (conflict resolution).

# Baza podataka



**korisnik**

korisnik\_id **INT UN PK AI UQ NN**

korisnicko\_ime  **VC 64 UQ NN**

lozinka\_hash **VC 128 NN**

ime **VC 64 NN**

prezime **VC 64 NN**

email **VC 64 NN**

jmbg **VC 13 NN UQ**

created\_at **TS CT**

is\_active **ENUM(aktivan,neaktivan) D(neaktivan)**

aktivacioni\_kod **VC255 D(NULL) UQ**

**prijava\_korisnika**

prijava\_korisnika\_id **INT UN PK AI UQ NN**

logged\_at **TS CT**

status **TY 1 UN D(1)**

korisnicko\_ime **VC 64**

lozinka\_hash **VC 128**

**usluga**

usluga\_id **INT UN PK AI UQ NN**

naziv **VC 128 UQ NN**

opis **TEXT**

sifra\_usluge **VC 64 UQ NN**

kategorija **ENUM(preventivna,redovna,hirurska) NN**

cena\_pojedinacna\_dete

**FLOAT D(0) NN UN**

cena\_pojedinacna\_penzioner

**FLOAT D(0) NN UN**

cena\_pojedinacna\_ostali

**FLOAT D(0) NN UN**

cena\_paket\_dete **FLOAT D(0) NN UN**

cena\_paket\_penzioner **FLOAT D(0) NN UN**

cena\_paket\_ostali **FLOAT D(0) NN UN**

status **ENUM(aktivna,neaktivna) D(aktivna) NN**

**pacijent**

pacijent\_id **INT UN PK AI UQ NN**

ime **VC 64 NN**

prezime **VC 64 NN**

jmbg **VC 13 NN UN UQ**

adresa **VC 128 NN**

telefon **VC 64 NN**

email **VC 64 NN**

status **ENUM(aktivan,neaktivan) D(aktivan) NN**

senioritet **ENUM(dete,penzioner,ostali) D(ostali) NN**

korisnik\_id **FK INT UN NN**

**zub**

zub\_id **INT UN PK AI UQ NN**

broj **INT (1-8) NN**

vilica

**ENUM(gornja,donja) NN**

tip **ENUM(sekutic,ocnjak,kutnjak) NN**

strana **ENUM(leva,desna) NN**

sifra\_zuba **VC 64 UQ**

**racun**

racun\_id **INT UN PK AI UQ NN**

created\_at **TS CT**

tip\_usluge **ENUM(pojedinacna,paket) NN**

senioritet **ENUM(dete,penzioner,ostali) NN**

pacijent\_id **FK INT UN NN**

korisnik\_id **FK INT UN NN**

**racun\_usluga**

racun\_usluga\_id **INT UN PK AI UQ NN**

zub\_id **FK INT UN NN**

usluga\_id **FK INT UN NN**

pacijent\_id **FK INT UN NN**

racun\_id **FK INT UN NN**

## #Legenda simbola:

## INT – integer (celobrojna) vrednost

## FLOAT – decimalna vrednost

## ENUM() – nabrojivi tip

## VC – Varchar vrednost (niz znakova)

## TEXT – tekst

## TY 1 – tiny integer vrednost velicine 1

## UN – unsigned (vrednost bez znaka – pozitivan broj)

## NN – not null (vrednost ne sme imati NULL vrednost)

## AI – auto increment (automatsko uvecavanje vrednosti za 1)

## UQ – unique (jedinstvena vrednost – ne sme se ponavljati)

## PK – primary key (primarni kljuc)

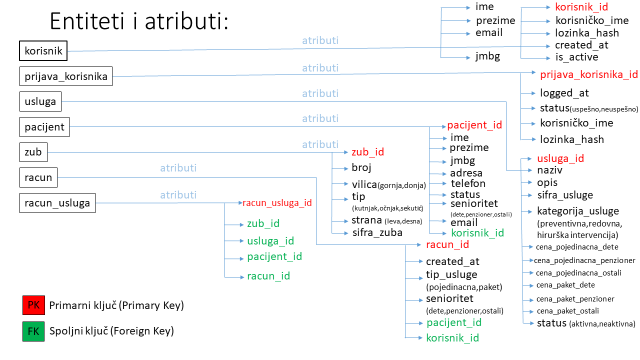
## FK – foreign key (spoljni kljuc)

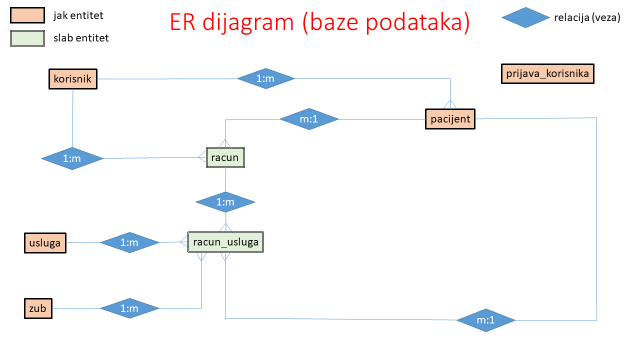
## D() – default (podrazumevana) vrednost

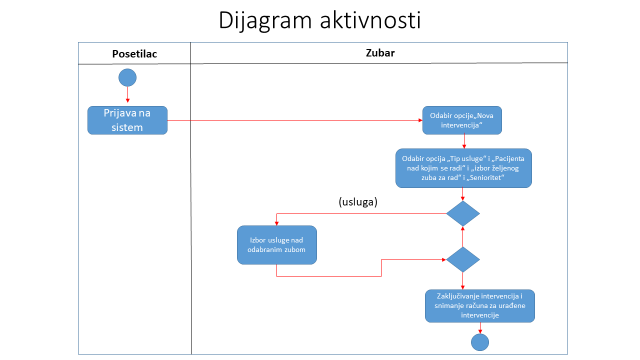
# Use-Case dijagrami , ER dijagram i dijagrami aktivnosti

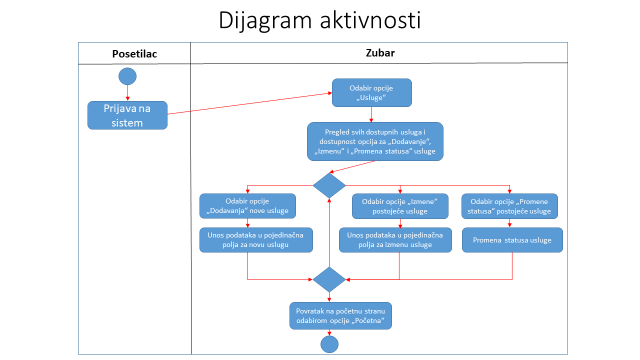
Ukoliko ne možete jasno da pročitate neke delove dijagrama aktivnosti, svi dijagrami su kreirani u zasebnoj powerpoint prezentaciji pod nazivom “Analiza zahteva - Use case dijagram , Dijagram aktivnosti, ER dijagram.pptx” koja je dostupna u okviru foldera gde se nalazi dokumentacija projekta

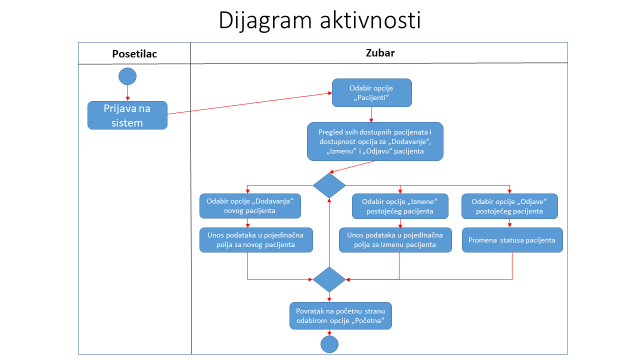


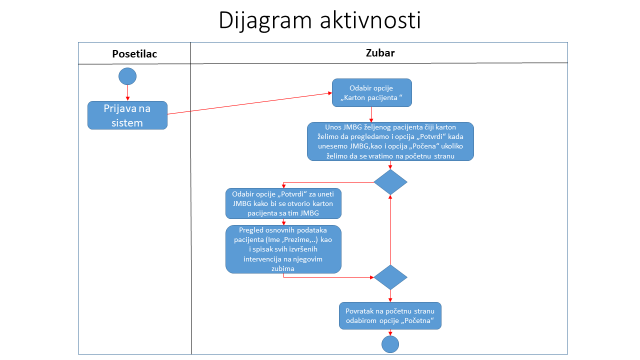




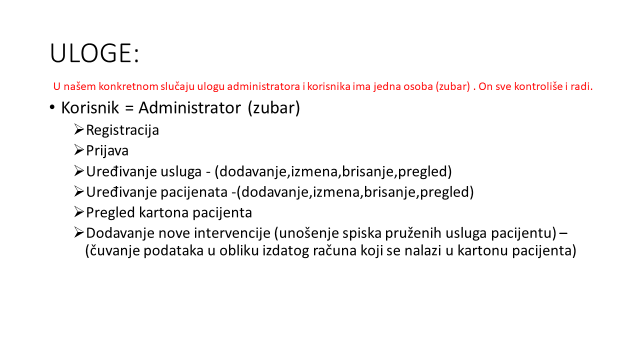








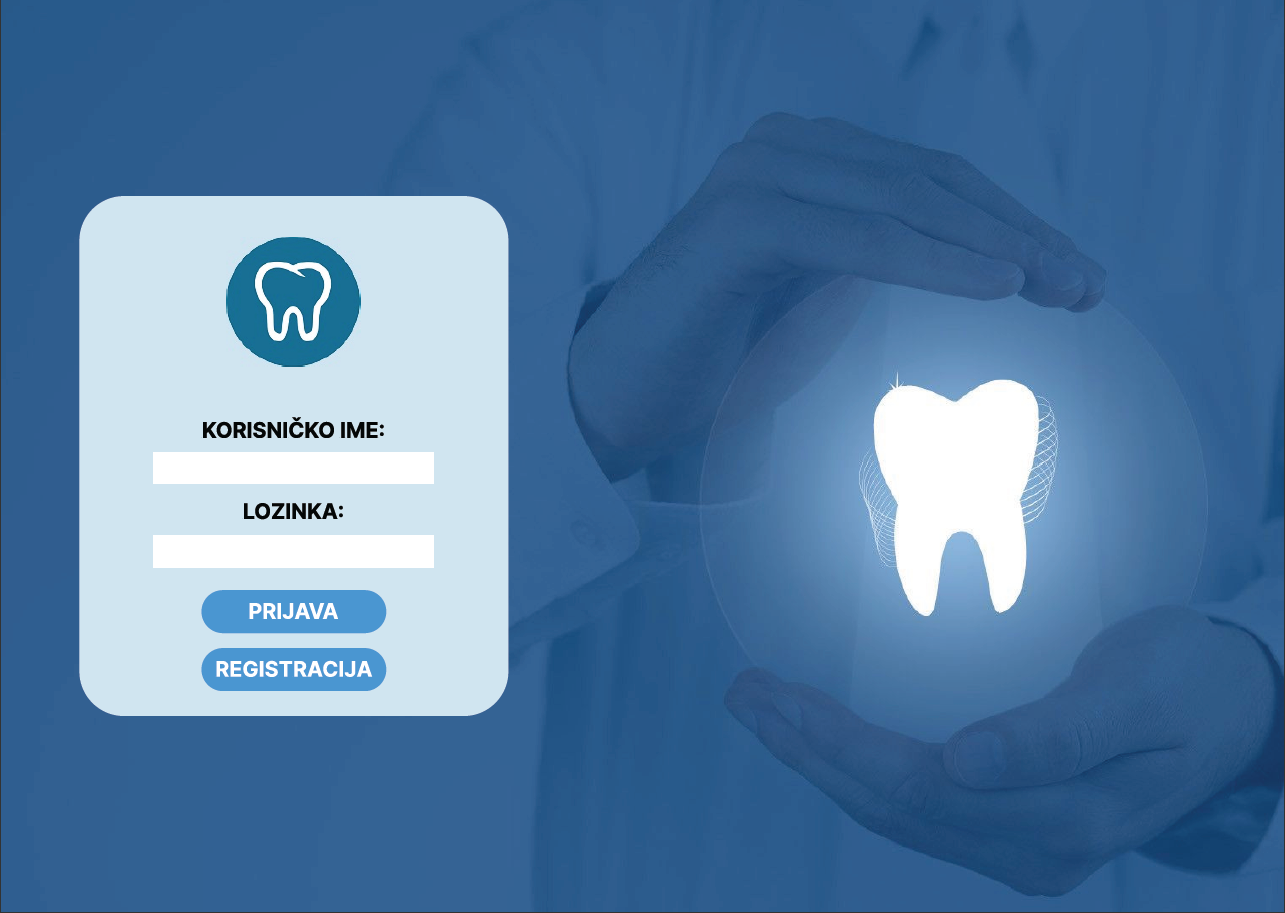
# Uloge korisnika





# Dizajn prototipa front-end dela aplikacije (rađeno u Figma)

#1



Klikom na dugme **prijava** ukoliko su uneti parametric validni prikazuje se Dashboard korisnika koji je prikazan na slici #2.2

Klikom na dugme **registracija** prikazuje vam se ekran sa formom za pravljenje novog naloga (prikazano na slici #2.1)

#2.1



Klikom na dugme **odustani** vraćate se na početnu stranu za prijavu (slika #1)

Popunjavanjem potrebnih podataka u formi i klikom na dugme **kreiraj nalog** kreira se novi nalog i tom prilikom Vam se šalje aktivacioni mejl na Vašu e-mail adresu kako biste taj novokreirani nalog aktivirali i prikazuje vam se ekran sa obaveštenjem da ste uspešno kreirali nalog (kao na slici #2.1.1)

#2.1.1



Dobili ste obaštenje da je nalog uspešno kreiran i klikom na dugme **prijava** vraćate se na početnu stranu za prijavu (kao na slici #1)

#2.2



Nakon što ste se uspešno prijavili sa svojim nalogom , prikazuje vam se ovakav Dashboard.

Pritiskom na dugme **odjava** se odjvaljujete i vraćate se na početnu stranu za prijavu (kao na slici #1)

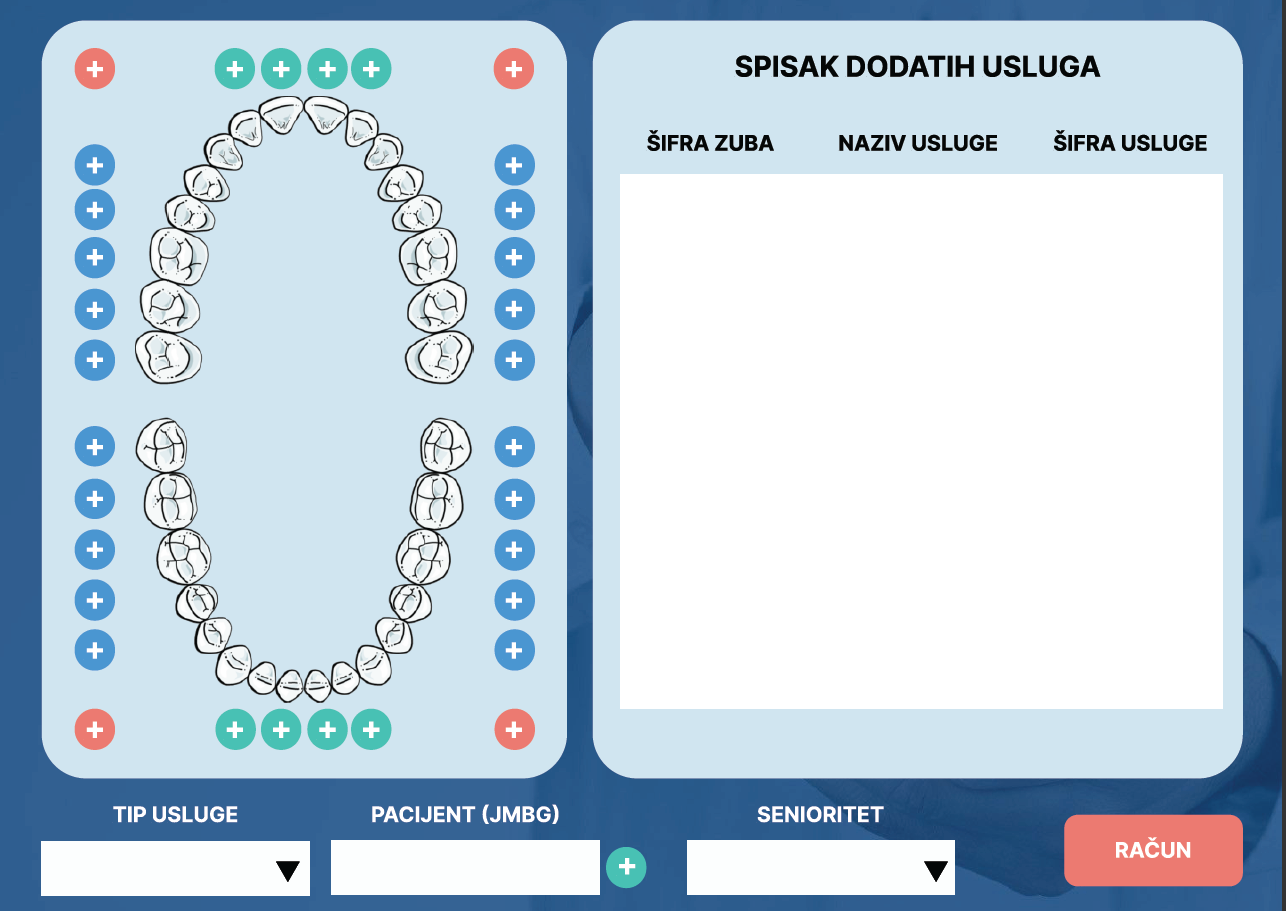
Pritiskom na dugme **usluge** ,otvara se strana sa spiskom svih usluga dostupnih u sistemu (kao na slici #5)

Pritiskom na dugme **pacijenti**, otvara se stranica sa spiskom svih dostupnih pacijenata u sistemu (kao na slici #6)

Pritiskom na dugme **karton pacijenta,** otvara se stranica sa poljem za unos jmbg broja pacijenta čiji karton želimo da pogledamo (kao na slici #4)

Pritiskom na dugme **nova intervencija**, otvara se stranica sa prikazom šeme vilice spiskom urađenih(dodatih) intervencija za pacijenta zajedno ssa podešavanjima tipa usluge i senioriteta pacijenta (kao na slici #3).

#3



Pritiskom na određene **+ dugmiće** za željeni zub otvara se dijalog tog zuba gde možete uneti šifru usluge koja je odrađena na tom zubu, a koja će biti prikazana ovde u “Spisku dodatih usluga”. (kaon a slici #3.1)

U dnu stranice možete uneti podatke kao što si tip usluge (paket,pojedinčna), senioritet (dete,penyioner,ostali) I JMBG pacijenta na kom se vrše intervencije

Takođe, pored polja za unos JMBG broja pacijenta postoji **zeleno + dugme** koje otvara novu stranicu sa spiskom svih dostupnih pacijenata u sistemu,ako želimo da saznamo brzo dodatne informacije o pacijentima . (kaona slici #7)

U donjem desnom uglu se nalazi dugme **račun** pomou koga se formira cena na osnovu unetih parameter usluga, i otvara se nova stranica za potvrdu računa sa spiskom svih izvršenih usluga ,podacima o pacijentu i ukupnoj ceni. (kao na slici #8)

#4



Klikom na dugme **nazad** se vraća na prethodnu stranu , Dashboard , kao na slici #2.2

Unosom JMBG broja pacijenta čiji karton se želi pogledati i pritiskom na dugme **potvrdi** , se odlazi na stranicu gde je prikazan karton tog željenog pacijenta sa informacijama od svim izvršenim intervencijama na tom pacijentu. (kao na slici #4.1)

#5



Pritiskom na dugme **početna** se vraćamo na prethodnu stranicu u kontrolni panel, Dashboard . (kao na slici #2.2)

Pritiskom na dugme **dodaj** se otvara stranica za dodavanje nove usluge. (kao na slici #5.3)

Pritiskom na dugme **izmeni** se otvara stranica za izmenu usluge (kao na slici #5.2)

Pritiskom na dugme **(de)aktiviraj** se otvar stranica za unos šifre usluge čiji status želimo da promenimo (naktivna->aktivna ili aktivna->neaktivna) , kao na slici #5.1

#6



Pritiskom na dugme **početna** se vraćamo na prethodnu stranicu u kontrolni panel, Dashboard . (kao na slici #2.2)

Pritiskom na dugme **dodaj** se otvara stranica za dodavanje novog pacijenta. (kao na slici #6.3)

Pritiskom na dugme **izmeni** se otvara stranica za izmenu pacijenta (kao na slici #6.2)

Pritiskom na dugme **(de)aktiviraj** se otvar stranica za unos jmbg broja pacijenta čiji status želimo da promenimo (naktivan->aktivan ili aktivan->neaktivan) , kao na slici #6.1

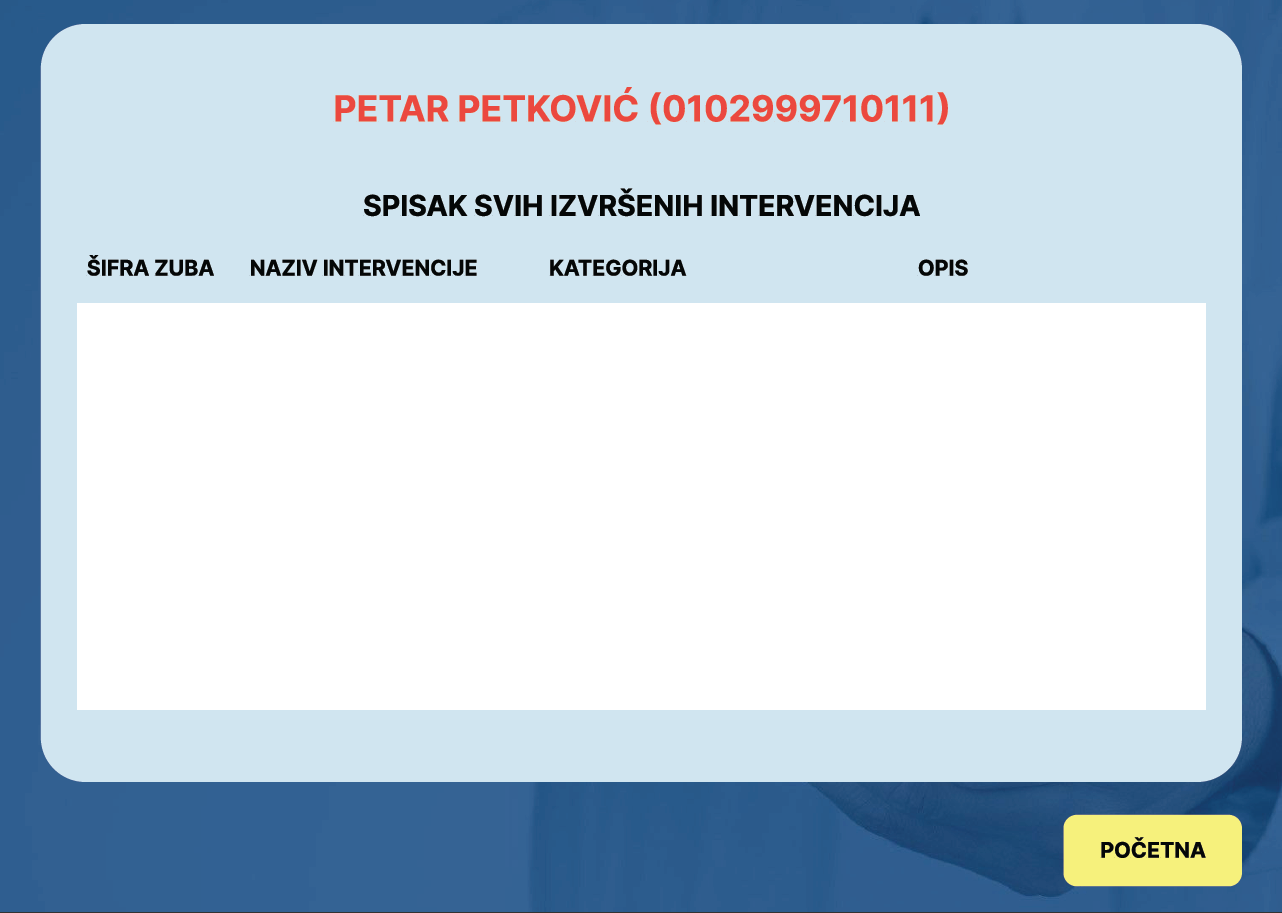
#3.1



Pritiskom na dugme **spisak svih usluga** otvara vam se stranica sa spiskom svih dostupnih usluga u sistemu i njihovim informacijama. (kao na slici #5)

Unosom odgovarajuće izvršene ***šifre usluge*** nad tim zubom i pritiskom na dugme **dodaj**, vraćate se na stranicu sa prikazom šeme vilice za dalje dodavanje usluga ili zaključenje računa , ali sada sa prikazanom dodatom uslugom u spisku izvršenih usluga sa informacijam na kom zubu je intervencija urađena. (kao na slici #3)

#4.1



Pritiskom na dumge **početna** se vraća na početni kontrolni panel ,Dashboard. (kao na slici #2.2)

#6.1



Pritiskom na dugme **nazad** se vraćate na prethodnu stranicu i odustajete od promene statusa pacijenta . (kao na slici #6)

Unosom JMBG broja pacijenta čiji status želimo da promenimo i pritiskom na dugme **(de)aktiviraj** menjamo trenutni status pacijenta I vraćamo se na prethodnu stranicu. (kao na slici #6)

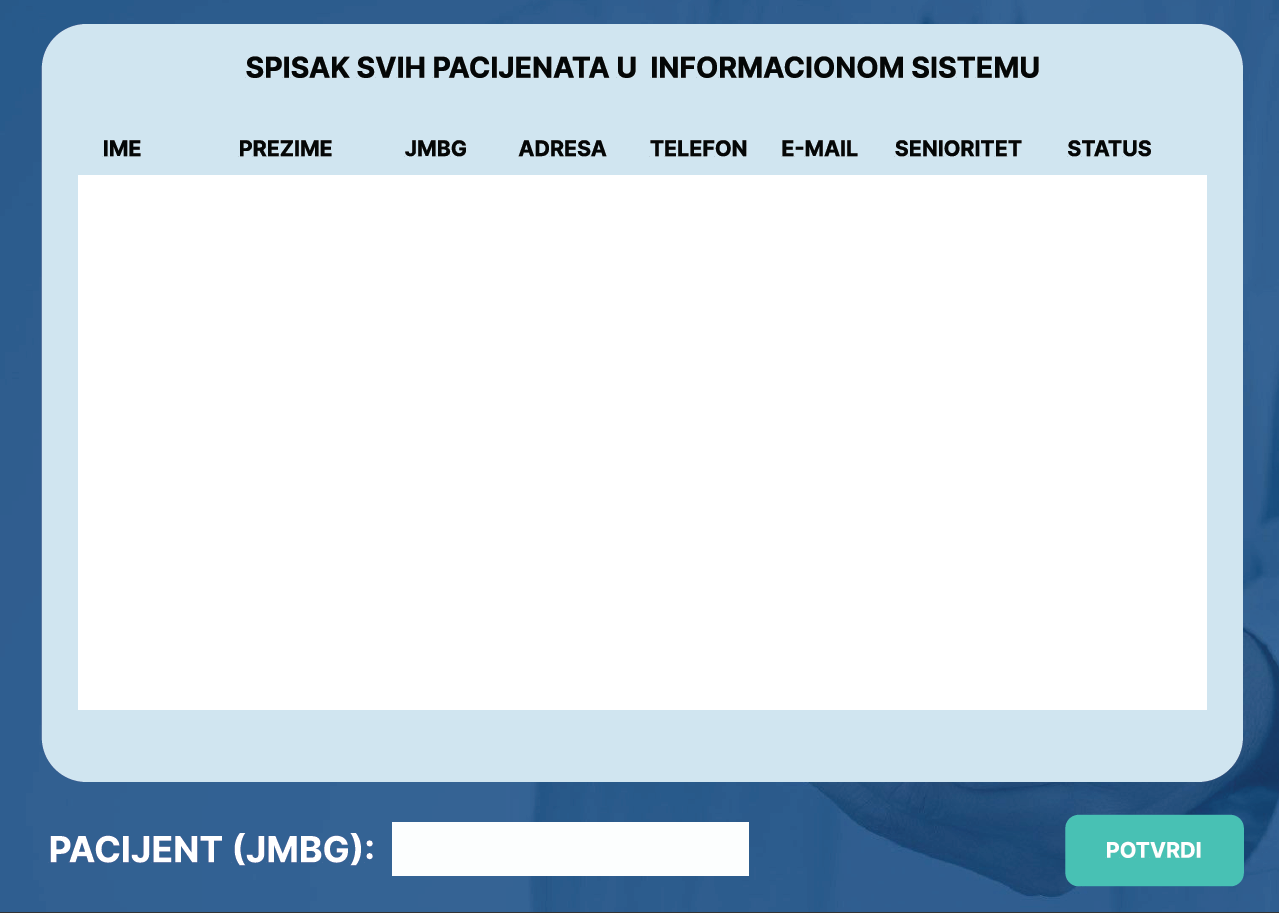
#5.1



Pritiskom na dugme **nazad** se vraćate na prethodnu stranicu i odustajete od promene statusa usluge . (kao na slici #5)

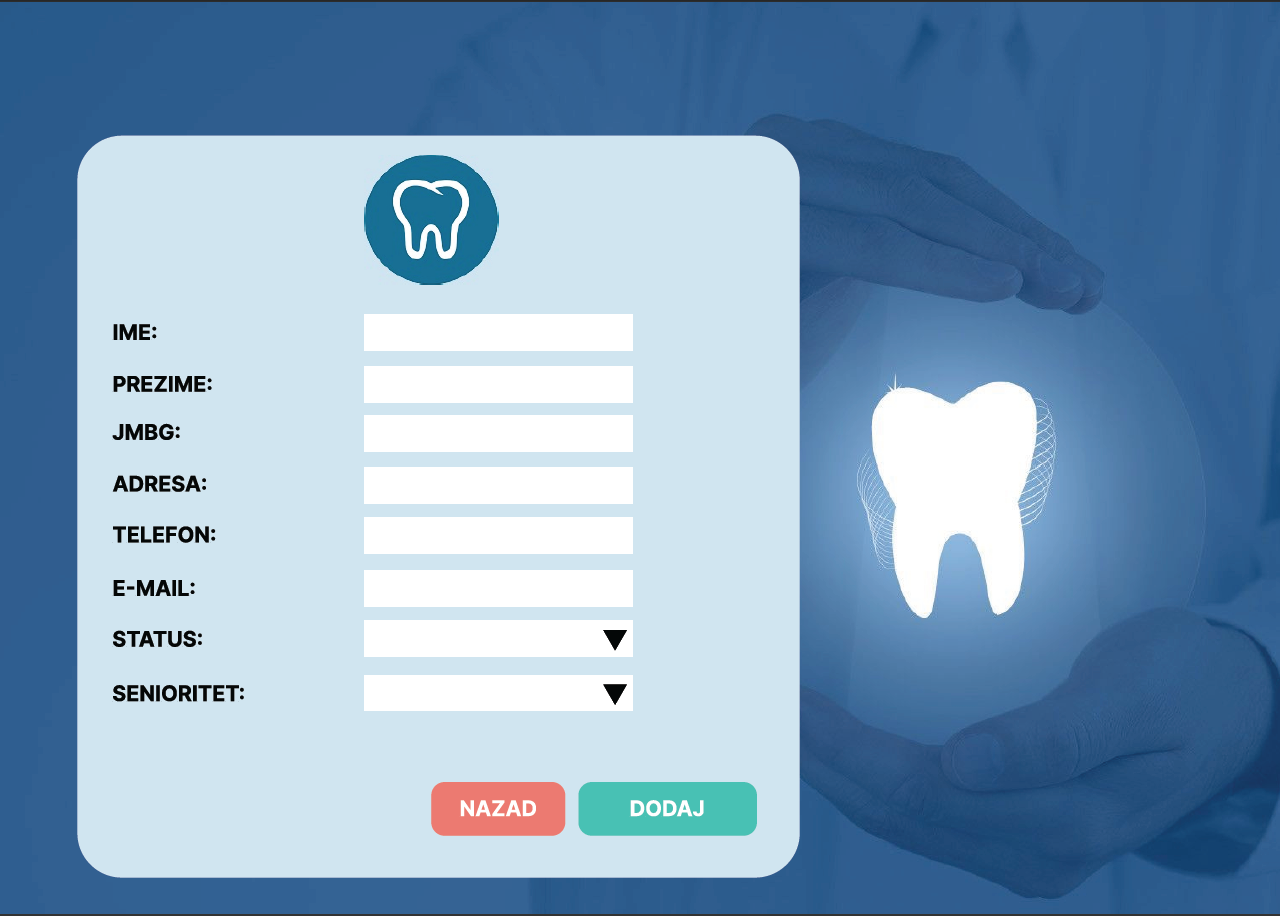
Unosom šifre usluge čiji status želimo da promenimo i pritiskom na dugme **(de)aktiviraj** menjamo trenutni status usluge i vraćamo se na prethodnu stranicu. (kao na slici #5)

#7



Klikom na dugme **potvrdi**  vraća se na prethodnu stranu sa prikazom šeme vilice gde se dodaju intervencije odrađene nad pacijentom. (kao na slici #3)

#6.3



Pritiskom na dugme  **nazad** se vraćamo na prethodnu stranu gde su prikazani svi pacijenti dostupni u sistemu. (kao na slici #6)

Popunjavanjem svih odgovarajćih polja forme i pritiskom na dugme **dodaj** se kreira novi pacijent u sistemu, i nakon toga se vraća na prethodnu stranu gde je prikazan spisak svih dostupnih pacijenata u sistemu. (kao na slici #6)

#5.3



Pritiskom na dugme  **nazad** se vraćamo na prethodnu stranu gde su prikazane sve usluge dostupne u sistemu. (kao na slici #5)

Popunjavanjem svih odgovarajćih polja forme i pritiskom na dugme **dodaj** se kreira nova usluga u sistemu, i nakon toga se vraća na prethodnu stranu gde je prikazan spisak svih dostupnih usluga u sistemu. (kao na slici #5)

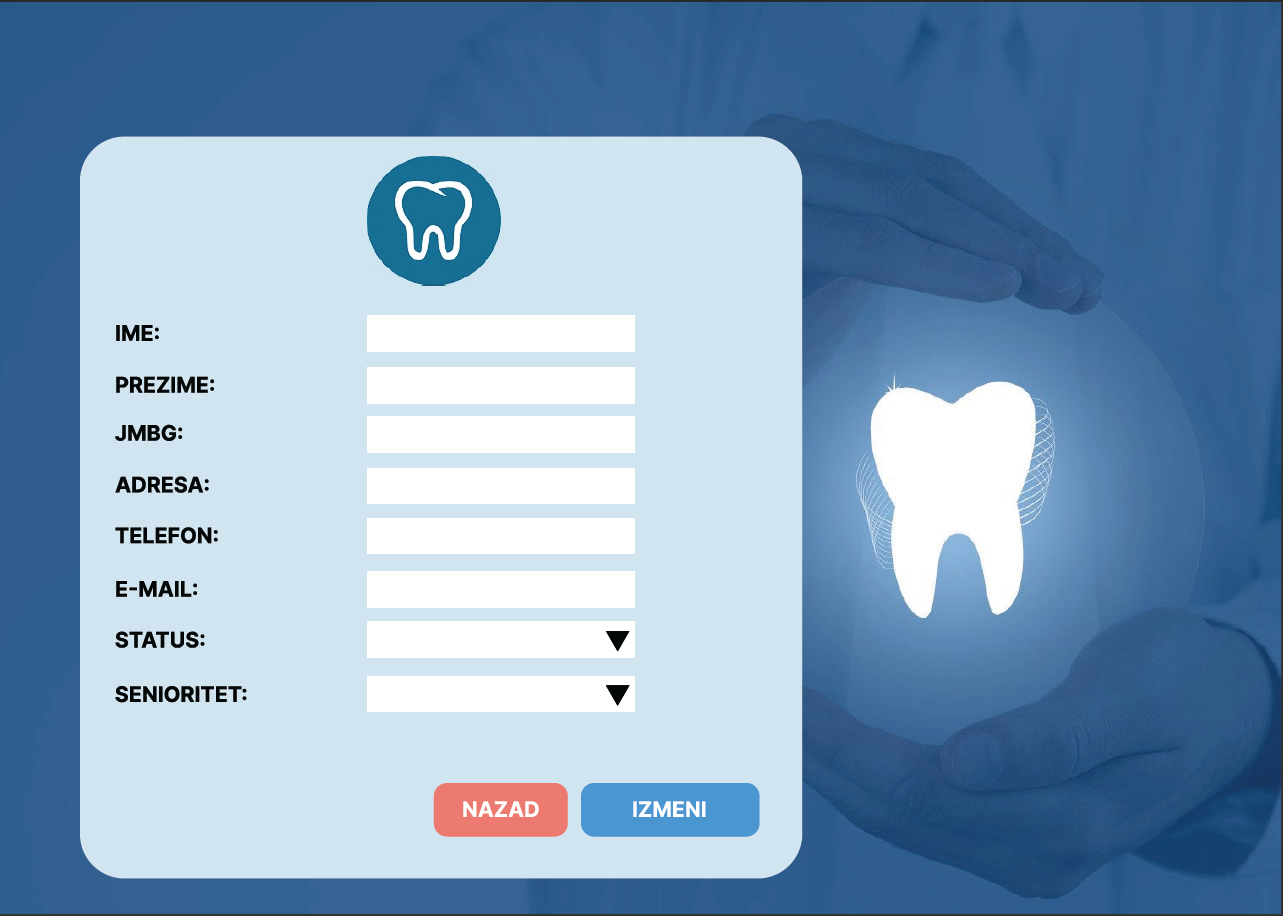
#8



Pritiskom na dugme **nazad** se vraća na prethodnu stranu sa prikazom vilice gde se doddaju izvršene intervencije nad pacijentom. (kao na slici #3)

Pritiskom na dugme **snimi** se zaključuje račun i odrađene intervencije nad pacijentom i sve potrebne informacije se čuvaju u bazi podataka sistema. Nakon toga se vraćamo na početni kontrolni panel, Dashboard. (kao na slici #2.2)

#6.2



Pritiskom na dugme  **nazad** se vraćamo na prethodnu stranu gde su prikazani svi pacijenti dostupni u sistemu. (kao na slici #6)

Popunjavanjem svih odgovarajćih polja forme i pritiskom na dugme **izmeni** se menjaju podaci o pacijentu u sistemu, i nakon toga se vraća na prethodnu stranu gde je prikazan spisak svih dostupnih pacijenata u sistemu. (kao na slici #6)

#5.2



Pritiskom na dugme  **nazad** se vraćamo na prethodnu stranu gde su prikazane sve usluge dostupne u sistemu. (kao na slici #5)

Popunjavanjem svih odgovarajćih polja forme i pritiskom na dugme **izmeni** se menjaju podaci usluge u sistemu, i nakon toga se vraća na prethodnu stranu gde je prikazan spisak svih dostupnih usluga u sistemu. (kao na slici #5)