

Sveučilište/Univerzitet Vitez
Fakultet informacionih tehnologija
Projektovanje informacijskih sistema

Informacioni sistem stomatološke ordinacije
Projektni zadatak

Godina studija: III
Student: Zulka Musić
Hanan Sejmen

Profesor: prof.dr. Branko Latinović

Viši asis: mr. Vernes Vinčević, dipl.ing.info

Travnik, akademska 2024./2025.

SADRŽAJ

1.	UVOD.....	3
2.	ORGANIZACIONI DIJAGRAM STOMATOLOŠKE ORDINACIJE	4
3.	OPIS I PLAN PROJEKTA.....	5
4.	KORISNIČKI ZAHTEVI	6
5.	SISTEMSKA ANALIZA PROBLEMA.....	7
5.1.	USE CASE DIJAGRAMI.....	7
5.2.	DIJAGRAMI AKTIVNOSTI	9
5.3.	DIJAGRAMI KLASA	12
6.	SISTEMSKI DIZAJN.....	14
6.1.	PROJEKAT BAZE PODATAKA	14
6.2.	DIZAJN FORMI I IZVJEŠTAJA PUTEM MOCKUP-A	16
6.3.	DIZAJN INTERFEJSA I DIJALOGA	20
7.	ARHITEKTURA SISTEMA.....	22
8.	BACKEND.....	23
8.1.	C# SA WINDOWS FORMS	23
8.2.	JAVA SA SPRINGBOOT-OM	24
9.	FRONTEND.....	26
10.	TESTIRANJE INFORMACIONOG SISTEMA.....	28
11.	ZAKLJUČAK	29

1. UVOD

U ovom projektu ćemo predstaviti razvoj sistema za stomatološku ordinaciju, koja proširuje svoj biznis. Ideja samog informacionog sistema je da omogući zubarima da lakše pronađe karton pacijenta (digitalni karton), te pacijentima omogućiti da preko web aplikacije zakažu termin kod zubara.

Sistem će pružiti pacijentima lakše zakazivanje termina preko web aplikacije, ali moći će i uživo i preko telefona zakazati isto termin. Pacijentu prilikom zakazivanja termina bit će ponuđeni zubari, prilikom izabiranja zubara prikazat će mu se slobodni termini (datum i vrijeme). Kada pacijent bude naručivao termin morat će upisati svoje ime i prezime, datum rođenja i da obilježi da li je prije bio ili mu je prvi dolazak kod zubara. Pacijent će moći da upiše uslugu koju želi (popravlak zuba, vađenje zuba, čišćenje kamenca, konsultacije, kontrola i slično), te morat će još upisati svoj e-mail i broj telefona. Potvrdu da je zakazao termin će dobiti na e-mail, ukoliko nešto pogriješi prilikom zakazivanja sistem će izbaciti grešku i pokazati šta nije uredu. Ovom sistemu pristup će imati administrator, medicinska sestra i pacijent. Administrator će moći da briše pacijente iz sistema kao i zubare iz ponude (ukoliko su dali otkaz i slično).

Na digitalnom kartonu zubar/medicinska sestra će imati pristup:

- imenu i prezimenu pacijenta,
- datumu rođenja,
- alergije,
- prethodno propisane lijekove od strane zubara,
- napomene (poput toga da pacijent pije tablete za razrijeđenje krvi i sl.),
- kad je zadnji put bio pacijent,
- šta je sve prethodno rađeno kod pacijenta.

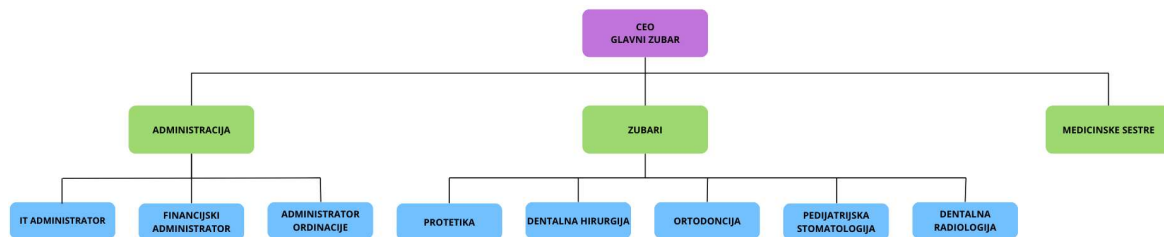
Ukoliko postoje pacijenti sa istim imenom i prezimenom pored svakog pacijenta će biti i datum rođenja. Administrator će imati pristup da briše karton pacijenta i zubar također. Medicinska sestra i zubar moći će da ažuriraju i pregledaju kartone pacijenata. Medicinska sestra i zubar će imati login podatke za pristup sistemu za kartone.

Ovaj sistem je kreiran za lakše praćenje termina i nalaženja kartona pacijenta, umjesto tradicionalnih kartona u ručno pisanom formatu. Sa sistemom se štedi vrijeme zubarima kao i pacijentima, čak se brine i okolini da se više ne pišu kartoni na papirima. Sistem će pružiti pretraživanja pacijenata sa njegovim kartonom putem desktop aplikacije, a pacijentu će biti omogućeno zakazivanje termina putem web aplikacije.

U nastavku ovog projektnog rada obrađeni su detalji razvoja informacionog sistema stomatološke ordinacije. Glavni cilj ovog projekta je dizajniranje i razvoj interaktivnog sistema za stomatološku ordinaciju. Predstavljeni su dijagrami koji prikazuju različite funkcionalnosti korisnika na sistemu, te način na koji svaki korisnik izvršava istu. Dotakli smo se i arhitekture sistema i baze podataka. Opisan je način implementacije informacionog sistema, uz predstavljanje tehnologija koje bi se koristile prilikom izrade samog sistema.

2. ORGANIZACIONI DIJAGRAM STOMATOLOŠKE ORDINACIJE

Na slici ispod je prikazan organizacioni dijagram stomatološke ordinacije, koji urađen u Canvi koja besplatna online aplikacija. Na čelu stomatološke ordinacije se nalazi CEO koji je ujedno i glavni zubar u stomatološkoj ordinaciji. U sklopu ordinacije imamo još administraciju, zubare i medicinske sestre. U administraciji imamo zaposlene IT administratora, financijskog administratora i administratora ordinacije. Uloga IT administratora je da sve servere i stranice drži pod kontrolom i obavlja redovne update sistema. Financijski administrator vodi brigu o finansijama i pravilnom upravljanju novca ordinacije. Administrator ordinacije ima ulogu da prima pacijente, uvodi u ordinacije i naplaćuje usluge. Što se tiče zubara, pošto se radi o naprednijoj stomatološkoj ordinaciji imamo i više vrsta zubara, a to su: protetičar, dentalni hirurg, ortodont, pedijatrijski zubar i radiologa. Protetičar ima zadatak da pravi zubne proteze, dentalni hirurg vrši teže dentalne zahvate gdje je potrebna operacija (tipa na korijenu odstraniti cistu), ortodont ima zadatak da vodi brigu o pacijentima koji nose fiksne ili mobilne aparatiće, pedijatrijski zubar ima zadatak da vodi brigu o zubima i higijeni najmlađih pacijenata, te dentalni radiolog ima zadatak da pacijentima slika pojedinačno zub ili čitavu vilicu.



Slika 1. Organizacioni dijagram stomatološke ordinacije

3. OPIS I PLAN PROJEKTA

Oduvijek su zubari imali pune ruke posla, da bi sebi skratili vrijeme da ne pretražuju po papirima šta se zadnji put radilo kod pacijenta, moći će na sistem upisati ime i prezime pacijenta i imat će pristup čitavom kartonu pacijenta. Danas je potrebno manje vremena za izgradnju i razvoj informacionih sistema prema potrebama korisnika. Modul koji će biti razvijen sastojat će se od nekog osnovnog šablona svih informacionih sistema.

Što se tiče samog planiranja projekta, plan projekta je napravljen u besplatnom alatu ProjectLibre. Proces razvoja ovog sistema trajao bi oko tri mjeseca, a nakon razvoja i implementacije klijentu će biti ponuđena mogućnost održavanja sistema na tri godine, uz mogućnosti produženja istog.

		Name	Duration	Start	Finish	Resource Names
1	👤	☐ Pripremna faza	3 days	10/11/24 8:00 AM	10/15/24 5:00 PM	Menadžer projekta
2		Ideja	2 days	10/11/24 8:00 AM	10/14/24 5:00 PM	Menadžer projekta
3		Analiza zahtjeva	3 days	10/11/24 8:00 AM	10/15/24 5:00 PM	Menadžer projekta
4		Okvirno određivanje budžeta	1 day	10/11/24 8:00 AM	10/11/24 5:00 PM	Menadžer projekta
5		Formiranje tima	1 day	10/11/24 8:00 AM	10/11/24 5:00 PM	Menadžer projekta
6	👤	☐ Faza planiranja	1.5 days	10/15/24 8:00 AM	10/16/24 1:00 PM	Menadžer projekta
7	👤	☐ Detaljna analiza ideje	1.5 days	10/15/24 8:00 AM	10/16/24 1:00 PM	Menadžer projekta
8		Okvirno određivanje trajanja projektiranja	1 day	10/15/24 8:00 AM	10/15/24 5:00 PM	Menadžer projekta
9		Određivanje resursa	1.5 days	10/15/24 8:00 AM	10/16/24 1:00 PM	Menadžer projekta
10		Podjela zaduženja	1 day	10/15/24 8:00 AM	10/15/24 5:00 PM	Menadžer projekta
11	👤	☐ Faza izvedbe	50 days	10/16/24 8:00 AM	12/24/24 5:00 PM	Dizajner baza podataka; Dizajner interfejsa; Hardver; Internet konekcija; Programer; Softver
12		☐ Dizajn sistema	50 days	10/16/24 8:00 AM	12/24/24 5:00 PM	
13		Dizajn baze podataka	25 days	10/16/24 8:00 AM	11/19/24 5:00 PM	Dizajner baza podataka
14		Programiranje sistema	35 days	10/16/24 8:00 AM	12/3/24 5:00 PM	Programer
15		Izrada interfejsa	20 days	10/16/24 8:00 AM	11/12/24 5:00 PM	Dizajner interfejsa
16		Vodjenje dokumentacije	50 days	10/16/24 8:00 AM	12/24/24 5:00 PM	Menadžer projekta
17	👤	☐ Završna faza	30 days	12/25/24 8:00 AM	2/4/25 5:00 PM	Tester sistema
18		Testiranje	30 days	12/25/24 8:00 AM	2/4/25 5:00 PM	Tester sistema
19		Korekcija	30 days	12/25/24 8:00 AM	2/4/25 5:00 PM	Tester sistema
20		Implementacija	15 days	12/25/24 8:00 AM	1/14/25 5:00 PM	Tester sistema
21	👤	Održavanje sistema	1,095 days	2/5/25 8:00 AM	4/17/29 5:00 PM	

Slika 2. Plan projekta

		Name	Type	Initials	Max. Units	Standard Rate	Overtime Rate	Cost Per Use	Accrue At	Base Calendar
1	👤	Menadžer projekta	Work	M	100%	\$10.00/hour	\$0.00/hour	\$0.00	Prorated	Standard
2	👤	Dizajner baza podataka	Work	DB	100%	\$9.00/hour	\$0.00/hour	\$0.00	Prorated	Standard
3	👤	Programer	Work	P	100%	\$10.00/hour	\$0.00/hour	\$0.00	Prorated	Standard
4	👤	Dizajner interfejsa	Work	DI	100%	\$9.00/hour	\$0.00/hour	\$0.00	Prorated	Standard
5	👤	Internet konekcija	Material	I		\$50.00		\$0.00	Prorated	
6	👤	Tester sistema	Work	T	100%	\$8.00/hour	\$0.00/hour	\$0.00	Prorated	Standard
7	👤	Hardver	Material	H		\$6000.00		\$0.00	Prorated	
8	👤	Softver	Material	S		\$500.00		\$0.00	Prorated	

Slika 3. Resursi projektnog plana

Plan projekta podijeljen je u nekoliko faza, a to su:

1. Pripremna faza – trajala bi tri dana, za to vrijeme je potrebno naći ideju, analizirati zahtjeve, odrediti okvirno budžet i formirati tim
2. Faza planiranja – detaljno bi se analizirala ideja i podijelila zaduženja
3. Faza izvedbe – trajala bi 50 dana, dizajnirao bi se sistem, a tokom izvršavanja bilo kojeg zadatka vodila bi se dokumentacija
4. Završna faza – trajala bi 30 dana, vršilo bi se testiranje i korekcije sistema i isti bi se implementirao.

4. KORISNIČKI ZAHTEJEVI

U razgovoru sa klijentom prikupljeni su korisnički zahtjevi. Klijent želi da ima:

- Desktop aplikaciju za kartone
- Web stranicu za pacijente
- Određene funkcionalnosti za sve tipove korisnika

Tipovi korisnika bili bi administrator, zubar, medicinska sestra i pacijent, u skladu s tim svaki ima određene funkcionalnosti na sistemu.

Admin:

- Mogućnost dodavanja zaposlenika i uklanjanja istih
- Mogućnost dodavanja i uklanjanja pacijenata
- Mogućnost ažuriranja zaposlenika i pacijenata
- Mogućnost pregleda svih korisnika
- Mogućnost pregleda termina i ažuriranja istih
- Mogućnost dodavanja novih doktora u sistem za termine

Zubar:

- Mogućnost pregleda kartona pacijenata
- Mogućnost dodavanja pacijenata i uklanjanje
- Mogućnost ažuriranja kartona pacijenta

Medicinska sestra:

- Mogućnost pregleda kartona pacijenata
- Mogućnost ažuriranja kartona pacijenta
- Mogućnost zakazivanja termina i pristupa istim

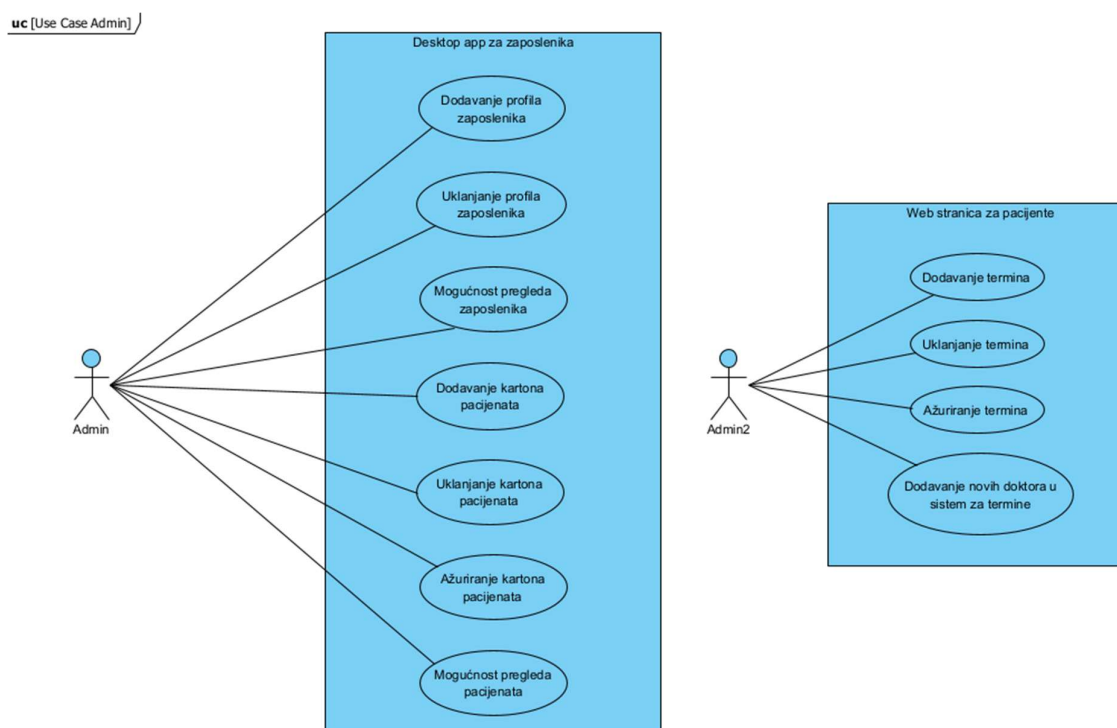
Pacijent:

- Mogućnost zakazivanja termina, kao i brisanja istog
- Mogućnost biranja zubara i željenog termina

5. SISTEMSKA ANALIZA PROBLEMA

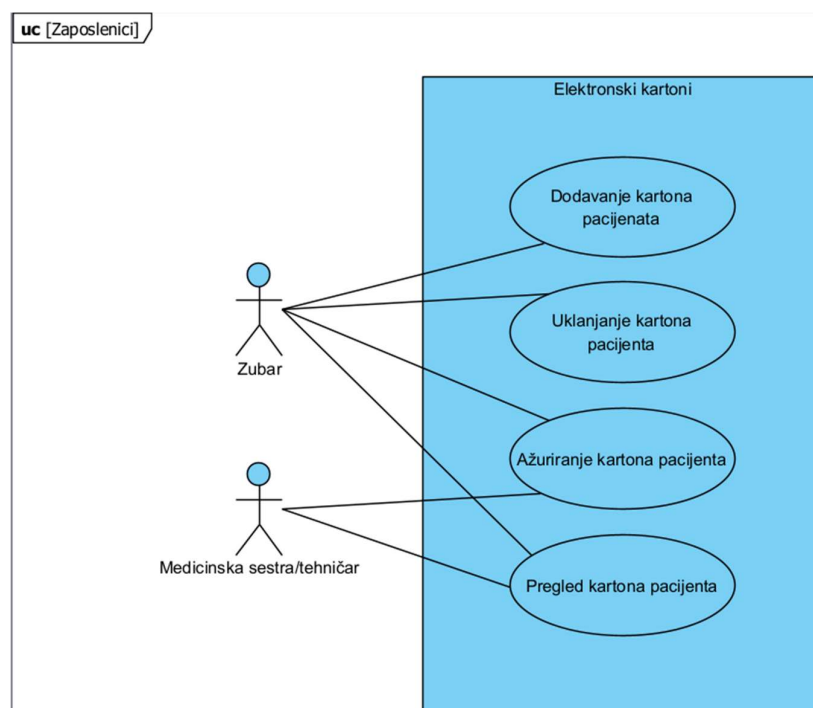
5.1. USE CASE DIJAGRAMI

Prvi use case dijagram će predstavljati ulogu administratora na sistemu. Administrator ima mogućnost da dodaje zaposlenike i uklanjanja istih, dodaje i uklanja pacijente, ažurira zaposlenike i pacijente, pregled svih korisnika, pregled termina i ažuriranja istih.



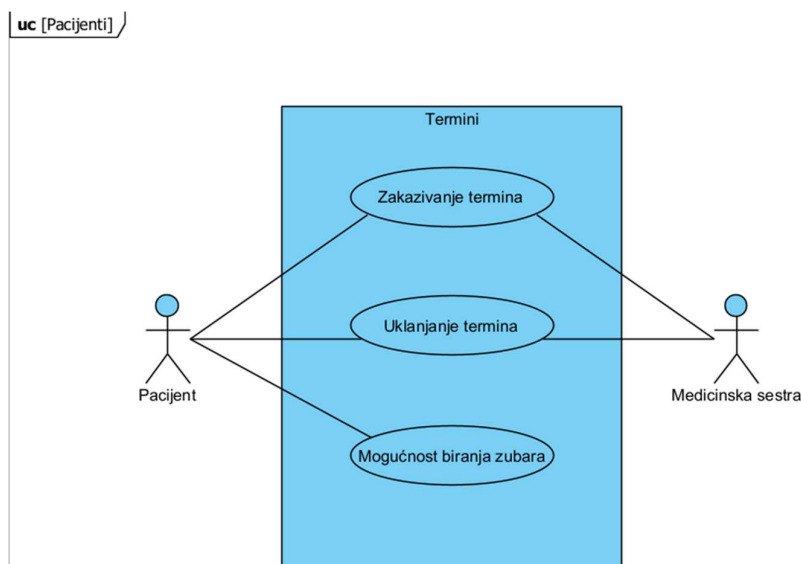
Slika 4. Use case dijagram za administratora

Drugi use case dijagram će predstavljati ulogu zubara i medicinske sestre na sistemu. Zubar će moći dodavati pacijente u sistem, ažurirati karton, pregledati karton, kao i brisati iste. Dok medicinska sestra će moći ažurirati karton i pregledati isti.



Slika 5. Use case dijagram kartoni

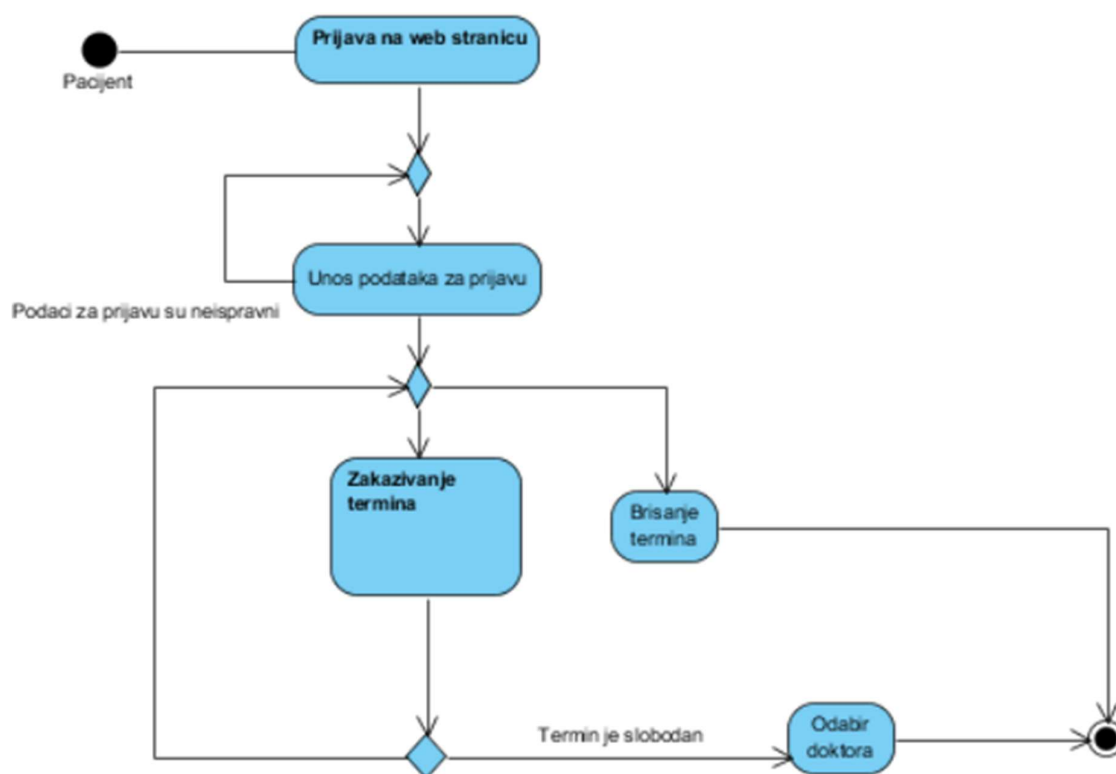
Treći use case dijagram predstavlja ulogu pacijenta u sistemu. Pacijent će moći zakazivati termin i brisati isti, te će mu biti omogućeno da bira zubara i željeni termin. Medicinska sestra će moći dodati termin i brisati iste.



Slika 6. Use case dijagram Termini

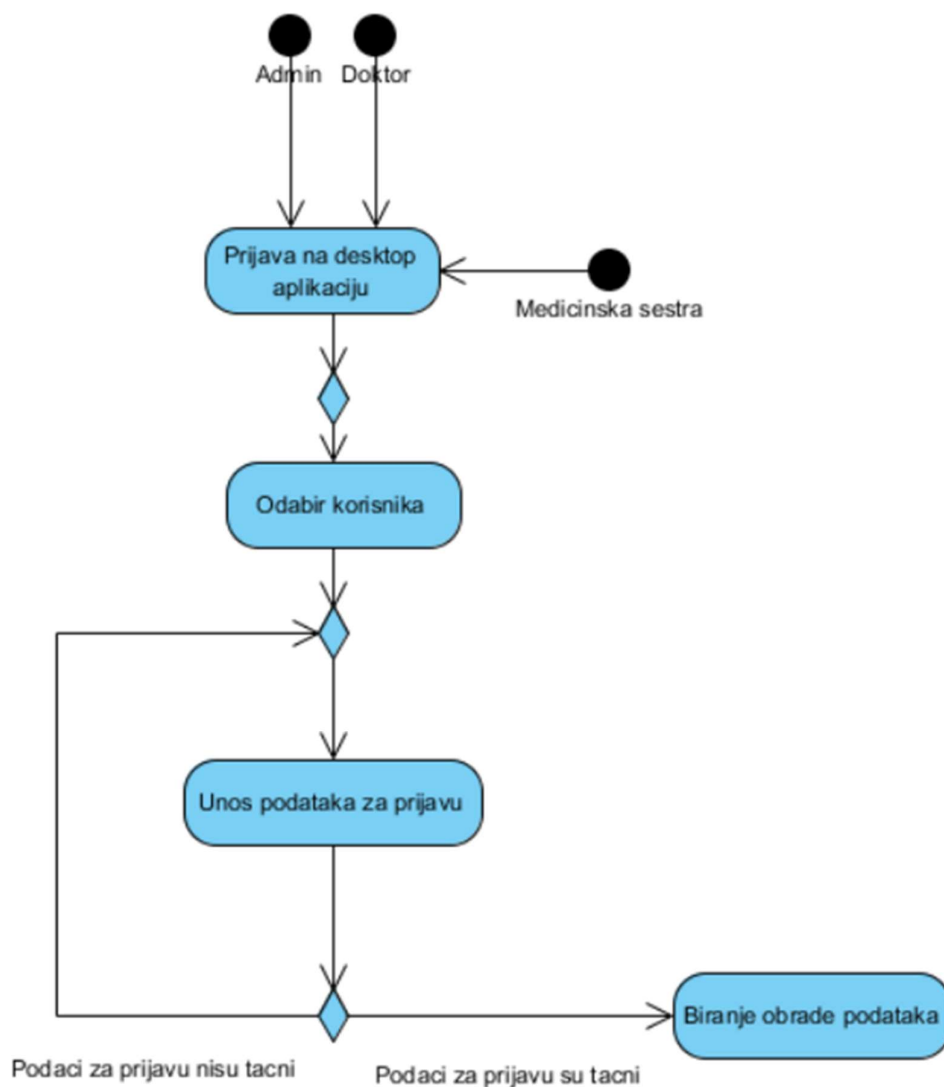
5.2. DIJAGRAMI AKTIVNOSTI

Sama aktivnost pacijenta na sistemu kreće kada uđe na samu web stranicu i upiše podatke za prijavu potom ima opciju da bira da li će zakazati termin ili obrisati zakazani termin. Kad klikne na dugme zakaži termin. Pacijent potom popunjava podatke ime i prezime, datum rođenja, da li je novi pacijent ili stari, moći će da upiše i uslugu koju želi, e-mail, broj telefona, moći će da izabere zubara kojeg želi te će mu sistem izbaciti koji termini su slobodni (datum i vrijeme). Provjerava se ispravnost unesenih podataka, te se šalje potvrda o uspješnosti na e-mail.



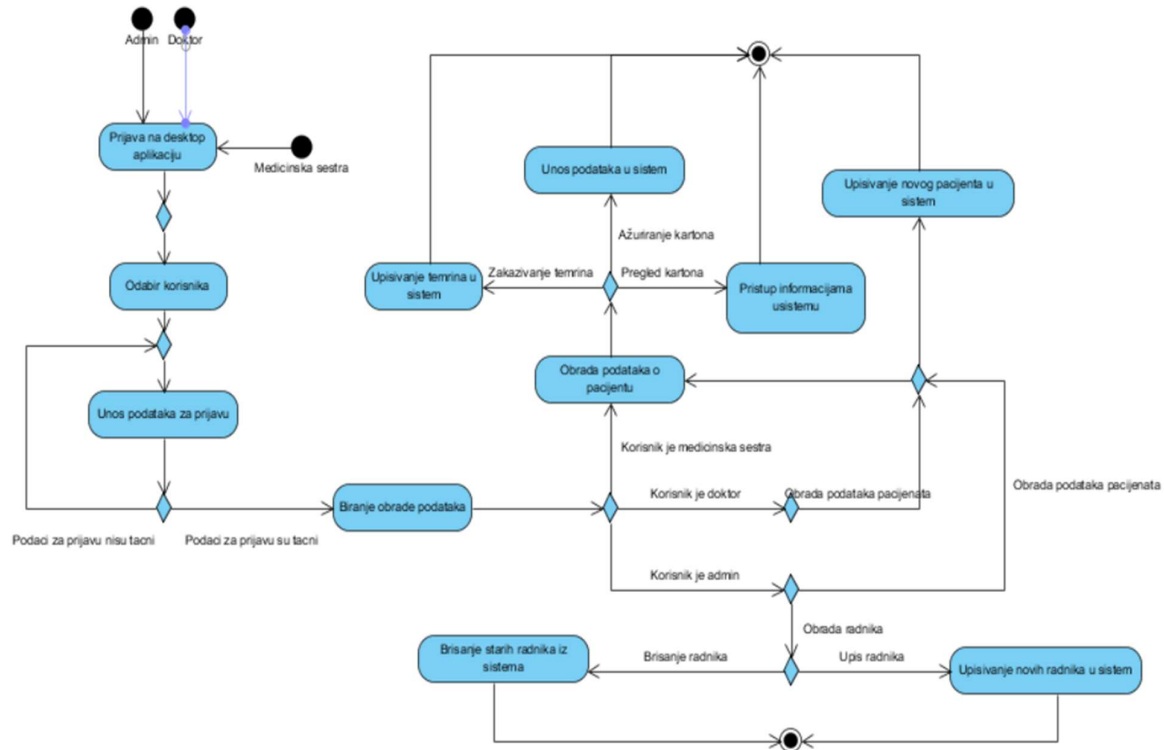
Slika 7. Dijagram aktivnosti za pacijenta

Sama aktivnost zubara/medicinske sestre na sistemu kreće kada uđu na desktop aplikaciju. Potom će se prijaviti na svoj korisnički nalog gdje će pisati svoj username i šifru. Vrš se provjera unesenih podataka.



Slika 8. Dijagram aktivnosti za login na desktop aplikaciju

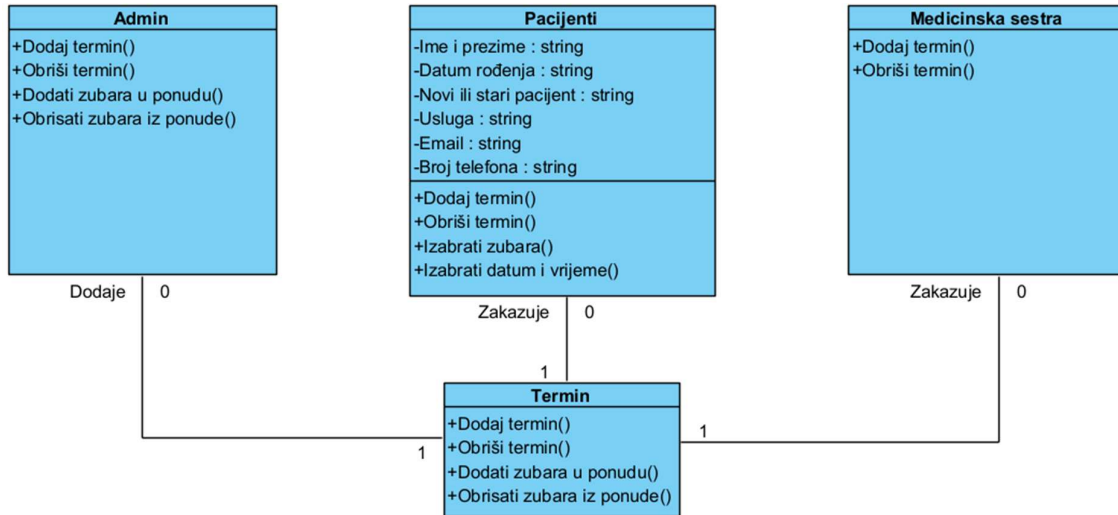
Tada će imati pristup kartonima ili da pravi novi karton. Prilikom pretraživanja od zubara/medicinske sestre se traži da upiše ime i prezime i datum rođenja pacijenta. Kada se sve upiše provjerava se ispravnost podataka, potom zubar/medicinska sestra imaju pristup kartonu. Ukoliko žele da ažuriraju karton imat će dugme ažuriraj i tad će moći dodati i promijeniti šta je potrebno, i morat će kliknut na opciju spremi karton. Ukoliko zubar treba napraviti novi karton od pacijenta će uzeti potrebne informacije i spremi karton.



Slika 9. Dijagram aktivnosti za desktop aplikaciju za elektronske kartone

5.3. DIJAGRAMI KLASA

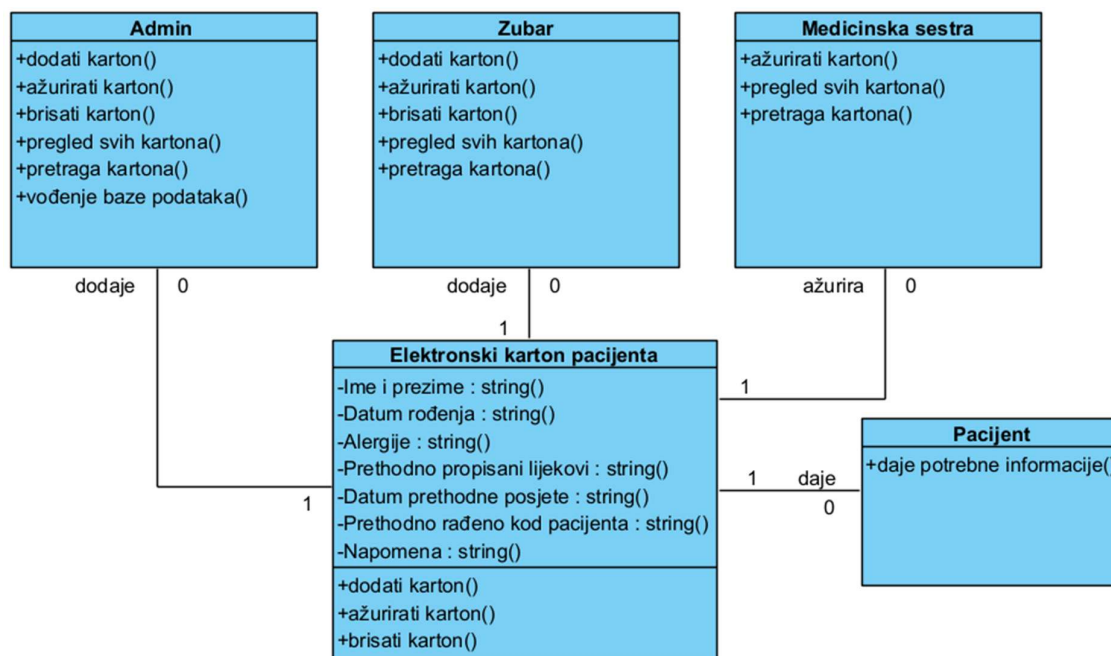
Prvi dijagram prikazuje elemente podataka u sistemu i odnos između njih što se tiče web aplikacije. Oni prikazuju statičku strukturu sistema.



Slika 10. Dijagram klasa Termin (za web aplikaciju)

Administrator može da doda termin na sistem, da izbriše termin i da dodaje ili briše zubare u ponudi. Medicinska sestra može da dodaje termin na sistem i da briše isti. Pacijent treba da popuni podatke koje sistem traži od njega, može da izabere zubara kojeg želi. Kada izabere zubara sistem će mu izbaci slobodne termine (datum i vrijeme). Potvrdu da je zakazao termin će dobiti na e-mail.

Drugi dijagram prikazuje elemente podataka u sistemu i odnos između njih što se tiče desktop aplikacije.



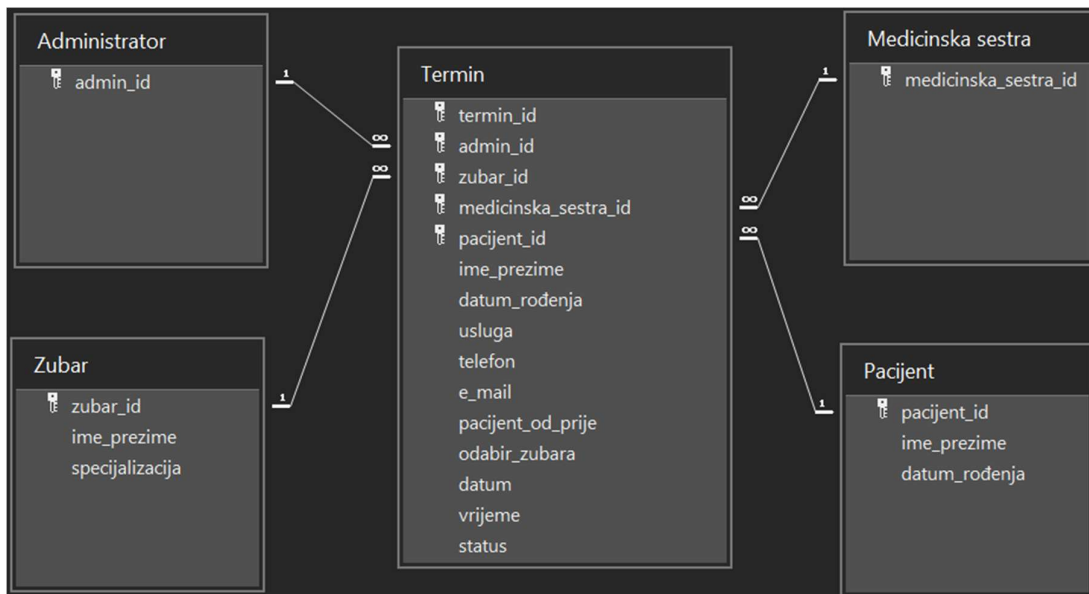
Slika 11. Dijagram klasa Elektronski karton (za desktop aplikaciju)

Administrator može da doda karton na sistem, ažurira podatke u kartonu, brisati kartone sa sistema, te imati pristup svim podacima na sistemu. Zubar će moći da doda karton na sistem, ažurira i briše karton, te pristup kartonu. Medicinska sestra će moći da ažurira karton i pristupa istom.

6. SISTEMSKI DIZAJN

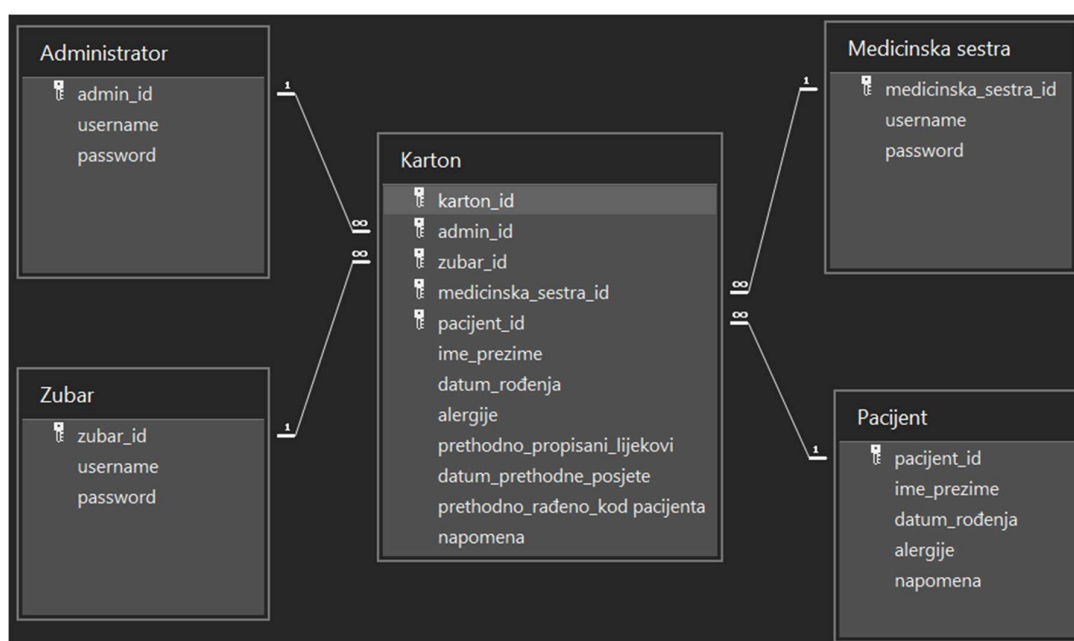
6.1. PROJEKAT BAZE PODATAKA

Baze podataka su rađene u Access-u. Na slici ispod je ER dijagram za web aplikaciju za termine. Na ER dijagramu možemo vidjeti entitet Termin sa svojim atributima: termin_id, admin_id, zubar_id, medicinska_sestra_id, pacijent_id, ime_prezime, datum_rođenja, usluga, telefon, e-mail, pacijent_od_prije, datum, vrijeme i status. Termin_id je primarni ključ u ovoj tabeli, dok ostali id su strani ključevi koji predstavljaju poveznicu sa ostalim tabelama u bazi. Ostali atributi su svi vezani šta treba pacijent popuniti prilikom zakazivanja termina. Status predstavlja da li je termin zauzet ili nije. Entiteti Medicinska sestra i Administrator imaju samo svoje primarne ključeve. Kod entiteta Zubar imamo njegov primarni ključ zubar_id, te attribute ime_prezime i specijalizacija, ti atributi su potrebni za termine prilikom izbora zubara od strane pacijenta. Te entitet Pacijent ima svoj primarni ključ pacijent_id i attribute ime_prezime i datum_rođenja. Sama implementacija baze podataka je spomenuta u arhitekturi sistema. Sloj baze podataka bi bio implementiran nad SQL serverom i samim tim imamo relacionu bazu podataka.



Slika 12. ER dijagram za web aplikaciju za termine

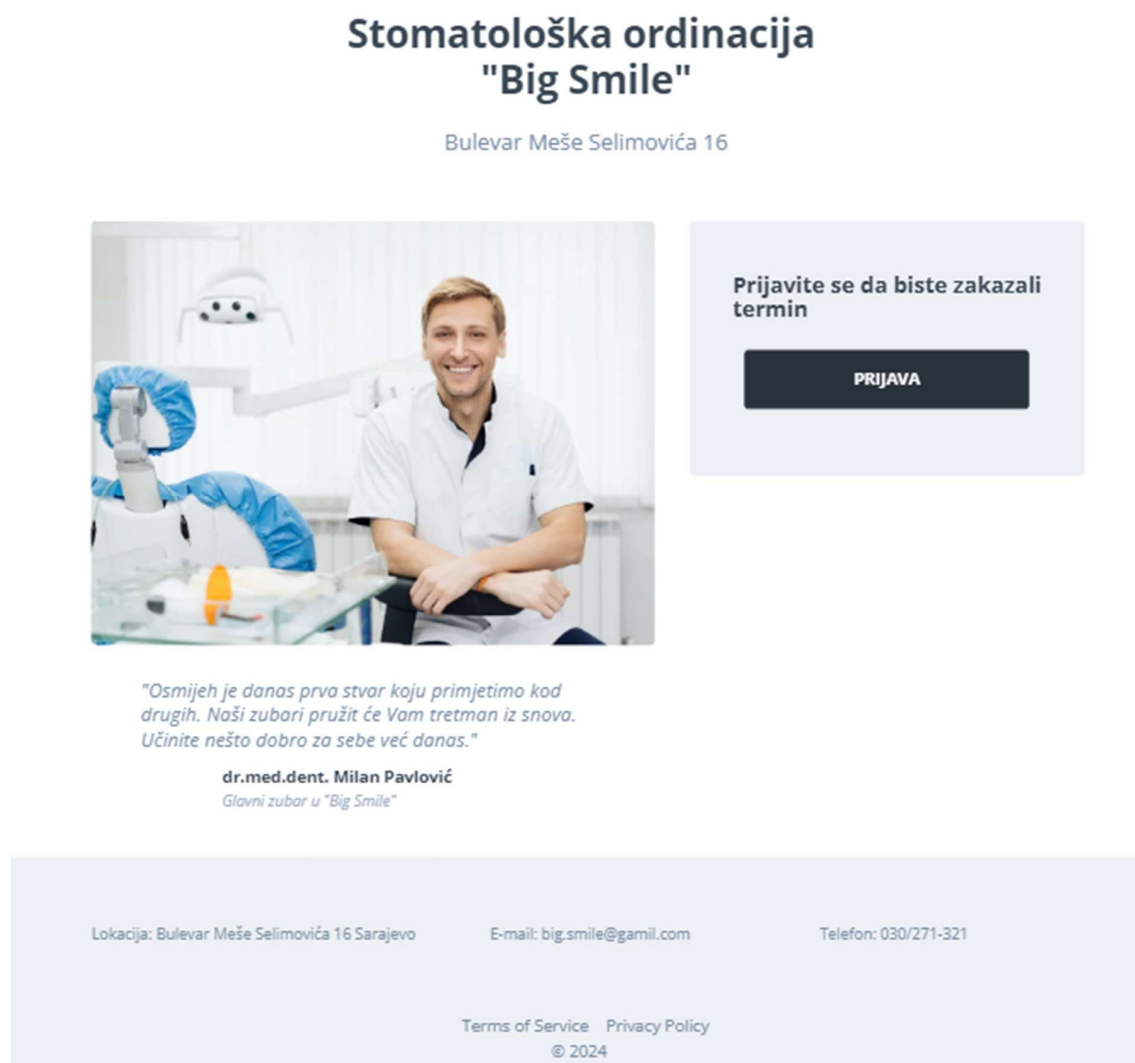
Na slici ispod je ER dijagram za desktop aplikaciju za kartone. Na ER dijagramu možemo vidjeti entitet Karton sa svojim atributima: `karton_id`, `admin_id`, `zubar_id`, `medicinska_sestra_id`, `pacijent_id`, `ime_prezime`, `datum_rođenja`, `alerigije`, `prethodno_propisani_lijekovi`, `datum_prethodne_posjete`, `prethodno_rađeno_kod_pacijenta` i `napomena`. `Karton_id` je primarni ključ u ovoj tabeli, dok ostali id su strani ključevi koji predstavljaju poveznicu sa ostalim tabelama u bazi. Ostali atributi su svi vezani šta treba zubar popuniti prilikom pravljenja novog kartona ili šta će izbaciti sistem kad uđe u neki već napravljen karton. Entiteti Medicinska sestra, Administrator i Zubar imaju svoje primarne ključeve i još imaju svoj username i password za pristup kartonima. Te imamo entitet Pacijent koji ima svoj primarni ključ `pacijent_id` i attribute `ime_prezime`, `datum_rođenja`, `alerigije` i `napomene`. Sloj baze podataka bi bio implementiran nad SQL serverom i samim tim imamo relacionu bazu podataka, kao što je gore prethodno navedeno.



Slika 13. ER dijagram za desktop aplikaciju za kartone

6.2. DIZAJN FORMI I IZVJEŠTAJA PUTEM MOCKUP-A

Svi dole u nastavku rada urađeni dizajnovi su napravljeni na besplatnoj stranici mockups.com. Slika ispod prikazuje izgled početne web stranice za pacijente. Pacijent na stranici ima prikazano ime stomatološke ordinacije, adresu, e-mail, te broj telefona ordinacije. Klikom na dugme prijava vodi ga na stranicu gdje bira da li će da zakaže termin ili da otkáže termin, potom kad izabere opciju zakaži termin popunjava potrebne podatke, ukoliko klikne na otkáže termin briše željeni termin.



Slika 14. Izgled web aplikacije za pacijente

Za zakazivanje termina pacijent mora popuniti sljedeće podatke: ime i prezime, datum rođenja, uslugu koju želi, broj telefona, e-mail, da zabilježi da li je od prije pacijent ili nije, izabere zubara, željeni datum i vrijeme. Kada pacijent unese sve tražene podatke, podaci se provjeravaju. Potom dobija potvrdu na e-mail da je uspješno zakazao termin ili da prepravi podatke. Ukoliko ima pacijent problema prilikom prijave može se obratiti adminu klikom na „Javite se adminu.“

Popunite

Vodite računa da podaci koje unesete u nastavku budu ispravni.

Ime i prezime

Datum rođenja

Usluga

Telefon

E-mail

☒ Pacijent od prije

Izaberite zubara

Izaberite željeni datum

Vrijeme

Zakažite termin

Imate problema prilikom zakazivanja termina?

[Javite se adminu.](#)

Slika 15. Izgled stranice za zakazivanje termina

Što se tiče izgleda desktop aplikacije ona je prikazana ispod na slici. Potom korisnik bira svoju ulogu admin, zubar ili medicinska sestra. Klikom na dugme otvara se nova stranica za popunjavanje login podataka.



Slika 16. Izgled desktop aplikacije

U nastavku je prikazan izgled login stranice gdje korisnik mora upisati svoj username i password, omogućeno je čak i da ga sistem zapamti. Ako se pojavi neka poteškoća prilikom prijave može kontaktirati admina klikom na „Javite se adminu.“

Slika 17. Prikaz mogućeg login-a za desktop aplikaciju

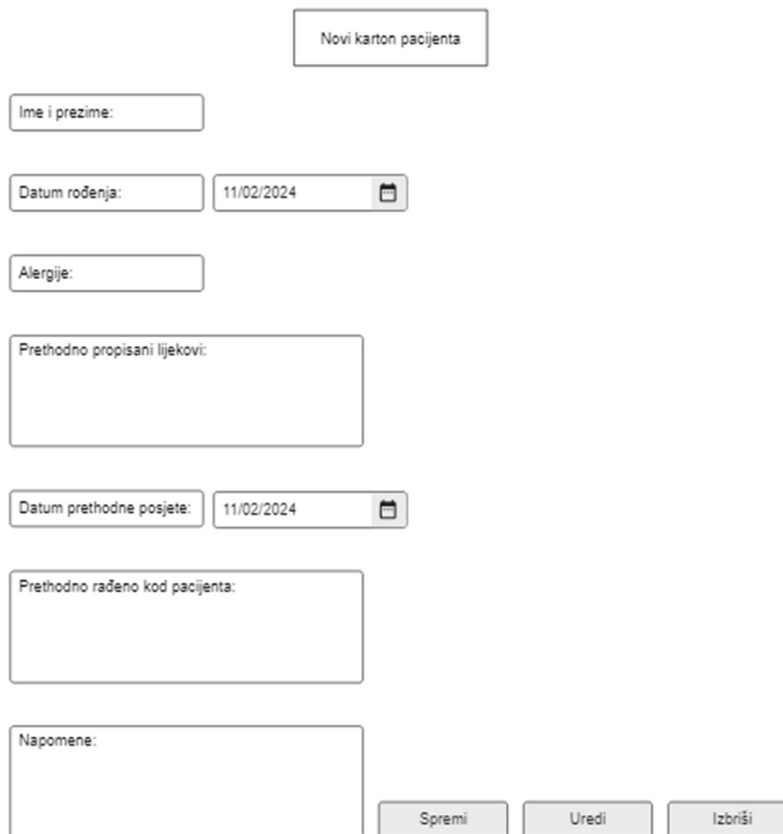
Kada se uspješno prijavi korisnik na sistem ima opciju da bira da li će pretražiti karton ili će napraviti novi karton. Medicinskoj sestri će izbaciti samo opciju pretraži karton. Ako korisnik izabere pretraži karton, trebat će upisati ime i prezime pacijenta, te datum rođenja. Ukoliko klikne na opciju napravi karton. Stranica će ga odvesti da pravi novi karton za pacijenta.

Pretražite karton ili napravite novi



Slika 18. Prikaz za pretragu ili pravljenje novog kartona

Izgled stranice za pravljenje kartona pacijentu je prikazan u nastavku. Zubar mora popuniti podatke tražene od njega vezano za pacijenta, a informacije će dobiti od samog pacijenta ispitivanjem. U samom dnu ima opciju da spremi karton, uredi ili obriše. Kod medicinske sestre će biti samo opcija da uredi i spremi podatke.



Slika 19. Prikaz izgleda elektronskog kartona

6.3. DIZAJN INTERFEJSA I DIJALOGA

Na slici ispod je prikazan dijalog između sistema i pacijenta. Gdje crvenim okvirom pokazano šta nije dobro napisano ili nije uopšte popunjeno.

Popunite

Niste ispravno unijeli podatke. Molimo Vas provjerite još jednom.

Ime i prezime

Muhamed Muhić

E-mail

✉ muhamed@gmail.com

Datum rođenja

19/11/1997

☒ Pacijent od prije

Izaberite zubara

Zubar

Usluga

Kontrola

Izaberite željeni datum

11/02/2024

Telefon

+387-61/111-711

Vrijeme

hh/mm/ss

Zakažite termin

Imate problema prilikom zakazivanja termina?

[Javite se adminu.](#)

Slika 20. Dijalog u slučaju pogrešnog unosa

Kada uspješno pacijent popuni podatke sistem mu izbacuje dijalog da provjeri svoj e-mail o potvrdi da je uspješno zakazao termin.

Provjerite svoj e-mail

Poslali smo Vam potvrdu da ste uspješno zakazali termin.

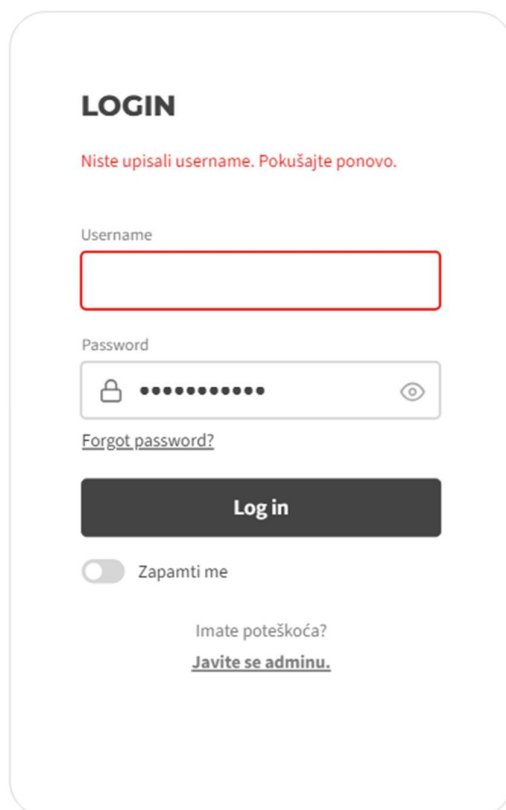
Niste dobili e-mail? Provjerite spam folder ili se javite adminu.

[Admin](#)

Izađite sa stranice

Slika 21. Prikaz ukoliko se uspješno termin zakaže

Ukoliko korisnik desktop aplikacije pogrešno upiše podatke sistem će ga obavijestiti i pokazati crvenom bojom šta je pogrešno upisano.



The image shows a login dialog box with the title "LOGIN". Below the title, there is a red error message: "Niste upisali username. Pokušajte ponovo." (You did not enter a username. Please try again). The "Username" field is highlighted with a red border, indicating the source of the error. The "Password" field is shown with a lock icon and a toggle for visibility. Below the password field, there is a link for "Forgot password?". A "Log in" button is positioned below the password field. At the bottom, there is a "Zapamti me" (Remember me) toggle switch and a link for "Javite se adminu." (Report to admin.).

Slika 22. Dijalog za pogrešno unešene login podatke

7. ARHITEKTURA SISTEMA

S obzirom na to da je projekat za zamišljen da bude web aplikacija za pacijente i desktop aplikacija za zaposlene, idealna arhitektura sistema bi bila slojevita arhitektura.

Slojevita arhitektura omogućava:

- Paralelan rad na različitim slojevima arhitekture bez konflikata
- Omogućava fleksibilnost, održivost i skalabilnost aplikacije
- Odvajanje prezentacijskog sloja od poslovne logike
- Omogućava raspoređivanje, ažuriranje, dodavanje novih pogodnosti na različitim slojevima i u različito vrijeme

Slojevita arhitektura sastoji se od slojeva:

- Prezentacijski sloj
- Klijent kontroler
- Server kontroler
- Baza podataka

Prezentacijski sloj je direktno dostupan korisniku sistema, u našem slučaju pacijentu, zubaru i medicinskoj sestri. Svako ko pristupi web stranici stomatološke ordinacije može da zakaže termin. Neregistrovanim korisnicima desktop aplikacije je moguće samo da dođu do login stranice dalje im nije dozvoljeno. Admin daje pristupne podatke novozaposlenom, dok starom uposleniku ukida pristup i briše ga sa sistema.

Implementacija bi bila izvršena pomoću JavaScript-a sa framework-a React uz pomoć HTML, CSS. Desktop aplikacija bi bila implementirana uz pomoć C# sa Windows Forms.

Klijent kontroler je sloj koji će sadržavati validaciju podataka koje korisnik unese u sistem na desktop aplikaciji.

Sloj server kontrolera sadrži informacije o validaciji i autentifikaciji. Ukoliko podaci nisu validni, klijentu se šalje informacija da je došlo do greške prilikom upisa podataka, te da ih mora popraviti. Server kontroler prihvata zahtjeve klijenta, obrađuje ih i komunicira sa bazom podataka. Implementacija bi bila izvršena u programskom jeziku Java sa SpringBoot framework-om.

Sloj baze podataka bi bio implementiran u SQL serveru i samim tim bi imao i relacionu bazu podataka.

8. BACKEND

Backend bi se implementirao uz pomoć dva programska jezika:

- Za desktop aplikaciju koristio bi se C# sa Windows Forms
- Za web aplikaciju bi se koristio Java sa SpringBoot-om

8.1. C# SA WINDOWS FORMS

C# je programski jezik koji je razvio Microsoft, a prvi put je predstavljen 2000. godine kao dio .NET framework-a.

Mogućnosti C#:

1. Objektno-orijentisan: C# podržava principe objektno-orijentisanog programiranja (OOP), što olakšava organizaciju i upravljanje kodom.
2. Tipizacija: C# je statički tipiziran jezik, što znači da se tipovi podataka definišu prilikom kompilacije, što pomaže u otkrivanju grešaka.
3. Jednostavnost sintakse: C# ima čistu i jasnu sintaksu, što olakšava učenje i korištenje.
4. Podrška za LINQ: Omogućava lahko upitivanje i manipulisanje podacima u kolekcijama i bazama podataka.
5. Multiplatformska podrška: Uz .NET Core (sada .NET 5/6 i novije), C# može raditi na Windowsu, Linuxu i macOS-u.

Upotreba C#:

- Razvoj desktop aplikacija: Koristeći Windows Forms ili WPF (Windows Presentation Foundation).
- Web aplikacije: Uz ASP.NET za izradu dinamičnih web stranica i RESTful API-a.
- Mobilne aplikacije: Kroz Xamarin, koji omogućava razvoj aplikacija za iOS i Android.
- Igre: Uz Unity, C# se koristi za razvoj video igara.
- Cloud i IoT aplikacije: C# se također koristi u razvoju aplikacija za cloud rješenja i Internet of Things (IoT) uređaje.

Windows Forms (WinForms) je grafički korisnički interfejs (GUI) framework koji se koristi za izradu desktop aplikacija na Windows operativnom sistemu. Razvijen je od strane Microsofta i prvi put je predstavljen 2002. godine kao dio .NET Framework-a.

WinForms se koristi za razvoj desktop aplikacija koje se izvode na Windows platformi. Ovaj framework omogućava programerima da kreiraju aplikacije koje su korisnički interaktivne i jednostavne za korištenje. Uz njega se i frontend realizira.

Karakteristike:

- Brza izrada: WinForms omogućava brzo prototipiranje i razvoj aplikacija zahvaljujući vizuelnom dizajneru.
- Bogata biblioteka kontrola: Pruža različite kontrole (dugmad, tekstualna polja, etikete, meniji, itd.) koje se lako mogu dodavati na forme.
- Event-driven arhitektura: Programiranje se bazira na događajima, što

omogućava reagovanje na korisničke akcije (npr. klikovi, unos teksta).

- Integracija sa .NET Framework-om: Koristi sve prednosti .NET ekosistema, uključujući biblioteke za pristup podacima, mrežne funkcionalnosti i druge alate.

Upotreba:

- WinForms se koristi za razvoj raznih vrsta desktop aplikacija, uključujući:
 - Administrativne alate.
 - Prilagođene aplikacije za preduzeća.
 - Aplikacije za obradu podataka.
 - Igre i multimedia aplikacije.

Razvoj i održavanje:

- Iako se pojavljuju noviji framework-ovi poput WPF-a i UWP-a, WinForms ostaje popularan zbog svoje jednostavnosti i stabilnosti. Također, uz .NET Core (sada .NET 5/6 i novije), WinForms aplikacije mogu biti razvijene za multiplatformsko korištenje.

Windows Forms je snažan alat za programere koji žele brzo i efikasno razvijati desktop aplikacije na Windows operativnom sistemu. Njegova jednostavnost i bogata funkcionalnost čine ga popularnim izborom za mnoge vrste aplikacija.

8.2. JAVA SA SPRINGBOOT-OM

Java je programski jezik koji je razvijen od strane kompanije Sun Microsystems, a prvi put je objavljen 1995. godine. Stvoren je kao objektnorijentisani jezik koji se može koristiti na različitim platformama, što ga čini veoma fleksibilnim i popularnim među programerima. Java je prvobitno razvijena za interaktivne televizore, ali je brzo evoluirala u univerzalni programski jezik. Danas je u vlasništvu Oracle Corporation, koja je nastavila sa razvojem i održavanjem jezika.

Mogućnosti:

- Objektno-orijentisano programiranje (OOP): Java podržava principe OOP, kao što su nasljeđivanje, polimorfizam i enkapsulacija, što olakšava organizaciju i ponovno korištenje koda.
- Platformna neovisnost: Java aplikacije se prevode u bytecode koji se izvršava na Java Virtual Machine (JVM), omogućavajući da se aplikacije pokreću na različitim platformama bez potrebe za ponovnim prevođenjem.
- Sigurnost: Java ima ugrađene sigurnosne funkcionalnosti koje omogućavaju zaštitu od raznih vrsta napada, što je čini pogodnom za razvoj aplikacija koje zahtijevaju visoki nivo bezbjednosti.
- Bogata biblioteka: Java dolazi sa velikim brojem ugrađenih biblioteka (Java Standard Library) koje olakšavaju razvoj aplikacija kroz unaprijed definisane funkcionalnosti.
- Multitasking: Podrška za više niti (threads) omogućava lako kreiranje aplikacija koje mogu obavljati više zadataka istovremeno.
- Podrška za mrežne aplikacije: Java olakšava razvoj mrežnih aplikacija kroz

različite API-jeve.

Upotreba:

- Web aplikacije: Java se široko koristi za razvoj dinamičkih web aplikacija pomoću frameworka kao što su Spring i JavaServer Faces (JSF).
- Mobilne aplikacije: Java je osnovni jezik za razvoj Android aplikacija.
- Desktop aplikacije: Mogu se razvijati desktop aplikacije koristeći JavaFX ili Swing.
- Server-side aplikacije: Java je popularna za razvoj backend rješenja, uključujući mikroservise.
- Igre: Koristi se i u razvoju igara, posebno onih koje se izvode na Java platformi.
- Veliki podaci: Java se često koristi u okruženjima za obradu velikih podataka (big data) kao što su Hadoop i Apache Spark.

Spring Boot je framework za razvoj Java aplikacija koji se bazira na Spring frameworku. Predstavljen je 2014. godine i ima za cilj pojednostavljivanje i ubrzavanje procesa razvoja aplikacija u Java ekosistemu. Spring Boot se koristi za brzo kreiranje samostalnih, produkcijskih aplikacija u Javi. Uglavnom se koristi za razvoj web aplikacija i mikroservisa.

Mogućnosti:

- Automatska konfiguracija: Spring Boot automatski konfiguriše aplikaciju na osnovu zavisnosti koje su dodate, čime se smanjuje potreba za manuelnom konfiguracijom.
- Samostalne aplikacije: Omogućava kreiranje aplikacija koje se mogu lako pokretati kao samostalni Java programi (npr. java -jar aplikacija.jar).
- Mikroservisi: Idealan je za razvoj mikroservisne arhitekture, pružajući alate za jednostavno kreiranje i upravljanje mikroservisima.
- Integracija sa drugim tehnologijama: Pruža lako povezivanje sa bazama podataka (preko Spring Data), sigurnosnim rešenjima (preko Spring Security) i mnogim drugim tehnologijama.
- RESTful API: Olakšava kreiranje RESTful web servisa pomoću jednostavnog programskog modela.
- Testiranje: Uključuje podršku za testiranje aplikacija kroz JUnit i druge testne biblioteke, čime se olakšava pisanje i izvođenje testova.
- Prikazivanje i upravljanje metrikama: Ugrađene funkcionalnosti za praćenje performansi aplikacije i upravljanje metrikama pomoću Spring Actuator.

Upotreba:

- Web aplikacije: Kreiranje dinamičkih web aplikacija uz korištenje Spring MVC-a.
- Mikroservisi: Razvoj i implementacija mikroservisa koji mogu komunicirati jedni s drugima.
- Backend za mobilne aplikacije: Kao backend servis za različite mobilne aplikacije.
- RESTful API: Kreiranje i izlaganje API-ja koji se može koristiti od strane klijentskih aplikacija.

9. FRONTEND

Frontend bi se implementirao uz pomoć JavaScript uz framework React za web aplikaciju.

JavaScript je dinamički, prototipno orijentisan programski jezik koji je prvobitno razvijen od strane Brendana Eichu u Netscape-u. Prvi put je predstavljen 1995. godine kao dio Netscape Navigatora, popularnog web pretraživača.

JavaScript se primarno koristi za dodavanje interaktivnosti i dinamičkog sadržaja na web stranicama. Također se koristi za razvoj server-side aplikacija i aplikacija za mobilne uređaje.

Mogućnosti:

- Interaktivnost: JavaScript omogućava dinamične promjene na web stranicama, kao što su animacije, validacija obrazaca i manipulacija DOM-om (Document Object Model).
- Asinhrono programiranje: Korištenjem tehnika poput Promises i async/await, JavaScript omogućava upravljanje asinhronim operacijama, što poboljšava korisničko iskustvo.
- Razvoj web aplikacija: Uz moderne JavaScript frameworke kao što su React, Angular i Vue.js, programeri mogu lako kreirati kompleksne web aplikacije.
- Server-side programiranje: Korištenjem Node.js, JavaScript se može koristiti za razvoj server-side aplikacija, omogućavajući programerima da koriste jedan jezik za cijelokupni razvoj aplikacije.
- Široka podrška: JavaScript je podržan od strane svih modernih web pretraživača, što ga čini univerzalnim jezikom za web.
- Biblioteke i framework: Postoji mnoštvo biblioteka (poput jQuery) i framework-a (poput Express.js) koji olakšavaju razvoj aplikacija i povećavaju produktivnost.

Upotreba:

- Frontend razvoj: Uključuje dodavanje interaktivnosti i dinamičkog sadržaja na web stranice.
- Backend razvoj: Razvoj server-side aplikacija koristeći Node.js.
- Mobilne aplikacije: Razvoj mobilnih aplikacija uz pomoć framework-a kao što su React Native ili Ionic.
- Igre: JavaScript se također koristi za razvoj web igara, zahvaljujući tehnologijama poput HTML5.

React je popularna JavaScript biblioteka za izradu korisničkih interfejsa, koja je razvijena od strane Facebook-a. Prvi put je predstavljena 2013. godine.

React se koristi za izradu korisničkih interfejsa, posebno za dinamične i interaktivne web aplikacije. Fokusira se na efikasno upravljanje stanjem aplikacije i re-renderovanje samo onih delova interfejsa koji se mijenjaju.

Mogućnosti:

- Komponentni pristup: React omogućava razvoj aplikacija kroz komponente, što olakšava ponovno korištenje koda i organizaciju projekta.
- Virtuelni DOM: React koristi virtuelni DOM za optimizaciju performansi, što

znači da se promjene u interfejsu primjenjuju samo na dijelovima DOM-a koji su se promijenili, a ne na cijelom dokumentu.

- Jednostavna integracija: Može se lahko integrisati sa drugim bibliotekama ili framework-ima, kao što su Redux za upravljanje stanjem ili React Router za upravljanje rutama.
- Bogat sistem: Postoji mnogo biblioteka i alata koji se koriste uz React, uključujući Next.js za server-side rendering i Gatsby za statički sajt.
- Podrška za JSX: React koristi JSX, sintaksu koja omogućava kombinovanje HTML-a i JavaScript-a, što olakšava pisanje komponenti.

Upotreba:

- Web aplikacije: React se koristi za izradu jedinstvenih web aplikacija (SPA) koje nude bogato korisničko iskustvo.
- Mobilne aplikacije: Korišćenjem React Native, programeri mogu razvijati mobilne aplikacije za iOS i Android koristeći isti kod kao za web aplikacije.
- Integracija s drugim tehnologijama: Može se koristiti u kombinaciji sa backend tehnologijama kao što su Node.js, Express, i mnogim drugim.

10. TESTIRANJE INFORMACIONOG SISTEMA

Poslije dizajniranja i implementiranja informacionog sistema stomatološke ordinacije slijedi faza testiranja sistema. U ovoj fazi tester sistema detaljno svaki segment sistema ispituje. Ukoliko postoji neka greška šalje nazad programerima da poprave grešku. Zadatak testera je da pronade sve greške koje su promakle programerima. Nakon što sistem prođe sve faze testiranja pušta se u upotrebu. Nakon puštanja moguće su neke greške u sistemu da se dese za to je tu administrator koji će popravljati greške. Pošto je klijentu pruženo održavanje sistema na tri godine za to vrijeme sve greške i moguće nadogradnje sistema se mogu odraditi. Naš sistem uspješno omogućava pacijentima da naruče termin preko web stranice, te zubarima i medicinskim sestrama lakše pronalaženje kartona putem desktop aplikacije.

11. ZAKLJUČAK

Informacioni sistem za stomatološku ordinaciju ima bogat korisnički interfejs tako da korisnici mogu lahko pristupiti istom. Desktop aplikacija pruža uvid u sve kartone svih pacijenata, a web aplikacija pruža uvid u zakazane termine. Sa desktop aplikacijom smanjeno je vrijeme traženja kartona pacijenta, ali i potrošnja papira sa čime brinemo i o našoj prirodi. Softver će se razvijati dalje pod odgovarajućim nadzorom implementirajući sve module koji su navedeni gore u tekstu.