

Programsko inženjerstvo

Ak. god. 2022./2023.

Sustav za naručivanje

Dokumentacija, Rev. 1

Grupa: *Null*

Voditelj: *Darijan Gudelj*

Datum predaje: 18.11.2022.

Nastavnik: *Goran Rajić*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
3	Specifikacija programske potpore	10
3.1	Funkcionalni zahtjevi	10
3.1.1	Obrasci uporabe	12
3.1.2	Sekvencijski dijagrami	24
3.2	Ostali zahtjevi	28
4	Arhitektura i dizajn sustava	29
4.1	Baza podataka	31
4.1.1	Opis tablica	31
4.1.2	Dijagram baze podataka	34
4.1.3	Dijagram baze podataka	34
4.2	Dijagram razreda	35
4.3	Dijagram stanja	38
4.4	Dijagram aktivnosti	39
4.5	Dijagram komponenti	40
5	Implementacija i korisničko sučelje	41
5.1	Korištene tehnologije i alati	41
5.2	Ispitivanje programskog rješenja	42
5.2.1	Ispitivanje komponenti	42
5.2.2	Ispitivanje sustava	45
5.3	Dijagram razmještaja	51
5.4	Upute za puštanje u pogon	52
5.4.1	Izrada koričkog računa na Microsoftovom Azure servisu . . .	52
5.4.2	Zauzimanje nove virtualne mašine	52
5.4.3	Logiranje u novu virtualnu mašinu	52

5.4.4	Instaliranje potrebnih knjižnica i programa te downloadanje programa	53
5.4.5	Punjenje Postgres baze podataka	53
5.4.6	Pokretanje aplikacije	54
6	Zaključak	55
	Popis literature	57
	Indeks slika i dijagrama	60
	Dodatak: Prikaz aktivnosti grupe	61

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Dodani funkcionalni zahtjevi, opisi obrazaca uporabe i <i>Use Case</i> dijagrami	Branimir Tomelj, Darijan Gudelj, Luka Slugečić	14.11.2022.
0.2	Opis projektnog zadatka Dodani sekvencijski dijagrami i ostali zahtjevi Upisani sastanci	Bruno Rački, Luka Slugečić	15.11.2022.
0.3	Započeta arhitektura projekta Dodan opis baze podataka i dijagrami	Vilim Ivanković, Luka Slugečić, Bruno Rački	15.11.2022.
0.4	Dodan dijagram razreda i njegov opis	Luka Slugečić i Vilim Ivanković	16.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.5	Dodana arhitektura projekta i napravljena revizija rada	Bruno Rački i Luka Slugečić	16.11.2022.
0.6	Ispravljena baza podataka i opis projektnog zadatka	Vilim Ivan- ković, Bruno Rački, Darijan Gudelj	17.11.2022.
1.0	Dodan još jedan dijagram razreda i opis Ispravljen dnevnik Verzija samo s bitnim dijelovima za 1. ciklus	Luka Slugečić, Bruno Rački	18.11.2022.
1.1	Dodan dijagram razmjestaja i opis	Branimir Tomelj	10.1.2022.

2. Opis projektnog zadatka

Cilj projektnog zadatka

Kako je zdravlje najvažnija stvar u ljudskome životu za normalnu funkcionalnost društva, došlo je vrijeme da stvari krenu na bolje i da se naše zdravstvo modernizira. Zdravstvo samo po sebi je ogroman sustav sa jako puno sudionika, bilo radnika ili pacijenata, stoga se na prvu čini jako teško riješiti problem ubrzanja naručivanja pregleda i poboljšanja korisničkih usluga u zdravstvu. No, nakon sažimanja i kompresiranja problema u manje stavke, možemo sa sigurnošću reći da je problem rješiv i ostvariv. Cilj ovog projektnog zadatka je realizirati sustav planiranja zdravstvenih usluga i komunikacije s korisnicima koji bi omogućio učinkovite primarne, polikliničke i bolničke zdravstvene usluge u hrvatskom zdravstvu, lako dostupne svim građanima. Kako su korisnici ove aplikacije svi punoljetni građani Republike Hrvatske, sustav mora omogućavati jednostavnu uporabu sustava naručivanja koja mora biti prilagođena svim uzrastima.

Potencijalna korist projekta

Korist ovog projekta može se iskazati na više načina. Korisnost samog rada biti će na pomoć najprije djelatnicima hrvatskog zdravstva koji će biti odriješeni javljanja na telefon i odgovaranja na email poruke svojih pacijenata, već će biti u mogućnosti fokusirati se na svoj rad, a ne gubiti vrijeme na formalnosti. Naravno, ovaj sustav će biti jednako tako koristan i pacijentima koji neće morati pisati duge mailove i čekati nekoliko dana kako bi im se djelatnik javio na poziv. Dakle s obje strane imamo veliko poboljšanje. Sustav će omogućiti konstantnu mogućnost prijave pregleda kod željenog zdravstvenika, neovisno koji je dan u tjednu. Pacijenti će tako imati sigurniji život i puno će brže obavljati samo prijavu pregleda.


Primjeri sličnih rješenja u stvarnosti

Primjer korištenja ovakvih funkcionalnosti u stvarnosti je stranica Reservatic.com koja pruža online sustav rezervacija za uslužne sektore.



Slika 2.1: Početna stranica Reservatic.com

Početna stranica posjeduje kratke informacije o sustavu, a header gumbe za registraciju i prijavu, koji će biti slični u implementaciji Null grupe 2022./2023.


Reservatic
Live your time

Registrirajte se besplatno

Ovdje možete registrirati osobni račun u sustavu Reservatic. Račun nije vezan uz rezervaciju cijepljenja i te rezervacije neće biti spremljene na vaš račun.

Ako se želite registrirati za cijepljenje protiv COVID-19, najprije se morate registrirati na <https://registrace.mzcr.cz>, a zatim <https://reservatic.com/ockovani>.

tvoj e-mail

Ponovno pošaljite e-poštu


☐ Slažem se [opći uvjeti i odredbe](#)
☐ Ovime se slažem [za prikupljanje, zaštitu i obradu mojih osobnih podataka](#)

prijavite se besplatno

Imate li račun?

Jedinstveni sustav rezervacija

Aplikacija Reservatic nudi jednostavno i elegantno rješenje koje se sastoji od korištenja mogućnosti online rezervacije ili rezervacije putem mobilne aplikacije.



Slika 2.2: Registracija u Reservatic.com

Kupcima nudimo puno više za manje nego što bi očekivali...

Cjenik za tvrtke	FREE 0 HRK / Mjesec	STANDARD 85 HRK / Mjesec	PROFI 289 HRK / Mjesec
	Pokušati besplatno	Pokušati besplatno	Pokušati besplatno
Kalendari	1	2 do 5	6 do 20
Broj administratora	neograničen	neograničen	neograničen
Broj rezervacija	neograničen	neograničen	neograničen
nadopune	✓	✓	✓
SSL sigurnost	✓	✓	✓
Neograničen broj rezervacija	✓	✓	✓
Napredna statistika	✗	✓	✓

Slika 2.3: Razlika između Reservatic.com i Null grupe

Registracija na stranici Reservatic.com je jednostavna i laka za uporabu. Dizajn Reservatic.com stranice vrlo je intuitivan i jednostavan za razumijevanje. Zanimljiva činjenica je da je Reservatic.com dodao naplaćivanje korištenja sustava za sve tvrtke po čemu se stranice Null grupe i Reservatic.com razlikuju.

Ciljana skupina korisnika

Korisnici aplikacije će biti pacijenti, liječnici, medicinske sestre i administratori. Većinu korisnika činit će pacijenti i oni su najvažniji dio populacije. Aplikacija mora biti prilagođena svim uzrastima što će se postići izradom što jednostavnijih sučelja i dizajna. Ciljane dobne skupine su od 18 godina starosti do najstarijih građana. Aplikacija mora biti jednostavna za korištenje kako bi svaki građanin mogao pristupiti istoj i koristiti ju.

Opseg projekta

Opseg projekta opisat ćemo kroz korisničke zahtjeve i ciljeve, vremenske robove i troškove. Korisnici aplikacije će biti pacijenti, liječnici, medicinske sestre i administratori. Osnovna funkcionalnost dostupna pacijentima je zakazivanje pregleda kod liječnika opće medicine te slanje podsjetnika za termine putem SMS poruka ili e-pošte. Prilikom registracije u aplikaciju pacijent unosi osnovne podatke te lozinku i odabir liječnika. Za unesene osnovne podatke osiguravamo provjeru konzistentnosti s podacima iz baze podataka. Pri uspješnoj registraciji pacijent prima potvrdu na svoju registriranu adresu e-pošte. Nakon toga korisnik postaje pacijent i dobiva neke mogućnosti. Pacijenti se prijavljuju za termin kod liječnika ili sestre. Za prijavu kod liječnika pacijent odabire liječnika, termin i tip pregleda. Za prijavu kod sestre, pacijentu se dodjeljuje prvi slobodan termin od unaprijed dogovorenih termina. Pacijenti potvrđuju odabrani termin, a liječnici dobivaju obavijest o rezervaciji termina. Pacijent ima mogućnost otkazati termin najkasnije 24 sata prije početka. Sustav automatski generira potvrdu termina i šalje obavijest putem preferiranog kanala komunikacije SMS ili e-pošta. Sustav automatski šalje podsjetnike pacijentima o zakazanom terminu s eventualnom personaliziranim porukom.

Vremensko ograničenje za prvu verziju projekta je 7 tjedana. Za drugu verziju,

odnosno ukupan projekt predviđeno je 14 tjedana rada. Na projektu radi 6 osoba, za svaku je predviđeno 10 sati rada tjedno. Dakle ukupno je predviđeno vrijeme od 840 sati projektnog rada. 840 sati trošit će se većinski na rad, ali isto tako značajan dio odlazi na sastanke koji se održavaju minimalno svaki tjedan po 3 sata.

Troškovi projekta svedeni su na minimum. Satnica radnicima biti će 0 kuna jer se projekt radi u svrhe provjere znanja studenata. Null grupa spremna je odvojiti iznos od 100 EUR (753 HRK) za ulaganje u hosting poslužitelja i web aplikacije.

Moguće nadogradnje

Projektanti Null grupe 2022./2023. dogovorili su da nemaju plan mijenjanja korisničkih zahtjeva te će se sve odraditi kako je i sam korisnik htio. Ukoliko bude potrebe, Null grupa je spremna promijeniti dijelove plana programa i izvršiti implementaciju dodatnih mogućnosti u rad sustava i inovaciju sadržaja.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnik
 - Pacijent
 - Liječnik
 - Medicinski tehničar/sestra
2. Administrator
3. Naručitelj
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) pregledati naslovnu stranicu za neprijavljene korisnike
 - (b) se registrirati u sustav kao pacijent, stvoriti novi korisnički račun za koji su mu potrebni e-mail adresa, lozinka, ime, prezime, spol, broj mobitela i odabrani liječnik
2. Pacijent (inicijator) može:
 - (a) pregledati naslovnu stranicu za pacijente, svoj kalendar sa zakazanim pregledima i prozor za poruke o pomaknutim terminima
 - (b) potvrditi ili odbiti pomaknute termine
 - (c) odabrati pregled kod liječnika ili specifičnu vrstu usluge kod medicinskog tehničara/sestre, te zakazati termin
 - (d) zakazani termin otkazati
 - (e) dobiti podsjetnik na pregled, ili obavijest o pomaknutom pregledu na svoj e-mail ili sms
3. Liječnik (inicijator) može:

- (a) pregledati naslovnu stranicu za liječnike, svoj kalendar s rezerviranim terminima
- (b) za rezervirane termine vidjeti tip pregleda i podatke o pacijentu
- (c) definirati raspoloživost termina
- (d) definirati vlastita pravila o rezervaciji termina i proizvoljno iz mijenjati
- (e) pomaknuti rezervirani termin
- (f) unositi potvrdu dolaska pacijenta na rezervirani termin

4. Medicinski tehničar/sestra (inicijator) može:

- (a) pregledati naslovnu stranicu za medicinskog tehničara, svoj kalendar sa rezerviranim terminima
- (b) za rezervirane termine vidjeti vrstu specifične usluge
- (c) definirati slobodne termine za specifične vrste usluga
- (d) pomaknuti rezervirani termin
- (e) unositi potvrdu dolaska pacijenta na rezervirani termin

5. Administrator (inicijator) može:

- (a) kreirati specijalizirane tipove korisnika (liječnik i medicinski tehničar/sestra)
- (b) kreirati medicinske timove u kojima grupa liječnike i medicinske tehničare/sestre

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o zakazanim i slobodnim terminima

7. Sustav (sudionik):

- (a) generira izvješća o učinkovitosti rezervacije i šalje ih u informacijski sustav HZZO-a

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Registrirati korisnika u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne podatke
 3. Pristiskom na dugme potvrđuje registraciju
 4. Po uspješnoj registraciji korisnik zaprima potvrdu na registriranu adresu e-pošte
 5. Pristup korisničkim funkcijama sukladno njegovom korisničkom tipu
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog e-maila, unos podataka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijaviti korisnika u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prijavu
 2. Unos korisničkog imena i lozinke
 3. Pristiskom na dugme potvrđuje prijavu
 4. Pristup korisničkim funkcijama sukladno njegovom korisničkom tipu
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime/lozinka

1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu
2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od prijave

UC3 - Pregled zakazanih termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Omogućiti pacijentu uvid u zakazane termine
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao pacijent
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za pacijente
 2. Korisnik na vlastitom kalendaru ima uvid u svoje termine

UC4 - Zakazivanje termina kod liječnika

- **Glavni sudionik:** Pacijent
- **Cilj:** Omogućiti pacijentu odabir termina pregleda kod liječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen korisnik kao pacijent
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za pacijente
 2. Korisnik odabire opciju zakazivanja pregleda kod liječnika
 3. Na prikazanom kalendaru odabire željeni slobodni termin
 4. Dodatno naznačuje tip pregleda
 5. Pristiskom na dugme potvrđuje odabrani termin
 6. Korisnik dobiva potvrdu o uspješno odabranom terminu na preferirani kanal komunikacije
 7. Korisnikov kalendar se osvježava novim podacima
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je previše puta rezervirao termine
 1. Sustav onemogućuje zakazivanje termina i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu
 - 2.b Korisnik se previše puta nije pojavio na zakazanom terminu
 1. Sustav onemogućuje zakazivanje termina i korisniku šalje odgovarajuću poruku

2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu

UC5 - Zakazivanje termina kod medicinskog tehničara

- **Glavni sudionik:** Pacijent
- **Cilj:** Omogućiti pacijentu odabir termina usluge kod medicinskog tehničara
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen korisnik kao pacijent
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za pacijente
 2. Korisnik odabire zakazivanje pregleda kod medicinskog tehničara
 3. Korisnik odabire vrstu usluge unutar predefiniranih termina
 4. Pristiskom na dugme potvrđuje odabranu uslugu
 5. Sustav korisniku dodjeljuje prvi slobodni termin
 6. Korisnik dobiva potvrdu o uspješno odabranom terminu na preferirani kanal komunikacije
 7. Korisnikov kalendar se osvježava novim podacima
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je previše puta rezervirao termine
 1. Sustav onemogućuje zakazivanje termina i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu
 - 2.b Korisnik se previše puta nije pojavio na zakazanom terminu
 1. Sustav onemogućuje zakazivanje termina i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu

UC6 - Otkazivanje termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Omogućiti pacijentu otkazivanje termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen korisnik kao pacijent, postoji zakazan pregled
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranicu za pacijente
 2. Korisnik na prikazanom kalendaru odabire zakazani termin
 3. Korisnik odabire opciju za otkazivanje pregleda
 4. Pristiskom na dugme potvrđuje odabranu opciju

5. Korisnik dobiva potvrdu o uspješno otkazanom terminu na preferirani kanal komunikacije
6. Korisnikov kalendar se osvježava novim podacima
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je pokušao otkazati termin 24 sata nakon zakazivanja
 1. Sustav onemogućuje otkazivanje i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu

UC7 - Potvrđivanje novih termina nakon pomicanja

- **Glavni sudionik:** Pacijent
- **Cilj:** Pacijentu omogućiti potvrdu ili odbijanje novog termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prikavljen kao pacijent, liječnici tim je pomaknuo zakazani termin
- **Opis osnovnog tijeka:**
 1. Sustav korisniku šalje poruku o promjenjenom terminu
 2. Korisniku se prikaže naslovna stranicu za pacijente
 3. Korisniku se u posebnom prozoru prikaže poruka o promjeni uz opcije prihvaćanja promjene
 4. Korisnik odabire opciju za prihvaćanje novog termina
 5. Pristiskom na dugme potvrđuje odabranu opciju
 6. Korisnik dobiva potvrdu o uspješno prihvaćenom ili odbijenom terminu na preferirani kanal komunikacije
 7. Korisnikov kalendar se osvježava novim podacima

UC8 - Podsjećanje pacijenta na zakazani termin

- **Glavni sudionik:** -
- **Cilj:** Slanje podsjetnika pacijentu na termin
- **Sudionici:** Baza podataka, pacijent
- **Preduvjet:** Postoji zakazani termin
- **Opis osnovnog tijeka:**
 1. Prikupljanje podataka o pregledu i preferiranom kanalu komunikacije
 2. Konverzija u željeni oblik
 3. Slanje podsjetnika pacijentu putem preferiranog kanala komunikacije
- **Opis mogućih odstupanja:**

3.a Sustav nije mogao poslati podsjetnik zbog greške u e-mail/sms sustavu

1. Sustav šalje podsjetnik putem opcionalnog kanala komunikacije

UC9 - Pregled rezeviranih termina

- **Glavni sudionik:** Liječnik, medicinski tehničar
- **Cilj:** Omogućiti medicinskom osoblju uvid u zakazane termine s dodatnim informacijama
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao liječnik ili medicinski tehničar
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za liječnike ili medicinske tehničare
 2. Pritiskom na željeni termin u kalenadru korisnik dobije više informacija

UC10 - Potvrđivanje dolaznosti pacijenta

- **Glavni sudionik:** Liječnik, medicinski tehničar
- **Cilj:** Omogućiti medicinskom osoblju potvrđivanje dolaznosti pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao liječnik ili medicinski tehničar, završio zakazani termin
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za liječnike ili medicinske tehničare
 2. Korisnik u prozoru sa prikazanim završenim terminima, potvrđuje dolazak pacijenta

UC11 - Definiranje raspoloživosti termina

- **Glavni sudionik:** Liječnik
- **Cilj:** Omogućiti liječniku definiranje raspoloživosti termina 10 dana unaprijed
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao liječnik
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za liječnike
 2. Korisnik bira opciju za uređivanje raspoloživosti termina
 3. Korisnik za 10 dana unaprijed definira raspoloživost termina
 4. Korisnik pritiskom na dugme sprema promjene
 5. Korisnikov kalendar se osvježava novim podacima

UC12 - Definiranje vlastitih pravila o rezervaciji termina

- **Glavni sudionik:** Liječnik
- **Cilj:** Omogućiti liječniku definiranje vlastita pravila o rezervaciji termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao liječnik
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za liječnike
 2. Korisnik bira opciju za uređivanje vlastitih pravila o rezervaciji termina
 3. Korisnik na padajućem izborniku bira koliko sati unaprijed pacijent mora rezervirati termin
 4. Korisnik pritiskom na dugme sprema promjene

UC13 - Definiranje termina za specifične vrste usluga

- **Glavni sudionik:** Medicinski tehničar/sestra
- **Cilj:** Omogućiti medicinskom tehničaru definiranje slobodnih termina za specifične vrste usluga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao medicinski tehničar
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za medicinske tehničare
 2. Korisnik bira opciju za uređivanje slobodnih termina
 3. Korisnik za 10 dana unaprijed definira slobodne termine za specifične vrste usluga
 4. Korisnik pritiskom na dugme sprema promjene
 5. Korisnikov kalendar se osvježava novim podacima

UC14 - Pomicanje zakazanih termina

- **Glavni sudionik:** Medicinski tehničar/sestra, Liječnik
- **Cilj:** Omogućiti medicinskom tehničaru ili liječniku pomicanje zakazanog termina do 24 sata nakon rezervacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao medicinski tehničar ili liječnik
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za medicinske tehničare ili liječnike
 2. Korisnik na vlastitom kalendaru bira termin koji želi pomaknuti
 3. Korisnik odabire opciju za mijenjanje termina i bira novi termin
 4. Korisnik pritiskom na dugme sprema promjene

5. Korisnikov kalendar se osvježava novim podacima
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je odabrao termin od čije je rezervacije prošlo više od 24 sata
 1. Sustav onemogućuje pomicanje termina i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na svoju naslovnu stranicu

UC15 - Stvaranje liječnika

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru stvaranje specijaliziranog tipa korisnika - liječnika, s omogućenim posebnim aktivnostima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za administratore
 2. Korisnik bira opciju za stvaranje liječnika
 3. U formi unosi podatke o novom liječniku
 4. Korisnik pritiskom na dugme sprema promjene
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog e-maila, unos podataka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za stvaranje novog liječnika
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od stvaranja novog liječnika

UC16 - Stvaranje medicinskog tehničara/sestre

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru stvaranje specijaliziranog tipa korisnika - medicinskog tehničara/sestre, s omogućenim posebnim aktivnostima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za administratore
 2. Korisnik bira opciju za stvaranje medicinskog tehničara
 3. U formi unosi podatke o novom medicinskom tehničaru

4. Korisnik pritiskom na dugme sprema promjene
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog e-maila, unos podataka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za stvaranje novog medicinskog tehničara
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od stvaranja novog medicinskog tehničara

UC17 - Pregled medicinskih timova

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru pregled medicinskih timova
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za administratore
 2. Korisnik bira opciju za pregled medicinskog timova

UC18 - Brisanje medicinskog tima

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru brisanje medicinskog tima, čime medicinsko osoblje čini slobodnim za grupiranje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao administrator, u bazi postoji barem jedan medicinski tim
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za administratore
 2. Korisnik bira opciju za pregled medicinskih timova
 3. Korisnik za odabarni tim pritisne brisanje tima
 4. Korisnik pritiskom na dugme sprema promjene
 5. Korisnikov popis timova se osvježi novim podacima

UC19 - Stvaranje novog medicinskog tima

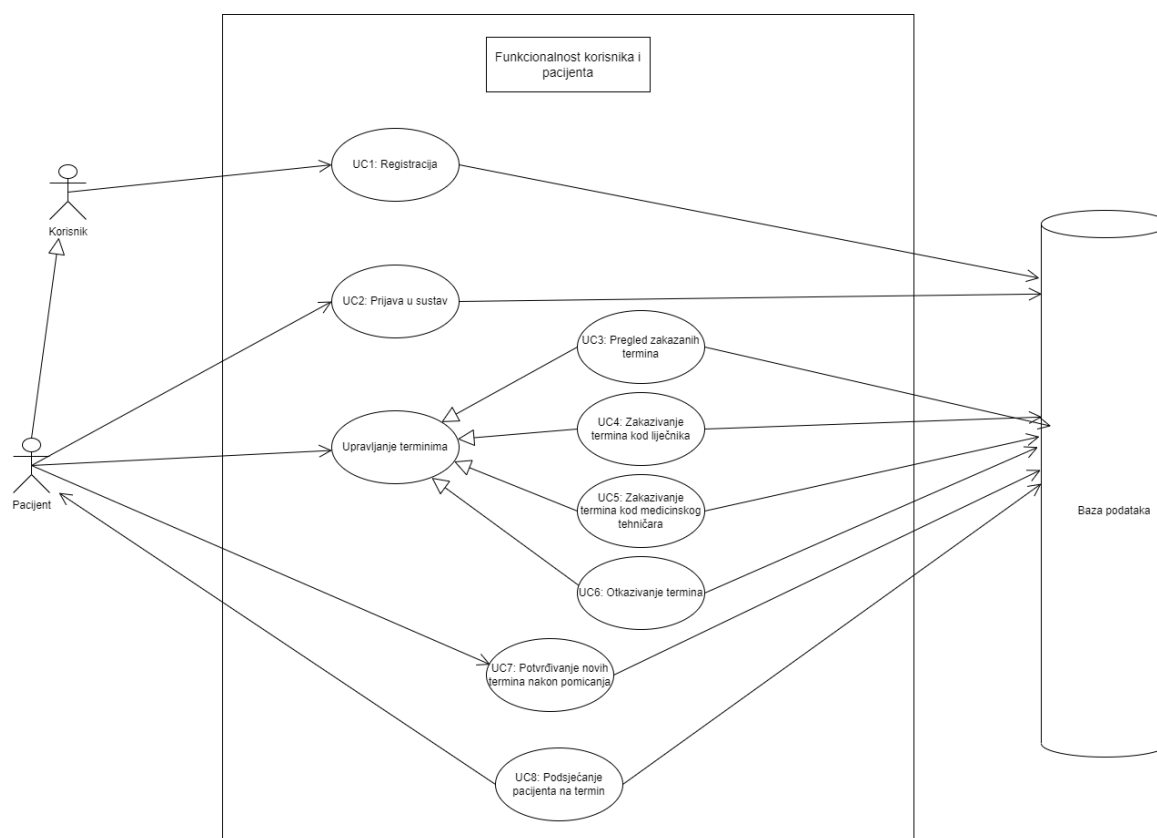
- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru grupiranje liječnika i medicinskog tehničara u medicinski tim

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao administrator, u bazi postoji barem jedan liječnik i medicinski tehničar
- **Opis osnovnog tijeka:**
 1. Korisniku se prikaže naslovna stranica za administratorea
 2. Korisnik bira opciju za stvaranje novog medicinskog tima
 3. U formi unosi imena liječnika i medicinskog tehničara koje želi grupirati
 4. Korisnik pritiskom na dugme sprema promjene
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je odabrao liječnika ili medicinskog tehničara koji već je u nekom timu
 1. Sustav onemogućuje stvaranje tima i korisniku šalje odgovarajuću poruku
 2. Korisnik je preusmjeren natrag na stvaranje novog tima

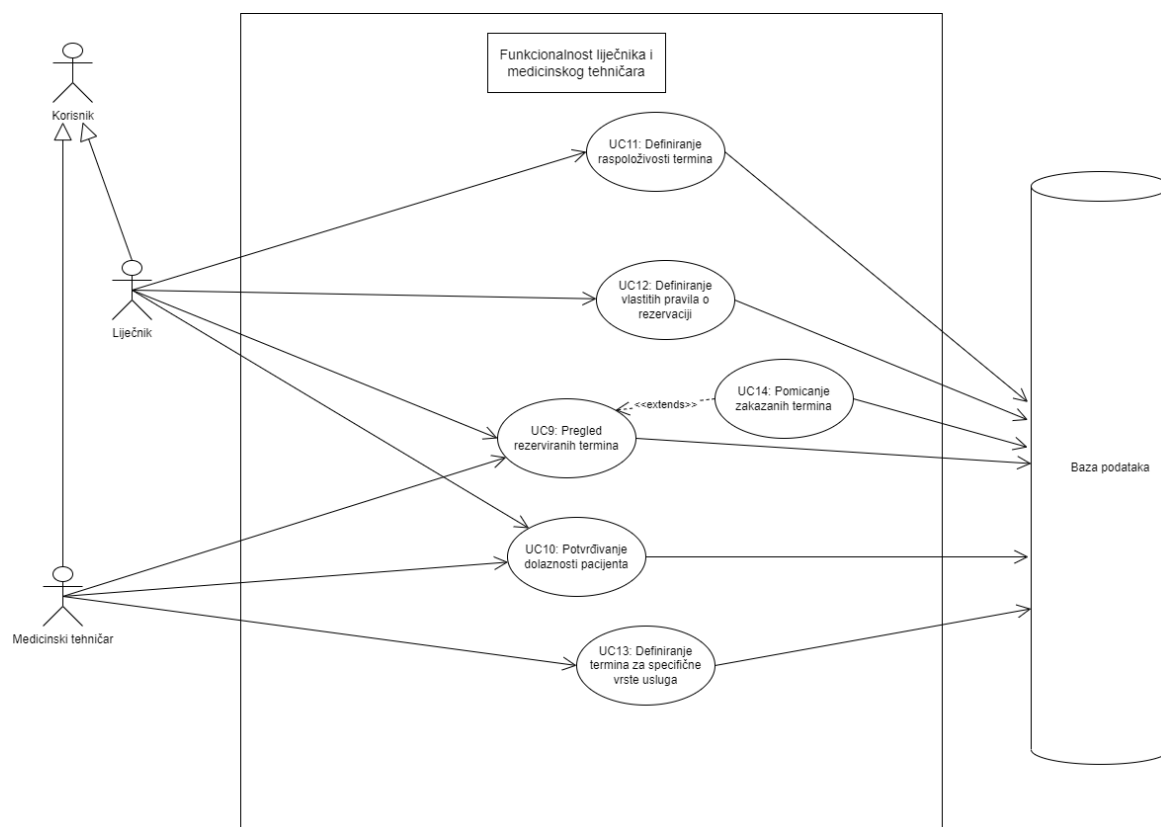
UC20 - Generiranje izvješća

- **Glavni sudionik:** -
- **Cilj:** Generiranje dnevnih i mjesečnih izvješća i slanje u informacijski sustav HZZO-a
- **Sudionici:** Baza podataka, HZZO
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Prikupljanje podataka unesenih u proteklom razdoblju
 2. Konverzija podataka u željeni oblik
 3. Slanje podataka HZZO-u
- **Opis mogućih odstupanja:**
 - 2.a Informacijski sustav HZZO-a nije dostupan
 1. Sustav čeka 1 sat i pokušava ponovo

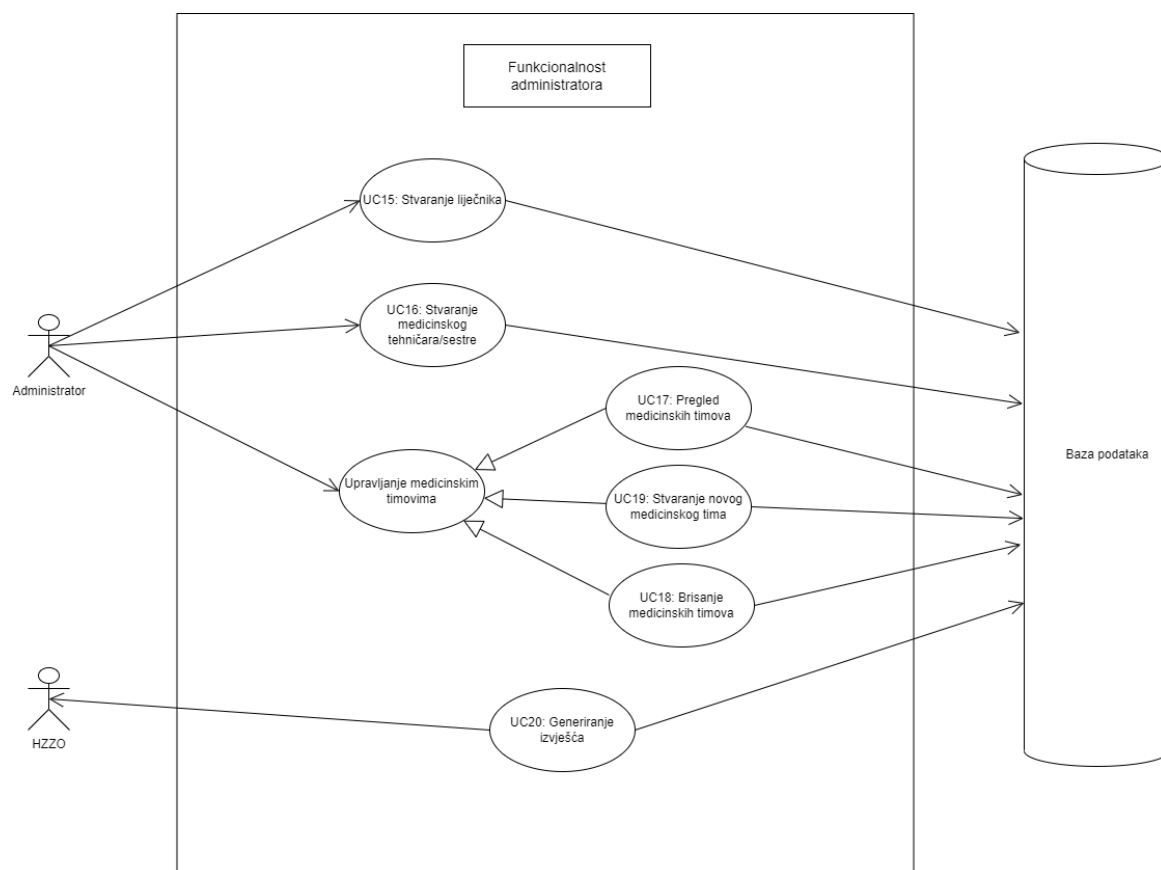
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost korisnika i pacijenta



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost liječnika i medicinskog tehničara/sestre

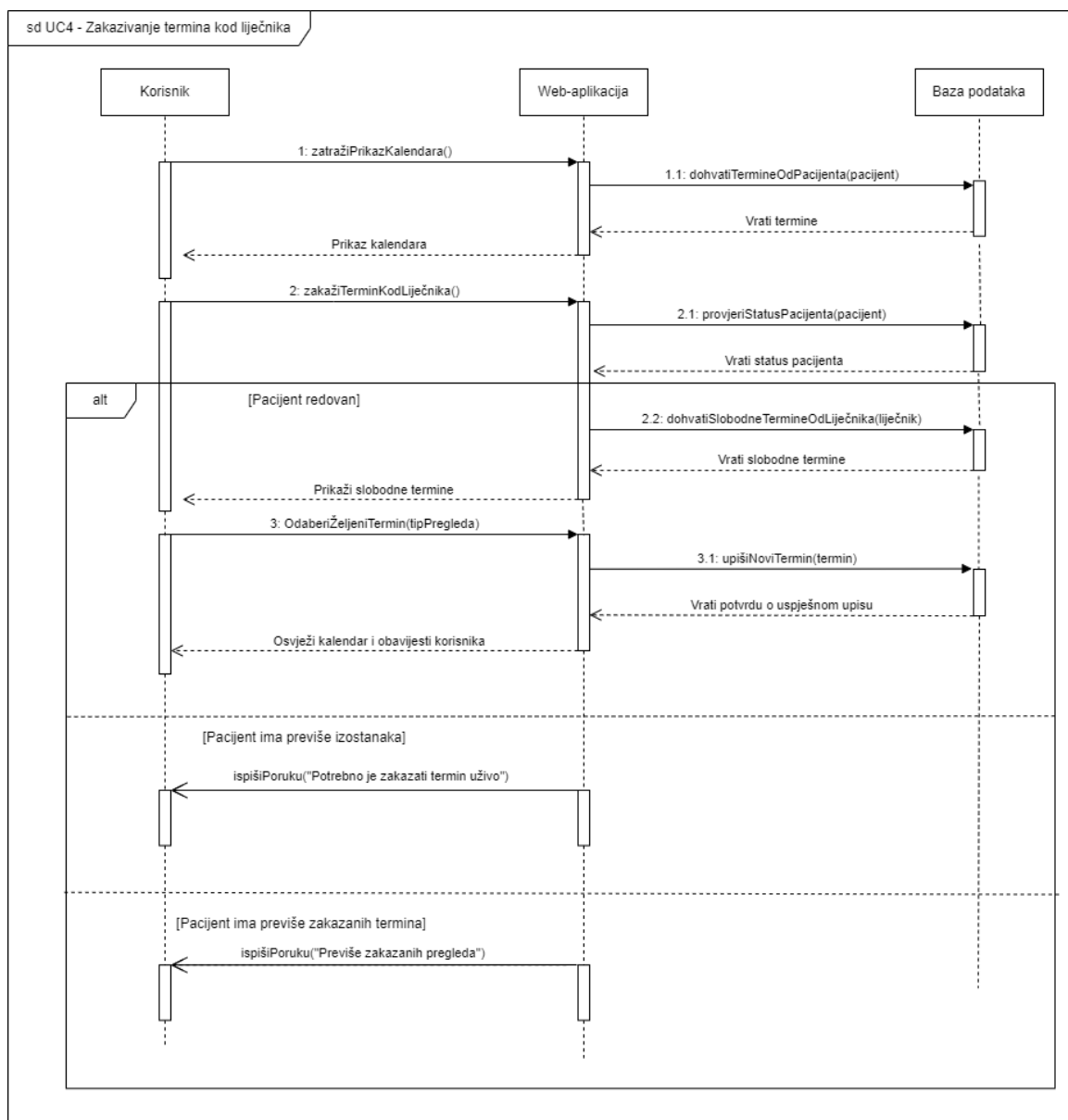


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC4 - Zakazivanje termina kod liječnika

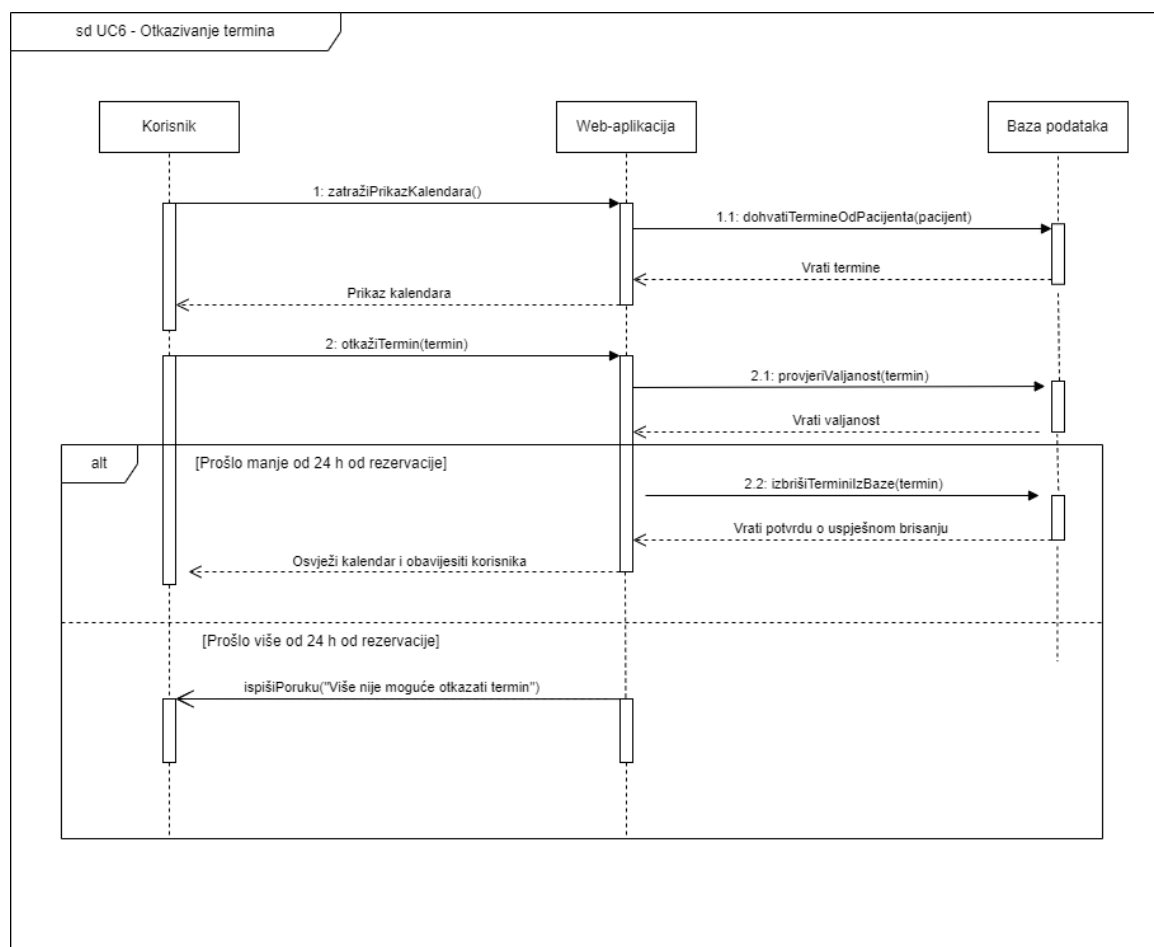
Korisnik, prijavljen kao pacijent, želi vidjeti kalendar sa svojim terminima kako bi mogao zakazati sljedeći termin. Poslužitelj dohvaća trenutne termine i prikazuje ih. Korisnik želi zatražiti novi termin kod liječnika, međutim poslužitelj najprije mora provjeriti koji je status korisnika. Ako je sve u redu, pacijent je redovan, i poslužitelj dohvaća slobodne termine korisnikovog liječnika iz baze podataka i prikazuje ih korisniku. Korisnik bira željeni termin i dodatno naznačuje tip pregleda. Poslužitelj šalje nove podatke u bazu te obavještava korisnika te mu osvježava kalendar. Ako je u korisnikovom statusu označeno da ima previše izostanaka sa zakazanih termina ili je zakazao previše termina, poslužitelj odbija zakazati novi termin i šalje odgovarajuću poruku korisniku.



Slika 3.4: Sekvencijski dijagram za UC4

Obrazac uporabe UC6 - Otkazivanje termina

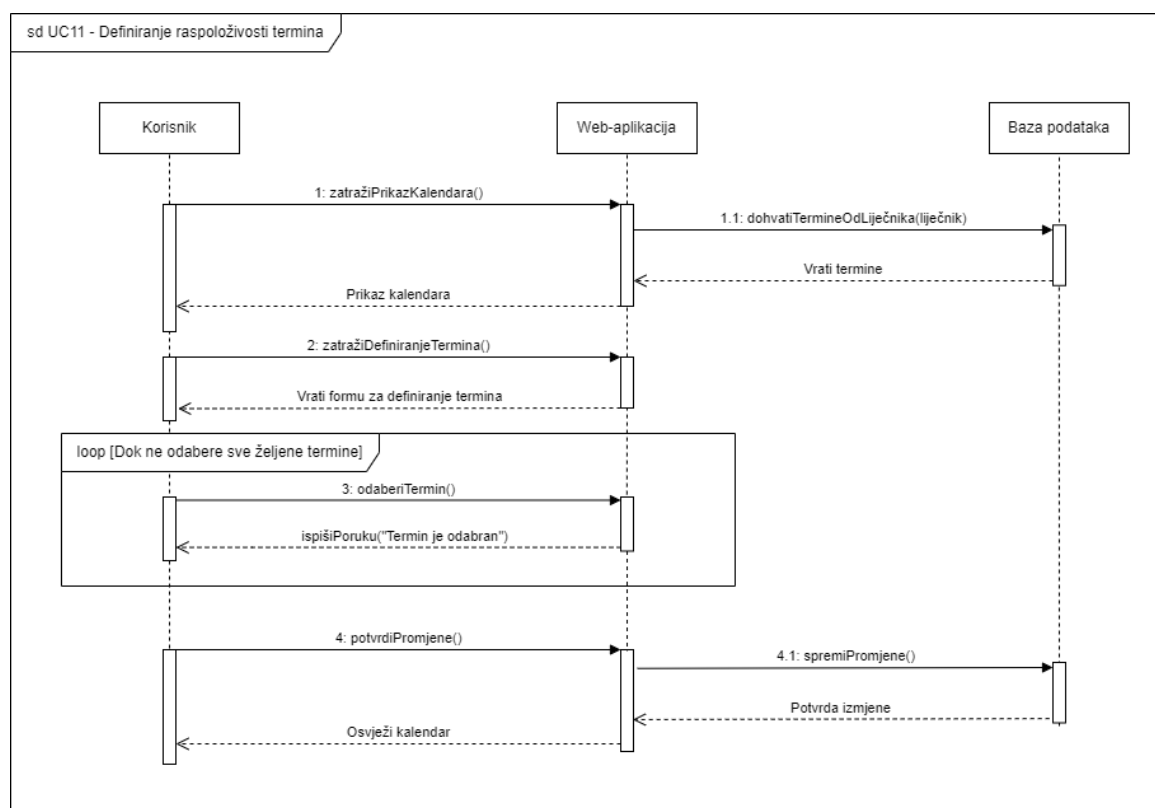
Korisnik, prijavljen kao pacijent, želi vidjeti kalendar sa svojim terminima kako bi mogao otkazati sljedeći termin. Poslužitelj dohvaća trenutne termine i prikazuje ih. Korisnik želi otkazati termin te na svom kalendaru pronalazi željeni termin i otkazuje ga. Poslužitelj tada najprije provjera koliko je vremena prošlo od rezervacije zadanog termina. Ako je prošlo više od 24 sata, tada poslužitelj odbija otkazati termin uz odgovarajuću poruku korisniku. Inače, poslužitelj šalje bazi podatak o terminu kojeg treba obrisati, te na potvrdu baze o uspješnom brisanju obavještava korisnika i osvježava mu kalendar.



Slika 3.5: Sekvencijski dijagram za UC6

Obrazac uporabe UC11 - Definiranje raspoloživosti termina

Korisnik, prijavljen kao liječnik, želi vidjeti kalendar kako bi imao uvid u svoje termine i mogao definirati njihovu raspoloživost. Poslužitelj dohvaća trenutne termine i prikazuje ih. Nakon što korisnik odluči definirati termine, poslužitelj mu vraća formu koja mu to omogućuje. Korisnik ima mogućnost definiranja termina sve dok ne potvrdi promjene. Nakon što korisnik potvrdi promjene, poslužitelj signalizira bazi da ih spremi. Poslije potvrde baze, poslužitelj osvježava korisnikov kalendar.



Slika 3.6: Sekvencijski dijagram za UC11

3.2 Ostali zahtjevi

- Sustav treba biti izveden kao web aplikacija kojoj će korisnici pristupati uz pomoć korisničkog imena i lozinke
- Oblikovanje aplikacije mora slijediti načela objektno-orijentiranog programiranja
- Aplikacija treba biti jednostavna za korištenje, a sučelje pregledno i intuitivno
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Sustav treba podržavati rad više korisnika u stvarnom vremenu
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Generirana dnevna i mjesečna izvješća o učinkovitosti šalju se u XML formatu

4. Arhitektura i dizajn sustava

Arhitekturu projektnog sustava možemo podijeliti u 3 cjeline:

- *Web aplikacija*
- *Web poslužitelj*
- *Baza podataka*

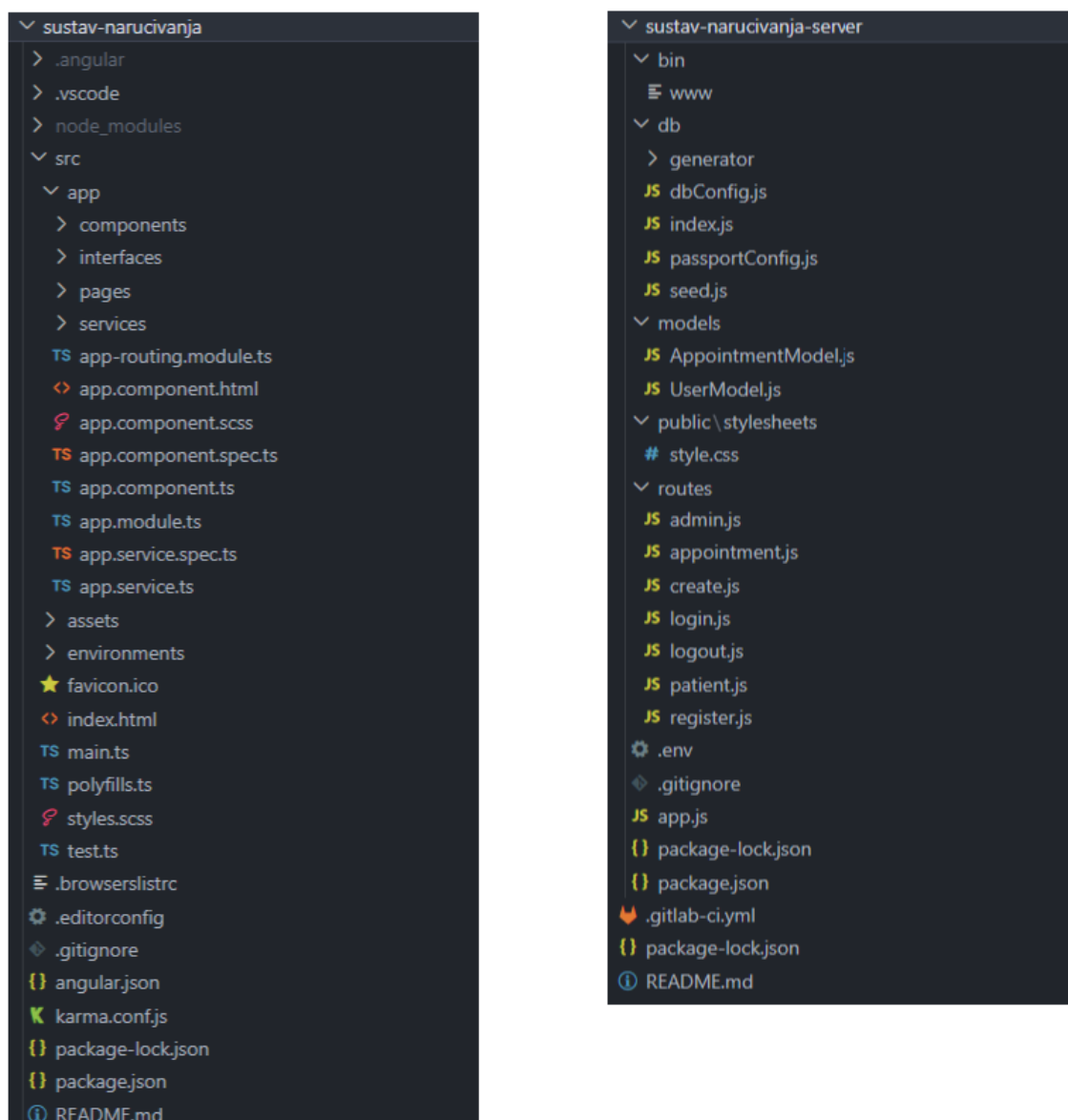
Web preglednik je dio arhitekture koji korisniku omogućuje pregled web stranice i svih sadržaja na njoj. Svaki internetski preglednik je prevoditelj. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. Njegov primaran zadatak je komunikacija klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. Hyper Text Transfer Protocol) protokola, što je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev

Korisnik koristi web aplikaciju za obrađivanje željenih zahtjeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu, pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Null grupa 2022./2023. za projekt na predmetu Programsko inženjerstvo 2022./2023. odabrala je Angular sustav za izradu frontenda (dijela sustava vidljivoga korisniku) i Express sustav za izradu backenda (dijela sustava koji nije vidljiv korisniku) projektnog zadatka. Arhitektura AngularJS-a temelji se na MVC dizajnu (Model, view, controller). Model View Controller ili MVC kako se popularno naziva, uzorak je dizajna softvera za razvoj web aplikacija. MVC dizajn sastoji se od sljedeća tri dijela:

- Model je najniža razina uzorka odgovornog za održavanje podataka.
- Prikaz(view) odgovoran je za prikazivanje svih ili dijela podataka korisniku.
- Kontroler je softverski kod koji kontrolira interakcije između modela i prikaza.



Slika 4.1: Struktura implementacije aplikacije iz VSC.

Aplikaciju smo rastavili na komponente, sučelja, stranice i servise. Komponente su implementirani dijelovi stranice koje možemo ubaciti kao cjelinu u neku od pages(stranica) kako ne bismo morali pisati dupli kod. Pages(stranice) su sve stranice u našoj aplikaciji. Svaki "page" sastavljen je od '.html' datoteke za strukturu, '.scss' datoteke za dizajn, '.module.ts' u kojoj možemo Typescriptom napraviti uvoz pojedinog modula Angulara u stranicu i '.ts' u kojoj definiramo putanje i povezivanje svih dokumenata.

Backend dio aplikacije izveden je u objektno orijentiranoj paradigmi. Pacijenti, doktori, medicinski tehničari te administratori su predstavljeni objektima

koji nasljeđuju od zajedničkog roditelja, klase korisnik. Sve podklase klase korisnik znaju provjeriti svoju lozinku i korisničko ime, spremi se u bazu te druge detalje koji su vezani za pojedinu klasu. Grupe i pregledi također su izvedeni kao klase koje sadrže metode za spremanje i dohvaćanje iz baze podataka.

Razvojno okruženje koje koristite programeri Null grupe je Visual Studio Code. Za izradu i doradu koda programeri koriste GitLab sustav. Za razvoj dokumentacije grupa će koristiti Overleaf aplikaciju radi sistematičnosti izrade.

4.1 Baza podataka

PostgreSQL baza podataka sastoji se od 7 tablica. Entitet users je supertip koji se disjointed grana na četiri tablice: admin, patient, doctor i nurse. Primarni ključ svih tih entiteta je id koji se u tablici user automatski generira. Tablica patient također sadrži strani ključ doctorid, jer je pacijent pri registraciji obavezan izabrati svog liječnika. Tablica nurse ima strani ključ teamid, referencu na tablicu team, jer u jednom timu kojeg administrator određuje može raditi više medicinskih tehničara. Također ta vrijednost može biti null jer tehničar ne mora nužno biti dio tima. Isto vrijedi i za liječnika. Osim tih tablica također baza podataka sadrži već prije spomenutu tablicu team te appointment. Primarni ključ tima je teamid, a tablica team također sadrži ime tima pod nazivom varijable name. Tablica appointment sadrži primarni ključ appointmentId koji se automatski generira te tri strana ključa. Patientid koji obavezno nije null, referenca na pacijenta koji dolazi na pregled. Druga dva strana ključa doctorId i nurseId su međusobno isključivi, odnosno ako jedan ima vrijednost, drugi mora biti null, jer pacijent dolazi na pregled ili kod liječnika ili kod tehničara. Pregled je također određen timestampom time, vremenom kad je ugovoren i intervalom duration.

4.1.1 Opis tablica

users		
id	INT	Jedinstveni id svakog korisnika, generira se automatski, primarni ključ tablice korisnik
name	VARCHAR	Ime korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

users		
surname	VARCHAR	Prezime korisnika
phoneNumber	VARCHAR	Telefonski broj korisnika, jedinstven ta svakog korisnika
mail	VARCHAR	Adresa elektroničke pošte korisnika, jedinstvena za svakog korisnika
password	VARCHAR	Lozinka korisnika za prijavu u sustav
sex	VARCHAR	spol korisnika
dateOfBirth	DATE	Datum rođenja korisnika

admin		
id	INT	Strani ključ, referenca na id u tablici users

patient		
nFailedAppointments	INT	Broj ugovorenih pregleda koje je pacijent propustio
id	INT	Strani ključ, referenca na id u tablici users
doctorid	INT	ID doktora kojeg je pacijent izabrao pri registraciji

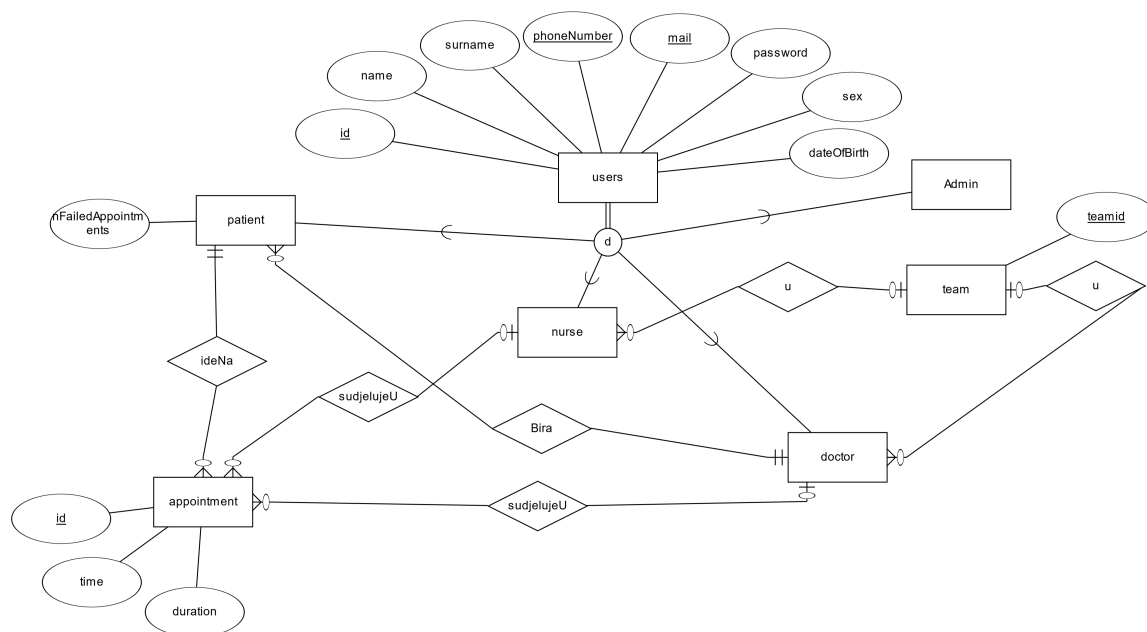
doctor		
id	INT	Strani ključ, referenca na id u tablici users
teamid	INT	Id tima kojem je doktor dodjeljen, može biti null

nurse		
id	INT	Strani ključ, referenca na id u tablici users
teamid	INT	Id tima kojem je medicinska sestra dodjeljena, može biti null

team		
teamid	teamid	Id tima
name	VARCHAR	Ime tima

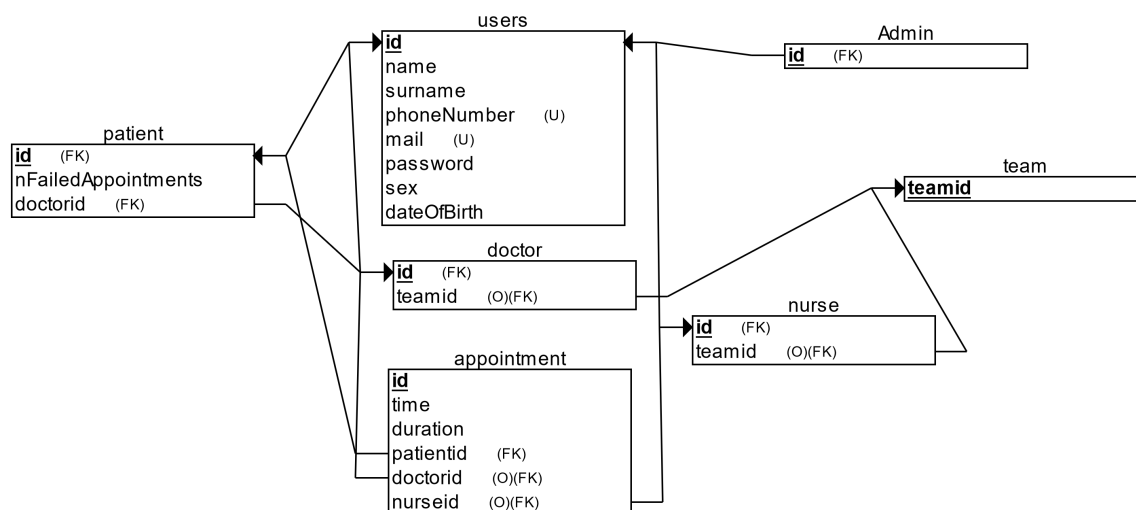
appointment		
id	INT	Jedinstveni ID pregleda, generira se automatski
patientid	INT	Strani ključ, ID pacijenta koji je zakazao pregled, ne može biti null
doctorid	INT	Strani ključ, ID doktora koji vrši pregled
nurseid	INT	Strani ključ, ID medicinskog tehničara koji vrši pregled
time	TIMESTAMP	Vrijeme u koje je pregled ugovoren
duration	INTERVAL	Trajanje pregleda

4.1.2 Dijagram baze podataka



Slika 4.2: ER dijagram baze podataka

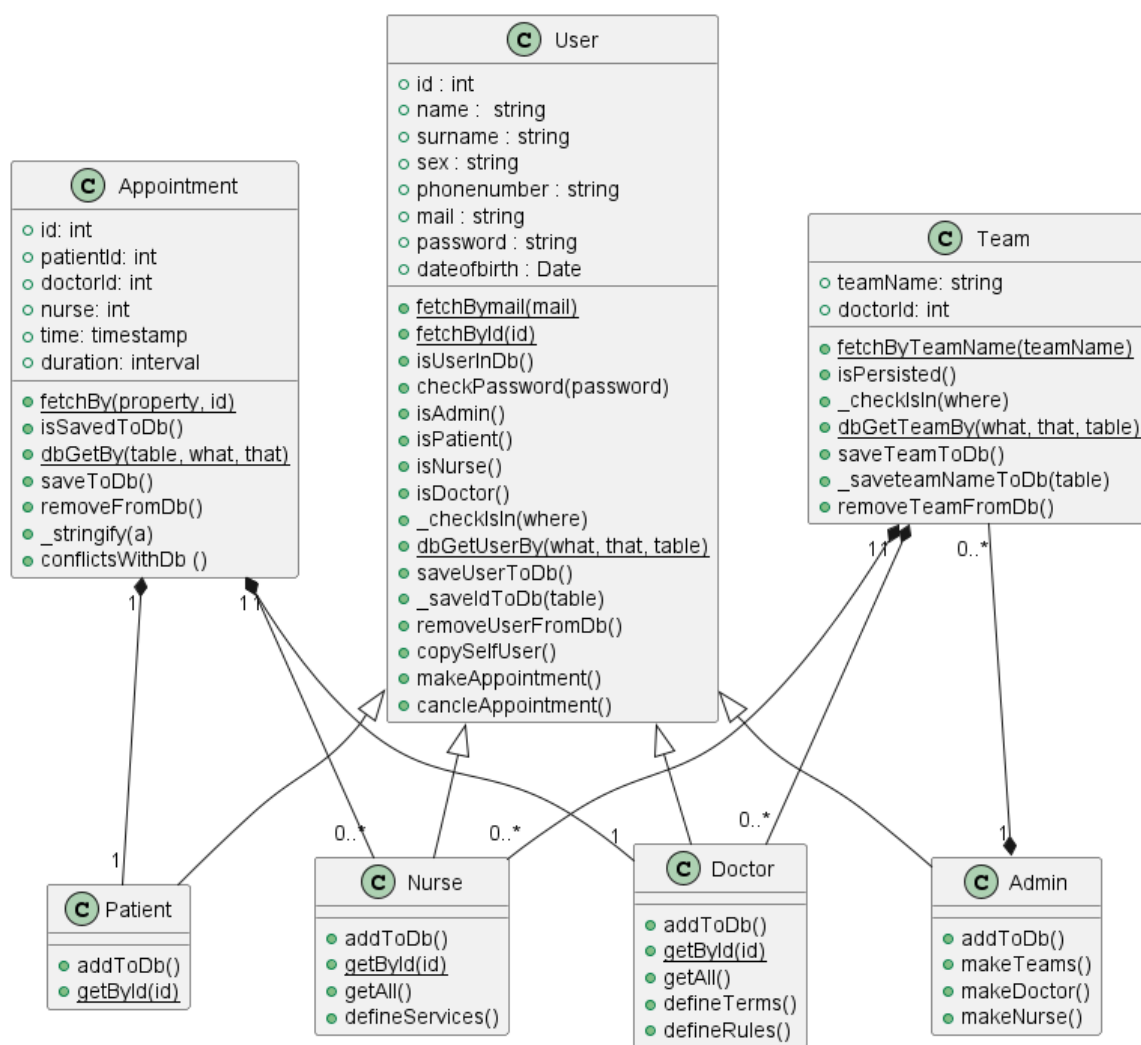
4.1.3 Dijagram baze podataka



Slika 4.3: Relacijski dijagram baze podataka

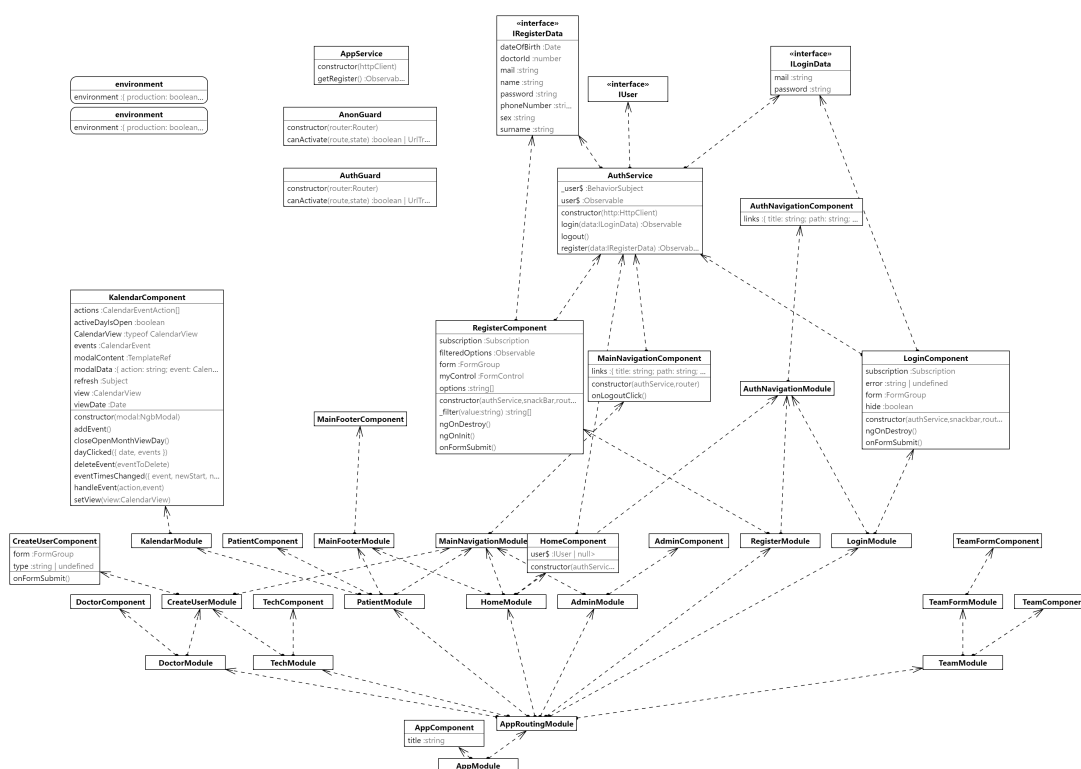
4.2 Dijagram razreda

Na slici 4.4 prikazan je dijagram razreda koji pripada backend dijelu arhitekture. Razredi iz dijagrama predstavljaju strukturu baze podataka u aplikaciji te svaki razred predstavlja jedan entitet iz baze. Metode koje su implementirane komuniciraju s bazom podataka, ako je potrebno, upisuju odgovarajuće podatke u bazu ili ih vraćaju iz baze. Razred User predstavlja svakog korisnika aplikacije i koji može koristiti određene funkcionalnosti sustava. Međutim, prije toga se mora registrirati s osobnim podacima: ime, prezime, spol, broj telefona, e-mail, lozinka i datum rođenja. Razred Pacijent predstavlja korisnika koji je registriran kao pacijent i ima pristup svojim terminima uz funkcionalnosti zakazivanja i otkazivanja termina, kojeg predstavlja razred Appointment. Nešto veću ovlast ima medicinska sestra koju predstavlja razred Nurse, ona može i unaprijed definirati termine usluga. Razred Doctor predstavlja liječnika koji može definirati slobodne termine pregleda, ali i definirati pravila za naručivanje. Najvišu ovlast ima administrator, kojeg predstavlja razred Admin. On može stvarati timove, koje predstavlja razred Teams, te upisati nove liječnike i medicinske sestre u bazu.



Slika 4.4: Dijagram razreda - Modeli

Dijagram razreda na slici 4.5 generirali smo pomoću 'Classdiagram -ts' ekstenzije autora Alexa Shena u razvojnom okruženju Visual Studio Code. Dijagram prikazuje strukturu frontend dijela aplikacije. Angular je framework koji se bazira na komponentama, te svaka komponenta, koja je osnovni i gradivni dio, koristi neki od modula koji ih povezuju. Tako korijenski modul aplikacije, AppRoutingModule koristi svi nadređene module kao što su HomeComponent, TeamModule, AdminModule, PatientModule, TechModule i DoctorModule, a svaka od njih koriste MainFooterModule i MainNavigationModule. Korištenjem komponenata smanjujemo redundanciju u kodu. Ovisno o tome koje ovlasti ima, bio on doktor, medicinska sestra pacijent ili administrator, korisniku su prikazane različite komponente. Na primjer, pacijent ima pristup komponentama: KalendarComponent, MainNavigationComponent i MainFooterComponent. Za svaku komponentu korisniku su prikazani njihovi html predlošci, css oblikovanja i funkcionalnosti definirane u pojedinom dijelu komponente.



Slika 4.5: Dijagram razreda - frontend

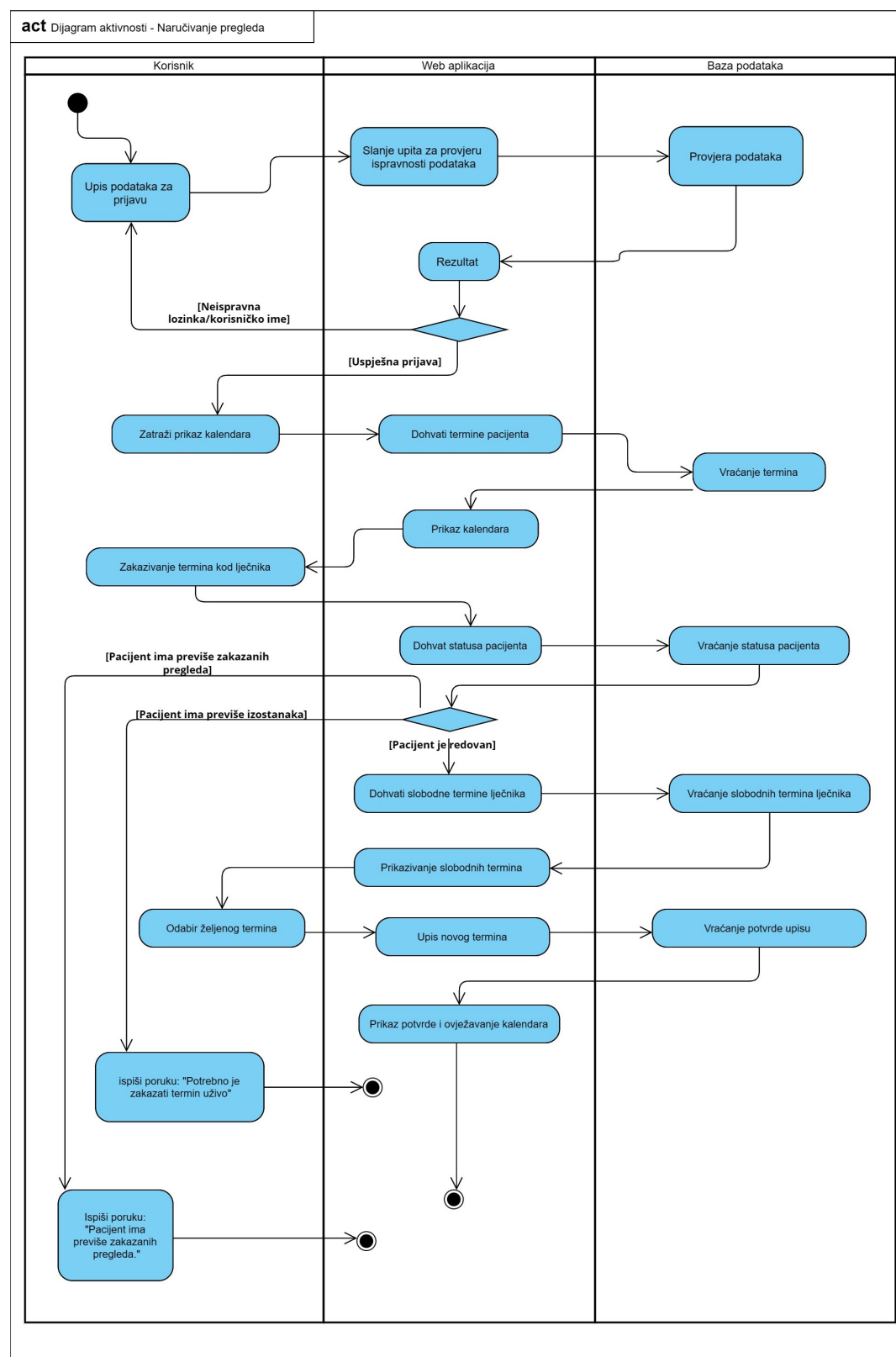
Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti



4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu postignuta je izradom Discord servera NULL grupe 2022./2023. i WhatsApp grupe kojima su pristup imali svi članovi tima. Ova dva kanala komunikacije omogućavali su nam brzu i jednostavnu komunikaciju, te spremnost svakog člana na pomoć drugome u nevolji u kratkom vremenu. Za izradu UML dijagrama koristili smo Draw.io, Astah Professional, Visual Paradigm, Microsoft Paint itd... Sustav za upravljanje, upravljan je izvornim kodom Git. Repozitorij projekta dostupan je na web platformi GitLab.

Razvojno okruženje korišteno pri izradi je Visual Studio Code, integrirano razvojno okruženje tvrtke Microsoft.

Prvenstveno se koristi za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, web-usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi Microsoft-ove platforme kao što su Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Backend domena izrađena je pomoću radnog okvira Node.js i jezika Javascript, a frontend domena korištenjem Angular okvira, te jezikom Typescript.

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitivanje komponenti ostvareno je korištenjem Jest alata za ispitivanje. U nastavku su opisani provedeni testovi te priložene funkcija za testiranje te izlaz koji se dobije testiranjem. Prikazani testovi pokrivaju klase `UserModel` i `AppointmentModel` koje implementiraju najvažnije funkcionalnosti aplikacije

```
sustav-narucivanja-server > models > JS UserModel.test.js > ...
1  const { User, Patient, Doctor, Nurse, Admin } = require("../models/UserModel");
2  const db = require("../db")
3
4
5  const cnt_users = async () => {
6    const sql = "SELECT * FROM users";
7    const result = await db.query(sql, []);
8    return result.length
9  }
10
11
12  test('User fetching by id', async () => {
13    let user = await User.fetchById(1);
14    expected_user = {
15      id:1,
16      name:'Julijan',
17      surname:'Zlataric',
18      sex:'M',
19      phonenumber:'0971012013',
20      mail:'JulijanZlataric@gmail.com',
21      password:'11112013',
22    }
23    for (const key in expected_user)
24      expect(user[key]).toBe(expected_user[key]);
25  });
26
27
28  test('User insertion and deletion', async () => {
29    let u = new User(
30      (id = undefined),
31      (name = "ante"),
32      (surname = "iva"),
33      (sex = "M"),
34      (phonenumber = "asdaf"),
35      (mail = "fff"),
36      (password = "password"),
37      (dateofbirth = "2020-01-01")
38    );
39
40    let users_before = await cnt_users()
41    await u.saveUserToDb();
42    expect((await cnt_users())).toBe(users_before+1);
43
44    await u.removeUserFromDb();
45    expect((await cnt_users())).toBe(users_before);
46  });
47
```

Slika 5.1: Prikazani testovi testiraju dohvaćanje usera iz baze podataka te njegovo ubacivanje i izbacivanje iz baze podataka. Navedene funkcionalnosti su nužne za pravilnu registraciju i login korisnika aplikacije.

```
PASS ./UserModel.test.js
  ✓ User fetching by id (46 ms)
  ✓ User insertion and deletion (31 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.338 s, estimated 1 s
Ran all test suites matching /UserModel/i.
```

Slika 5.2: Izlaz programa za testiranje klase UserModel s 2 testna primjera. Sa slike je vidljivo da klasa prolazi testne primjere.

```
sustav-narucivanja-server > models > js AppointmentModel.test.js > ...
1  const Appointment = require("../AppointmentModel");
2  const db = require("../db")
3
4
5  // fetching
6  test('Appointment fetching by id', async () => {
7    let app = (await Appointment.fetchBy('id', 1))[0];
8    expected_app = {
9      patientid:100,
10     doctorid:7,
11     nurseid:null,
12     time:"Sat Jan 10 2015 01:51:14 GMT+0100 (Central European Standard Time)",
13     duration:{minutes:20},
14     created_on: null,
15     changes_from: null,
16     patient_came: null,
17   }
18   for (const key in expected_app){
19     expect(app[key]+"").toBe(expected_app[key]+"");
20   }
21 });
22
23
24 // confliction detection
25 test('Appointment confliction detection - same time', async () => {
26   let app = (await Appointment.fetchBy('id', 1))[0];
27   app.id = undefined
28   app.time = app.time.toISOString()
29   app.duration = '00:01:00'
30   expect((await app.conflictsWithDb()+"").toBe(true+""));
31 });
32
33 test('Appointment confliction detection - same id', async () => {
34   let app = (await Appointment.fetchBy('id', 1))[0];
35   app.time = app.time.toISOString()
36   app.duration = '00:01:00'
37   expect((await app.conflictsWithDb()+"").toBe(false+""));
38 });
39
40 test('Appointment confliction detection - overlapping', async () => {
41   let app = (await Appointment.fetchBy('id', 1))[0];
42   app.id = undefined
43   app.time.setMinutes(app.time.getMinutes() + 10);
44   app.time = app.time.toISOString()
45   app.duration = '00:01:00'
46   expect((await app.conflictsWithDb()+"").toBe(true+""));
47 });
48
49 test('Appointment confliction detection - non overlapping', async () => {
50   let app = (await Appointment.fetchBy('id', 1))[0];
51   app.id = undefined
52   app.time.setMinutes(app.time.getMinutes() + 60*24*365*200);
53   app.time = app.time.toISOString()
54   app.duration = '00:01:00'
55   expect((await app.conflictsWithDb()+"").toBe(false+""));
56 });
57
58 // appointment change, used in reserving, canceling...
59 test('Appointment update', async () => {
60   let app = (await Appointment.fetchBy('id', 1))[0];
61   app.type = "test"
62   await app.updateDb()
63   app2 = (await Appointment.fetchBy('id', 1))[0];
64   for (const key in app2)
65     expect(app[key]+"").toBe(app2[key]+"");
66   app.type = undefined
67   await app.updateDb()
68 });
69
```

Slika 5.3: Prikazani testovi testiraju dohvaćanje određenog termina iz baze podataka, testiranje postoji li određeni termin koji je u vremenskom konfliktu s odabranim terminom te testiranje updateanja termina. Testiranje konflikata ima više rubnih slučajeva, naime isti termin mogli smo već dohvatiti iz baze podataka te tada nebi trebalo biti konflikata, zatim moguće je da se termin nalazi unutar drugoga, potpuno izvan...

```
PASS ./AppointmentModel.test.js
✓ Appointment fetching by id (46 ms)
✓ Appointment confliction detection - same time (6 ms)
✓ Appointment confliction detection - same id (3 ms)
✓ Appointment confliction detection - overlapping (2 ms)
✓ Appointment confliction detection - non overlapping (2 ms)
✓ Appointment update (11 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        0.312 s, estimated 1 s
Ran all test suites matching /Appointment/i.
```

Slika 5.4: Izlaz programa za testiranje klase AppointmentModel s 6 testnih primjera. Sa slike je vidljivo da klasa prolazi testne primjere.

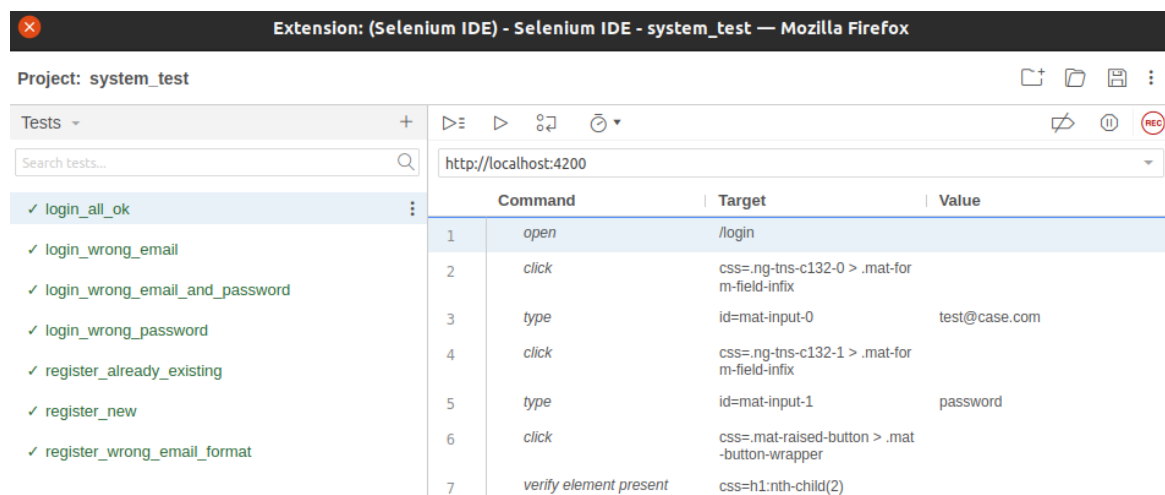
5.2.2 Ispitivanje sustava

Ispitivanje sustava provedeno je koristeći radni okvir Selenium¹ u obliku dodataka za preglednik **Selenium IDE** - preko kojeg se izvodilo snimanje korisnikovih akcija radi automatskog ponavljanja ispita. Razrađeno je **7 ispitnih slučajeva** u kojima su redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku. U nastavku su prikazani ispitni slučajevi za funkcije logiranje i registriranja. Prikazani su sljedeći testovi:

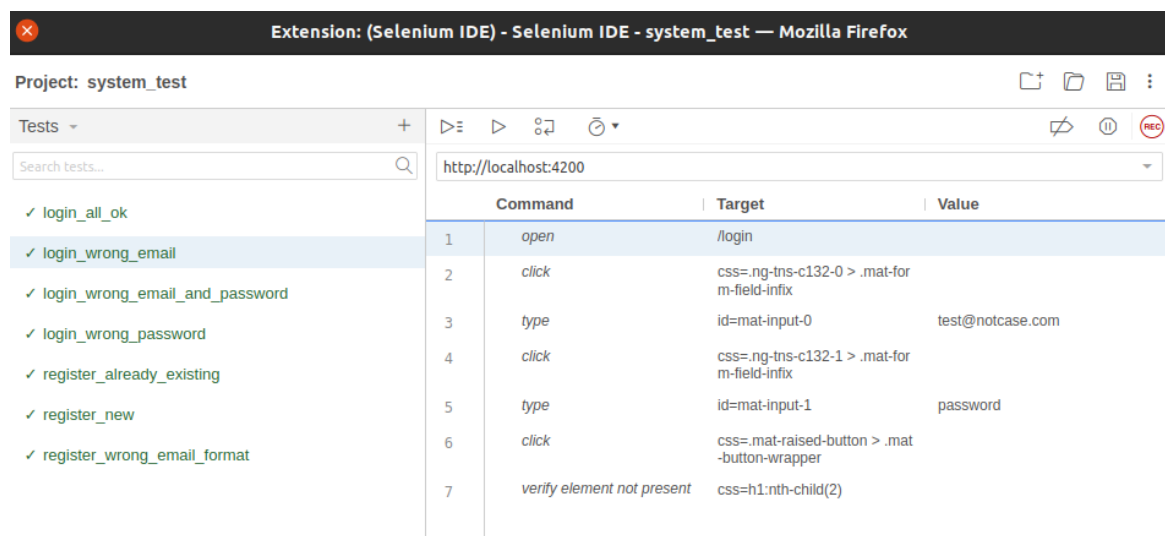
- logiranje u aplikaciju ukoliko je sve ispravno naravljeno
- logiranje u aplikaciju ukoliko je kriv e-mail
- logiranje u aplikaciju ukoliko je kriva lozinka
- logiranje u aplikaciju ukoliko je su i lozinka i e-mail krivi
- registracija korisnika gdje je sve ispravno napravljeno
- registracija korisnika koji je već registriran
- registracija korisnika gdje je e-mail zadan u krivom formatu

Svaki od testova prikazan je slikom te se potrebni ulaz te očekivani izlaz nalaze ispod slike.

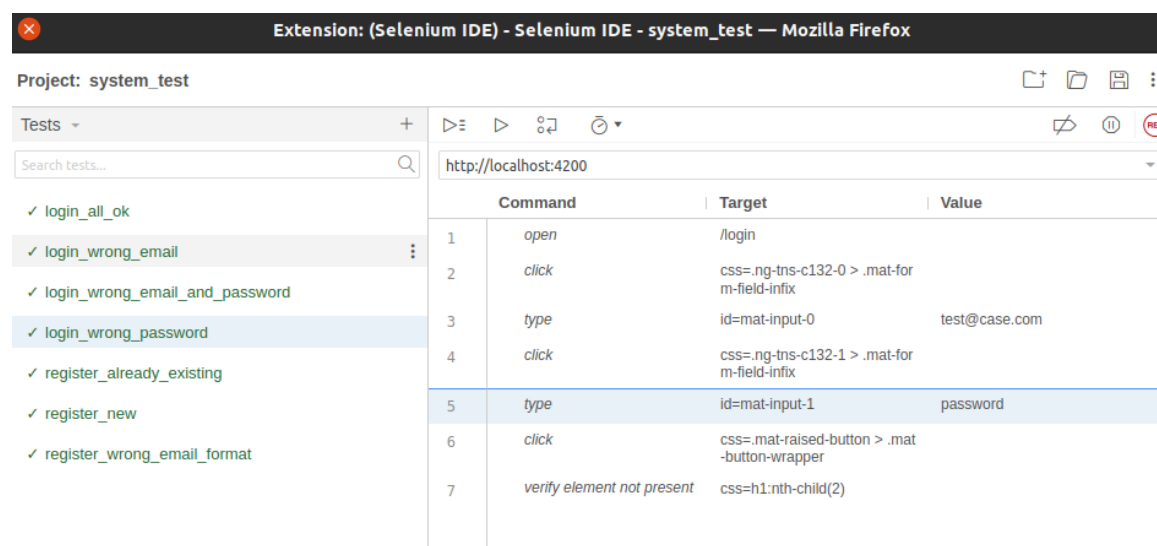
¹<https://www.seleniumhq.org/>



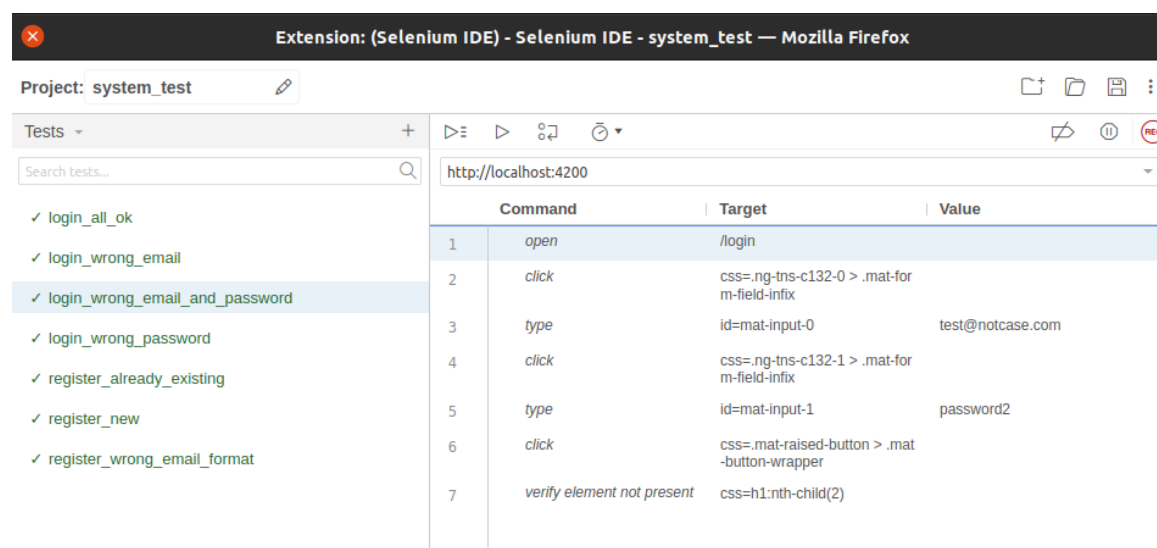
Slika 5.5: Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je sve ispravno naravljeno. Korišteni e-mail je "test@case.com" te lozinka "password". Očekivani rezultat je preusmjerenje na home page, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.



Slika 5.6: Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriv e-mail. Korišteni e-mail je "test@notcase.com" te lozinka "password". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.



Slika 5.7: Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriva lozinka. Korišteni e-mail je "test@case.com" te lozinka "password2". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.



Slika 5.8: Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriva lozinka i e-mail. Korišteni e-mail je "test@notcase.com" te lozinka "password2". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.

Extension: (Selenium IDE) - Selenium IDE - system_test — Mozilla Firefox

Project: system_test

Tests +

Search tests...

- ✓ login_all_ok
- ✓ login_wrong_email
- ✓ login_wrong_email_and_password
- ✓ login_wrong_password
- ✓ register_already_existing
- ✓ register_new
- ✓ register_wrong_email_format

http://localhost:4200

	Command	Target	Value
1	✓ open	/register	
2	✓ set window size	846x915	
3	✓ click	css=#mat-select-value-3 > .mat-select-placeholder	
4	✓ click	css=#mat-option-5 > .mat-option-text	
5	✓ click	id=mat-input-6	
6	✓ type	id=mat-input-6	123454321
7	✓ click	css=.ng-tns-c132-7 > .mat-form-field-infix	
8	✓ mouse over	css=.mat-datepicker-toggle-default-icon	
9	✓ click	css=.mat-datepicker-toggle-default-icon	
10	✓ mouse out	css=.mat-datepicker-toggle-default-icon	
11	✓ click	css=.mat-calendar-body-active	
12	✓ click	css=.mat-select-placeholder	
13	✓ click	css=#mat-option-1 > .mat-option-text	
14	✓ click	id=mat-input-4	
15	✓ type	id=mat-input-4	case
16	✓ click	id=mat-input-3	
17	✓ type	id=mat-input-3	test
18	✓ click	id=mat-input-2	
19	✓ type	id=mat-input-2	password
20	✓ click	id=mat-input-0	
21	✓ type	id=mat-input-0	testcase.com
22	✓ type	id=mat-input-1	password
23	✓ click	css=.mat-raised-button > .mat-button-wrapper	
24	✓ verify element not present	css=h1:nth-child(2)	

Slika 5.9: Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "testcase.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. Očekivani rezultat je ostajanje na register stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.

Extension: (Selenium IDE) - Selenium IDE - system_test — Mozilla Firefox

Project: system_test

Tests

- ✓ login_all_ok
- ✓ login_wrong_email
- ✓ login_wrong_email_and_password
- ✓ login_wrong_password
- ✓ register_already_existing
- ✓ register_new
- ✓ register_wrong_email_format

Search tests...

http://localhost:4200

	Command	Target	Value
1	open	/register	
2	set window size	846x915	
3	click	css=#mat-select-value-3 > .mat-select-placeholder	
4	click	css=#mat-option-5 > .mat-option-text	
5	click	id=mat-input-6	
6	type	id=mat-input-6	123454321
7	click	css=.ng-tns-c132-7 > .mat-form-field-infix	
8	mouse over	css=.mat-datepicker-toggle-default-icon	
9	click	css=.mat-datepicker-toggle-default-icon	
10	mouse out	css=.mat-datepicker-toggle-default-icon	
11	click	css=.mat-calendar-body-active	
12	click	css=.mat-select-placeholder	
13	click	css=#mat-option-1 > .mat-option-text	
14	click	id=mat-input-4	
15	type	id=mat-input-4	case
16	click	id=mat-input-3	
17	type	id=mat-input-3	test
18	click	id=mat-input-2	
19	type	id=mat-input-2	password
20	click	id=mat-input-0	
21	type	id=mat-input-0	test@case.com
22	type	id=mat-input-1	password
23	click	css=.mat-raised-button > .mat-button-wrapper	
24	verify element not present	css=h1:nth-child(2)	

Slika 5.10: Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "test@case.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. User s prikazanim e-mailom je već registriran. Očekivani rezultat je ostajanje na register stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.

Extension: (Selenium IDE) - Selenium IDE - system_test — Mozilla Firefox

Project: system_test

Tests +

Search tests...

- ✓ login_all_ok
- ✓ login_wrong_email
- ✓ login_wrong_email_and_password
- ✓ login_wrong_password
- ✓ register_already_existing
- ✓ register_new
- ✓ register_wrong_email_format

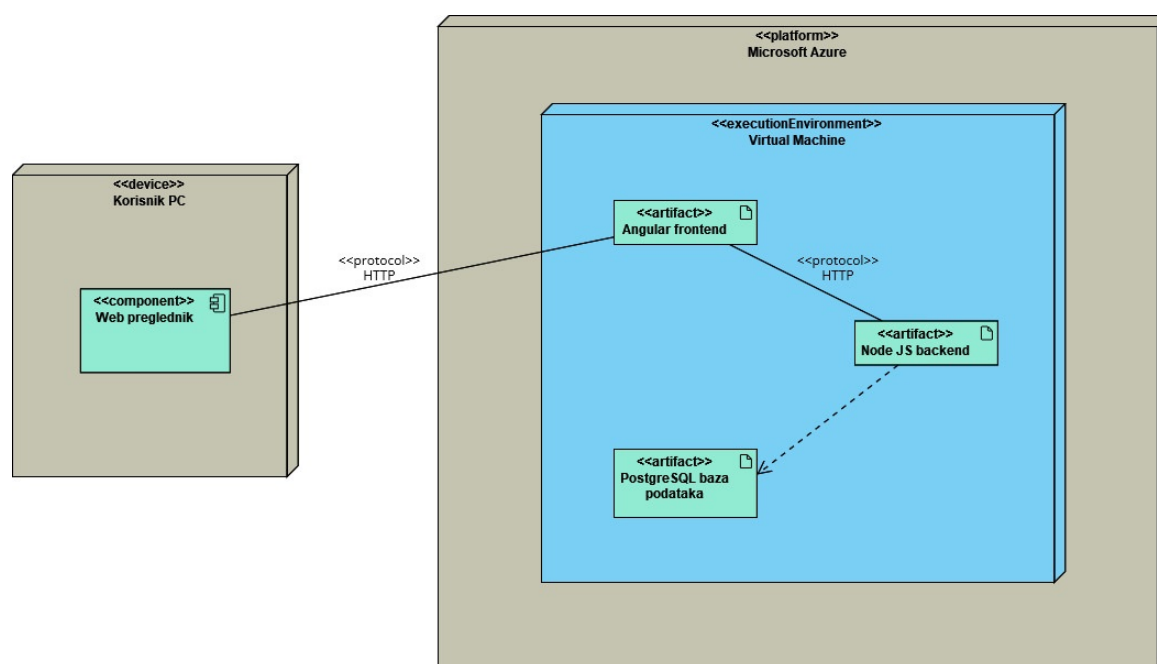
http://localhost:4200

	Command	Target	Value
1	open	/register	
2	set window size	846x915	
3	click	css=#mat-select-value-3 > .mat-select-placeholder	
4	click	css=#mat-option-5 > .mat-option-text	
5	click	id=mat-input-6	
6	type	id=mat-input-6	123454321
7	click	css=.ng-tns-c132-7 > .mat-form-field-infix	
8	mouse over	css=.mat-datepicker-toggle-default-icon	
9	click	css=.mat-datepicker-toggle-default-icon	
10	mouse out	css=.mat-datepicker-toggle-default-icon	
11	click	css=.mat-calendar-body-active	
12	click	css=.mat-select-placeholder	
13	click	css=#mat-option-1 > .mat-option-text	
14	click	id=mat-input-4	
15	type	id=mat-input-4	case
16	click	id=mat-input-3	
17	type	id=mat-input-3	test
18	click	id=mat-input-2	
19	type	id=mat-input-2	password
20	click	id=mat-input-0	
21	type	id=mat-input-0	test@case.com
22	type	id=mat-input-1	password
23	click	css=.mat-raised-button > .mat-button-wrapper	
24	verify element present	css=h1:nth-child(2)	

Slika 5.11: Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "test@case.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. Očekivani rezultat je preusmjerenje na home stranicu, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.

5.3 Dijagram razmještaja

Na slici 5.12 prikazan je dijagram razmještaja. Sustav je baziran na arhitekturi "klijent-poslužitelj". Korisnici pristupaju aplikaciji korištenjem web preglednika. Na platformi Microsoft Azure u VM-u (virtualnom stroju) se nalaze poslužitelji za frontend, backend i bazu podataka. Komunikacija između korisnika i poslužitelja za frontend, poslužitelja za frontend i poslužitelja za backend, ostvaruje se korištenjem protokola HTTP.



Slika 5.12: Dijagram razmještanja

5.4 Upute za puštanje u pogon

Puštanje u pogon sastoji se od nekoliko koraka:

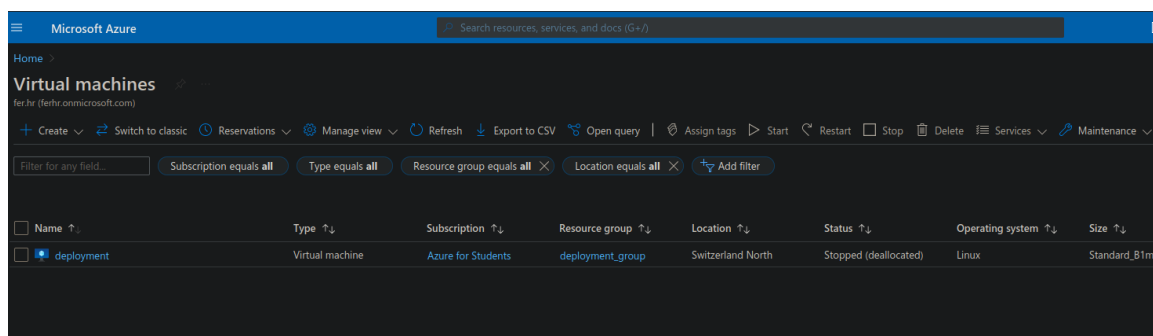
- *Izrada koričkog računa na Microsoftovom Azure servisu*
- *Zauzimanje nove virtualne mašine*
- *Logiranje u novu virtualnu mašinu*
- *Instaliranje potrebnih knjižnica i programa te downloadanje programa*
- *Punjenje Postgres baze podataka*
- *Pokretanje aplikacije s pripadajućom skriptom*

5.4.1 Izrada koričkog računa na Microsoftovom Azure servisu

Potrebno je posjetiti stranicu azure.com te napraviti korisnički račun.

5.4.2 Zauzimanje nove virtualne mašine

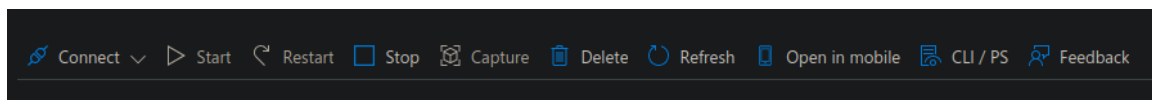
Potrebno je na svojem home pageu na Azureovom servisu zauzeti novu virtualnu mašinu. Za to je potrebno pritisnuti na "Virtual Machine" te pratiti potrebne korake koji automatski dolaze jedan iza drugoga. Također je potrebno dopustiti konekciju na port 80.



Slika 5.13: Prikaz stanja stranice kada se odabere kreiranje virtualne mašine.

5.4.3 Logiranje u novu virtualnu mašinu

Na home pageu potrebno je odabrati virtualnu mašinu te stisnuti na "Start" te zatim "Connect" i izabrati ssh. Zatim pratiti upute koje Azure navodi.



Slika 5.14: Prikaz opcija koje su vidljive na vrhu ekrana nakon što je odabrana virtualna mašina. Potrebno je stisnuti "Start", a zatim i "Connect", odabrati SSH i pratiti upute.

5.4.4 Instaliranje potrebnih knjižnica i programa te downloadanje programa

Potrebno je klonirati program backenda i frontenda na virtualnu mašinu, za što treba imati ovlasti za to kloniranje. To napravimo tako da na virtualnu mašinu stavimo ssh ključ za gitlab. Nadalje trebamo instalirati sve potrebno za rad kloniranoga repozitorija. To postizemo s "sudo apt install nodejs" za nodejs, "npm install -g @angular/cli" za angular, "sudo apt install postgresql postgresql-contrib" za Postgres. Također da smo sigurni da je Postgres upaljen treba provesti još "sudo systemctl start postgresql.service". Ostali dijelovi se dobivaju preko "npm install" u frontend i backend direktorijima.

5.4.5 Punjenje Postgres baze podataka

Potrebno je dodati u bazu podataka tablice koje se dalje koriste u backendu. Pozicionirati se u direktorij backenda ("cd sustav-narucivanja/sustav-narucivanja-server") te zatim "node db/seed.js"

```
Table patient created.
Table patient populated with data.
Creating table appointment.
CREATE TABLE appointment (
  id int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  patientid INT,
  doctorid INT,
  nurseid INT,
  time TIMESTAMP WITHOUT TIME ZONE,
  duration INTERVAL,
  created_on TIMESTAMP WITHOUT TIME ZONE,
  changes_from int,
  type TEXT,
  patient_came BOOLEAN,

  FOREIGN KEY (patientid) REFERENCES patient(id),
  FOREIGN KEY (doctorid) REFERENCES doctor(id),
  FOREIGN KEY (nurseid) REFERENCES nurse(id),
  FOREIGN KEY (changes_from) REFERENCES appointment(id)
)
Table appointment created.
Table appointment populated with data.
Creating indexes
```

Slika 5.15: Očekivani ispis prilikom pokretanja "seed.js". Imati na umu da je ovo samo krajnji dio ispisa, postoji i dio prije ovoga koji je izbačen iz slike u svrhu sažetosti.

5.4.6 Pokretanje aplikacije

Potrebno je pokreniti restart skriptu koja se nalazi u root dijelu. Za to joj je potrebno dati ovlasti za pokretanje i onda je pokrenuti "sudo ./restart.sh"

6. Zaključak

Zadatak naše grupe bila je implementacija sustava za naručivanje pregleda u hrvatskom zdravstvu. Kroz našu aplikaciju pacijenti mogu zakazivati termine ovisno o dostupnosti pojedinih doktora/medicinskih radnika. Nakon 14 tjedana rada, naša grupa je ostvarila upotrebljivu aplikaciju koja ispunjava sve korisničke zahtjeve koji su bili zadani za nju.

Prvi ciklus izrade projekta zahtijevao je okupljanje tima. Nakon okupljanja, prva podjela tima bila je na frontend i backend domenu. Darijan Gudelj, Branimir Tomelj i Vilim Ivanković preuzeli su backend domenu, a Luka Slugečić, Bruno Rački i Marin Teskera preuzeli su frontend domenu. Kako je frontend domena bila ograničena poslom, neki članovi frontend domene pomagali su pri izradi backend domene. Nakon izrade prvostupničke aplikacije koja je funkcionirala u minornim zahtjevima, naša grupa je krenula na izradu dokumentacije. Izradili smo sekvencijske dijagrame, obrasce upotrebe, dijagram razreda i model baze podataka. Svi ti modeli olakšali su razumijevanje i izradu novih komponenti u drugom ciklusu rada. Sastanci su se održavali barem jednom na tjednoj bazi, te su članovi NULL grupe ulagali jako puno samostalnog truda i rada između sastanaka.

Drugi ciklus izrade projekta zahtijevao je da se domene razmijenjuju, tako se većina timova prebacivala po potrebi u drugu domenu. Kako je posla bilo previše, NULL grupa sastajala se dva puta tjedno kako bi uspješno izvršila sve korisničke zahtjeve. U drugom ciklusu radili smo na implementaciji aplikacije koja posjeduje autentifikaciju, nema nepredvidive greške i točno su određene mogućnosti pojedinog korisnika u aplikaciji. Implementirali smo izvješća, SMS poruke, emailove, kalendar itd.. Kako je posao rastao i postajao sve kompleksnije NULL grupa komunicirala je dvjema kanalima, a to su Discord i Whatsapp koji su nam uvelike pomogli u rješavanju problema. Svakodnevno smo komunicirali i izmijenjivali iskustva u izradi ovoga sustava i to sa vjerodostojnošću radili. Nakon završene aplikacije izradili smo i dijagram aktivnosti, stanja i komponenti.

Mogućnost izrade ovog projekta svakome članu NULL grupe dala je nova programska i komunikativna iskustva koje nikada nećemo zaboraviti. Svaki član osjetio je koliko je bitno znati raditi u timu i koliko je svačiji rad blagonosan. Kao tim

smo se uspjeli organizirati, slušali smo jedni druge i ono što je najbitnije, učili jedni od drugih.

Svaki član je zadovoljan postignutim i truditi ćemo se u budućnosti koristiti znanje dobiveno na ovome projektu.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <https://www.fer.unizg.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Početna stranica Reservatic.com	6
2.2	Registracija u Reservatic.com	7
2.3	Razlika između Reservatic.com i Null grupe	7
3.1	Dijagram obrasca uporabe, funkcionalnost korisnika i pacijenta . . .	21
3.2	Dijagram obrasca uporabe, funkcionalnost liječnika i medicinskog tehničara/sestre	22
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	23
3.4	Sekvencijski dijagram za UC4	25
3.5	Sekvencijski dijagram za UC6	26
3.6	Sekvencijski dijagram za UC11	27
4.1	Struktura implementacije aplikacije iz VSC.	30
4.2	ER dijagram baze podataka	34
4.3	Relacijski dijagram baze podataka	34
4.4	Dijagram razreda - Modeli	36
4.5	Dijagram razreda - frontend	37
5.1	Prikazani testovi testiraju dohvaćanje usera iz baze podataka te nje- govo ubacivanje i izbacivanje iz baze podataka. Navedene funkci- onalnosti su nužne za pravilnu registraciju i login korisnika aplikacije.	42
5.2	Izlaz programa za testiranje klase UserModel s 2 testna primjera. Sa slike je vidljivo da klasa prolazi testne primjere.	43
5.3	Prikazani testovi testiraju dohvaćanje određenog termina iz baze po- dataka, testiranje postoji li određeni termin koji je u vremenskom konfliktu s odabranim terminom te testiranje updateanja termina. Testiranje konflikata ima više rubnih slučajeva, naime isti termin mogli smo već dohvatiti iz baze podataka te tada nebi trebalo biti konflikata, zatim moguće je da se termin nalazi unutar drugoga, pot- puno izvan...	44

- 5.4 Izlaz programa za testiranje klase AppointmentModel s 6 testnih primjera. Sa slike je vidljivo da klasa prolazi testne primjere. 45
- 5.5 Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je sve ispravno naravljeno. Korišteni e-mail je "test@case.com" te lozinka "password". Očekivani rezultat je preusmjeravanje na home page, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat. 46
- 5.6 Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriv e-mail. Korišteni e-mail je "test@notcase.com" te lozinka "password". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat. 46
- 5.7 Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriva lozinka. Korišteni e-mail je "test@case.com" te lozinka "password2". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat. 47
- 5.8 Prikazani su koraci testiranja gdje se testira logiranje u aplikaciju ukoliko je kriva lozinka i e-mail. Korišteni e-mail je "test@notcase.com" te lozinka "password2". Očekivani rezultat je ostajanje na login stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat. 47
- 5.9 Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "testcase.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. Očekivani rezultat je ostajanje na register stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat. 48

5.10 Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "test@case.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. User s prikazanim e-mailom je već registriran. Očekivani rezultat je ostajanje na register stranici, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.	49
5.11 Prikazani su koraci testiranja gdje se testira registracija korisnika gdje je sve ispravno napravljeno. Korišteni e-mail je "test@case.com", lozinka i ponovljena lozinka "password", ime "test", prezime "case", broj mobitela "123456789" te datum rođenja izabran putem date selectora. Očekivani rezultat je preusmjeravanje na home stranicu, što se prepoznaje i verificira putem elemenata stranice. Sa slike je vidljivo da se postiže očekivai rezultat.	50
5.12 Dijagram razmještanja	51
5.13 Prikaz stanja stranice kada se odabere kreiranje virtualne mašine. .	52
5.14 Prikaz opcija koje su vidljive na vrhu ekrana nakon što je odabrana virtualna mašina. Potrebno je stisnuti "Start", a zatim i "Connect", odabrati SSH i pratiti upute.	53
5.15 Očekivani ispis prilikom pokretanja "seed.js". Imati na umu da je ovo samo krajnji dio ispisa, postoji i dio prije ovoga koji je izbačen iz slike u svrhu sažetosti.	54

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak - cijeli tim

- Datum: 26. listopada 2022.
- Prisustvovali: D.Gudelj, B.Tomeljak, V.Ivanković, M.Teskera, L.Slugečić
- Teme sastanka:
 - napravljen gitlab repozitorij
 - dogovor oko tehnologija koje ćemo koristiti(Angular i Express)
 - podjela članova na backend: D.Gudelj, V.Ivanković, B.Tomeljak
 - podjela članova na frontend: M.Teskera, L.Slugečić, B.Rački
 - komentiranje i ispisivanje funkcionalnih zahtjeva
 - dodjela zadataka za sljedeći tjedan
 - napisana pitanja za konzultacije

2. sastanak - frontend dio

- Datum: 29. listopada 2022.
- Prisustvovali: M.Teskera, L.Slugečić, B.Rački
- Teme sastanka:
 - M.Teskera prezentira strukturu Angular projekta, komponente i glavne značajke
 - dogovor o daljnjem radu
 - implementacija kalendara

3. sastanak - cijeli tim

- Datum: 4. studenoga 2022.
- Prisustvovali: D.Gudelj, B.Tomeljak, V.Ivanković, B.Rački, L.Slugečić
- Teme sastanka:
 - prezentacija B.Tomeljaka o LaTeX uređivaču - Overleafu i napravljenom predlošku
 - frontend tim predstavlja do sada napravljeno - naslovnu stranicu, login i registraciju
 - D.Gudelj prezentira UML dijagram

- V.Ivanković prezentira konceptualni dijagram baze podataka
- dodjela zadataka za sljedeći tjedan
- napisana pitanja za konzultacije

4. sastanak - backend dio

- Datum: 4. studenoga 2022.
- Prisustvovali: D.Gudelj, B.Tomeljak, V.Ivanković
- Teme sastanka:
 - rješavanje problema oko generiranja skripte za bazu podataka
 - hosting baze podataka
 - dogovor oko polja u bazi podataka

5. sastanak - cijeli tim

- Datum: 8. studenoga 2022.
- Prisustvovali: D.Gudelj, B.Tomeljak, V.Ivanković, B.Rački, M.Teskera
- Teme sastanka:
 - M.Teskera prezentira napredak frontend dijela, većina je gotova, čeka se backend dio
 - D.Gudelj prezentira integraciju baze podataka, dio preuzet s Web1 i slaže se u projekt
 - baza podataka napravljena kao predložak za V.Ivankovića, koji nastavlja raditi na bazi
 - B.Tomeljak predstavlja svoj dio backenda - forme, login i password sigurno se čuvaju, potrebno je spojiti s ostalim dijelom backenda
 - dodjela zadataka za sljedeći tjedan - dio funkcionalnosti treba biti gotov zbog roka

6. sastanak - cijeli tim

- Datum: 12. studenoga 2022.
- Prisustvovali: D.Gudelj, B.Tomeljak, V.Ivanković, M.Teskera, L.Slugečić
- Teme sastanka:
 - dogovorena standardizacija imena prema onima u bazi podataka
 - M.Teskera implementira funkcije napravljene u backendu
 - L.Slugečić, B.Rački i V.Ivanković rade na dokumentaciji
 - git- spajanje grana backend tima na glavnu granu, izbrisane grane koje nisu potrebne
 - B.Tomeljak i D.Gudelj završavaju svoj backend dio

7. sastanak - članovi tima zaduženi za dokumentaciju

- Datum: 15. studenoga 2022.
- Prisustvovali: L. Slugečić, B. Rački, V. Ivanković
- Teme sastanka:
 - postignut dogovor oko sastava dokumentacije
 - razmijenjeni savjeti oko dodataka u pojedine sekcije dokumentaciji
 - rekonstrukcija dokumenta prema savjetima kolega

8. sastanak - svi

- Datum: 16. studenoga 2022.
- Prisustvovali: L. Slugečić, B. Rački, V. Ivanković, M. Teskera, D. Gudelj, B. Tomeljak
- Teme sastanka:
 - rješavanje problema oko hostanja aplikacije i baze podataka
 - prezentacija napravljene dokumentacije
 - dogovoreni zadaci za sljedeći sastanak
 - napisana pitanja za konzultacije

9. sastanak - svi

- Datum: 16. studenoga 2022.
- Prisustvovali: L. Slugečić, B. Rački, D. Gudelj
- Teme sastanka:
 - korekcije na dokumentaciji
 - zadani zadaci M. Teskeri i D. Gudelju

Tablica aktivnosti

	Darijan Gudelij	Marin Teskera	Luka Slugečić	Vilim Ivanković	Bruno Rački	Branimir Tomeljak	
Upravljanje projektom	6						
Opis projektnog zadatka					6		
Funkcionalni zahtjevi			1				
Opis pojedinih obrazaca	2		6				
Dijagram obrazaca	2		3				
Sekvencijski dijagrami			4				
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava	3			5	3	3	
Baza podataka	3			6		2	
Dijagram razreda			3	3	3		
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati	17	18	15	16	15	19	
Ispitivanje programskog rješenja	1	1	1	1	1	4	
Dijagram razmještaja							
Upute za puštanje u pogon	5						
Dnevnik sastajanja	1	3	3	4	2	4	
Zaključak i budući rad							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Darijan Gudelj	Marin Teskera	Luka Slugečić	Vilim Ivanković	Bruno Rački	Branimir Tomelj	
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>izrada početne stranice</i>					5		
<i>izrada logina i registracije</i>		10	5				
<i>izrada baze podataka</i>	2			12			
<i>spajanje s bazom podataka</i>	5			2		2	
<i>back end</i>	7			5		12	