

HealthCare

Arhitekturni dizajn softverskog sistema

Julije Kostov 15680

Strahinja Laktović 15691

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Pregled izmena

Datum	Verzija	Opis	Autor
05.12.2018.	1.0	Inicijalna verzija	JS

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Sadržaj

1. HealthCare sistem	4
2. Funkcionalni zahtevi	4
3. Nefunkcionalni zahtevi i ograničenja	6
4. Arhitekturni zahtevi	6
5. Arhitektura aplikacije	7
5.1 Arhitekturni obrasci	7
5.2 Logički pogled	8
5.3 Fizički pogled	9
5.4 Procesi	9
5.5 Deployment	11
6. Korišćene tehnologije	11

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

HealthCare sistem

Healthcare je sistem koji služi kao sredstvo doktora i pacijenata u zdravstvu kao i za njihovu međusobnu komunikaciju u realnom vremenu. Sistem treba da omogući lakšu i bržu komunikaciju između doktora i pacijenta kao i to da pacijentima pruži pomoć pri lečenju bolesti.

Funkcionalni zahtevi

1. Registracija korisnika
2. Logovanje korisnika

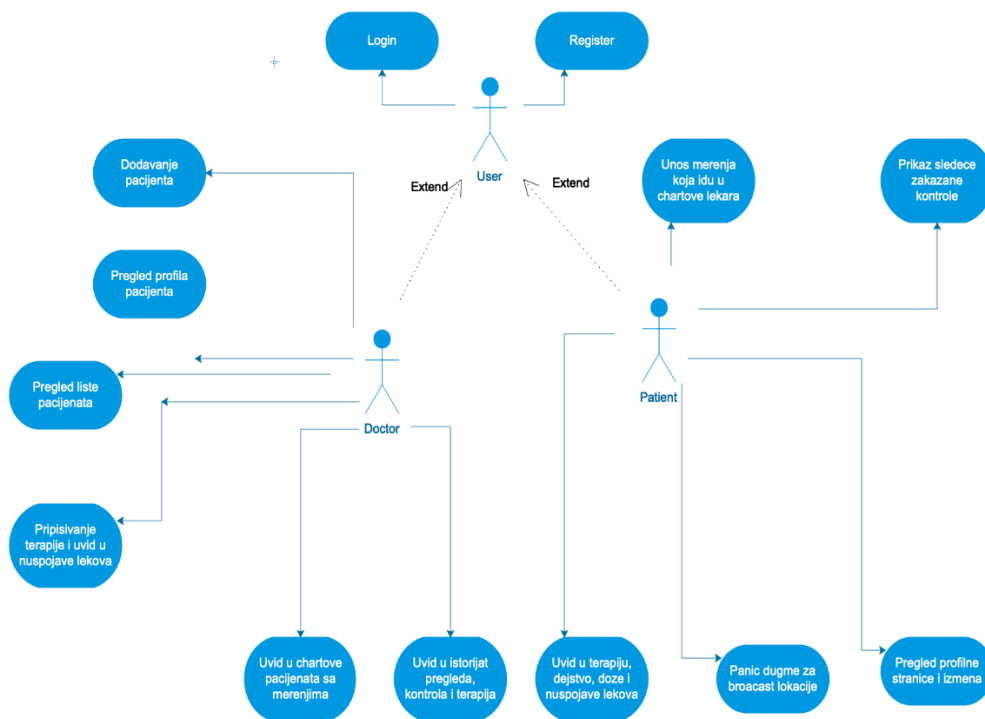
Funkcionalnosti vezane direktno za lekare:

1. Dodavanje novih pacijenata u listu pacijenata preko lbo-a, čime je on izabran kao njihov primarni lekar.
2. Pregled liste pacijenata kao i pregled profila svakog pacijenta pojedinačno.
3. Otvaranje profila pojedinačnog pacijenta gde može videti njegove osnovne podatke kao i podatke od značaja za dijagnozu i lečenje pacijenta.
4. Uvid u istorijat pregleda, zakazivanih kontrola, kao i prepisanih terapija u svakom terminu ponaosob.
5. Navigacija na stranicu sa chartovim-a vezanim za pacijenta, gde doktor može videti grafički sve promene odredjenih parametara koje pacijent unosi kroz vreme, kao što su krvni pritisak, temperatura i drugi, na osnovu čega on jasno i brzo može uočiti trendove rasta ili pada na graficima, čime može zaključiti da li je trenutna terapija efikasna, da li su potrebne neke izmene i slično.
6. Prilikom prepisivanja terapije lekar lekove prepisuje iz baze podataka. Tada može videti i razne kontraindikacije i nuspojave lekova, alergenske komponente i slično.

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Funkcionalnosti vezane direktno za pacijente:

1. Pacijent može da otvori svoju profilnu stranicu u aplikaciji, gde može inicijalno da doda svoje osnovne podatke kao što je krvna grupa, alergije, visina, težina, starost. Ta stranica će se videti i u offline mod-u da ne bi bilo ograničenja u pristupu tim informacijama.
2. Pacijent će moći da otvori stranu sa svojom trenutnom terapijom gde može kliknuti na dugme gde će videti jasno dejstva raznih lekova, doze, kontraindikacije i slično.
3. Prikaz sledeće zakazane kontrole.
4. Stranica gde će pacijent moći da unese informacije o najnovijim merenjima osnovnih parametara čime će se u realnom vremenu te informacije dodati u chartove izabranog lekara.
5. Panic dugme, dugme u slučaju nužde čijim pritiskom će pacijent moći da broadcastuje svoju trenutnu lokaciju, a drugi ljudi u odredjenom radijusu će dobiti obaveštenje kako bi pomogli.



HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Nefunkcionalni zahtevi i ograničenja

Prilikom projektovanja Healthcare aplikacije, ustanovljena je potreba za sledećim nefunkcionalnim zahtevima:

- Pouzdanost (reliability)
- Lakoća korišćenja (usability)
- Performanse (performance)
- Otpornost na greške (fault tolerance)
- Podrška za različite operativne sisteme (cross-platform)
- Lako održavanje i proširljivost (maintainability)

Arhitekturni zahtevi

Na osnovu navedenih nefunkcionalnih zahteva (odnosno atributa kvaliteta) i ograničenja, imamo sledeće arhitekturne zahteve:

- Pouzdanost - Arhitektura aplikacije mora da bude takva da omogući lako dodavanje novih funkcionalnosti na bazi trenutne aplikacije. Sistem treba da funkcioniše u zdravstvu i kao takav mora imati pouzdane podatke. Aplikacija treba da bude dostupna 24/7.
- Lakoća korišćenja - Aplikacija treba da ima jednostavan i intuitivan korisnički interfejs.
- Performanse - Aplikacija treba da podrži real-time komunikaciju i omogući brzu obradu korisničkih zahteva, pogotovo u vezi sa Panic button-om.
- Otpornost na greške - Sistem radi sa podacima koji moraju biti tačni, pogotovo u slučaju prepisivanja lekova, dakle otpornost na greške je jedan od najbitnijih zahteva.

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

- Podrška za različite operativne sisteme - Sistem treba da ima široku primenu. Web aplikacija u okviru sistema radiće u svim aktuelnim Web pretraživačima i nativna mobilna aplikacija za Android i IOS operativne sisteme.
- Lako održavanje - Aplikacija treba da bude projektovana tako da može biti lako održavana.

Arhitektura aplikacije

Ovaj odeljak prikazuje logičku arhitekturu sistema i sadrži opis najznačajnijih klasa, njihovu organizaciju u pakete i podsisteme, i organizaciju podsistema u slojeve.

Arhitekturni obrasci

Layered pattern

Na najvišem nivou primenjen je slojeviti (layered) arhitekturni obrazac.

Presentation layer - Mobilna React Native i Web React aplikacija, pružaju interfejs pacijentu odnosno doktoru.

Service layer - Sloj koji "obmotava" aplikaciju i pruža njene funkcionalnosti u vidu API-ja pomoću kojeg aplikacije na presentation layer-u mogu da komuniciraju sa ostatkom sistema pomoću HTTP zahteva.

Business logic layer - Čine ga interfejsi repozitorijuma i servisa u NEST serveru. Predstavljaju jezgro aplikacije nezavisno od prezentacionog ili persistence sloja.

Data Access Layer - Čine ga implementacije repozitorijuma, karakteristične za TypeORM.

Persistence Layer - Sloj gde su podaci pohranjeni. U Healthcare sistemu koriste se PostgreSQL i Redis baza.

Microservice pattern

U domenu backend servisa, iskorišćen je microservice pattern omogućavajući podelu zaduženja medju servisima i delegiranje odgovornosti.

Redis mikroservis čija je dužnost da komunicira sa Redis bazom i pribavlja lekove, ili ukoliko

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

tamo nisu pronađeni API pozivom na Remote 3rd-party API server pribavlja i reorganizuje podatke u Redis bazi. Time on služi kao neka vrsta proxy-ja između NEST servisa i time kada se zatraži lek prilikom kreiranja dijagnoze simulira neku vrstu "keš memorije" za njih.

NEST mikroservis pruža API koji je zadužen za pružanje CRUD operacija nad modelima podataka i perzistencijom podataka u bazu kao i izvršenja biznis logike.

Mikroservisi NEST i Redis komuniciraju međusobno putem molecular brokera.

Publish-subscribe

Za realtime event based komunikaciju koristiće se Pusher komponenta. Klijenti se mogu pretplatiti na određeni događaj. Server obradom zahteva prosleđuje Pusher komponenti da se desio određeni događaj koji se dalje prosleđuje klijentima koji su pretplaćeni na taj događaj.

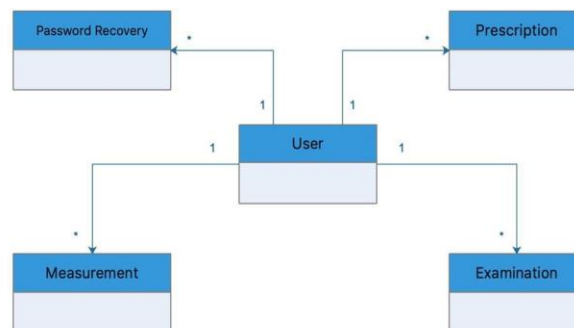
Pored toga sam molecular koji se koristi za komunikaciju između mikroservisa koristi publish subscribe metod kako bi se jedan event emitovan u određenom mikroservisu obradio u drugim mikroservisima koji su pretplaćeni na taj event, a kao transporter poruka koristi se Redis baza podataka.

MVC (Model-View-Controller)

NEST API mikroservis implementiran pomoću NEST framework-a koji koristi MVC obrazac tako što je biznis model zapravo model podataka, controller biznis logika koja se izvršava prilikom API poziva i view je klijentska aplikacija koja vrši API pozive.

Logički pogled

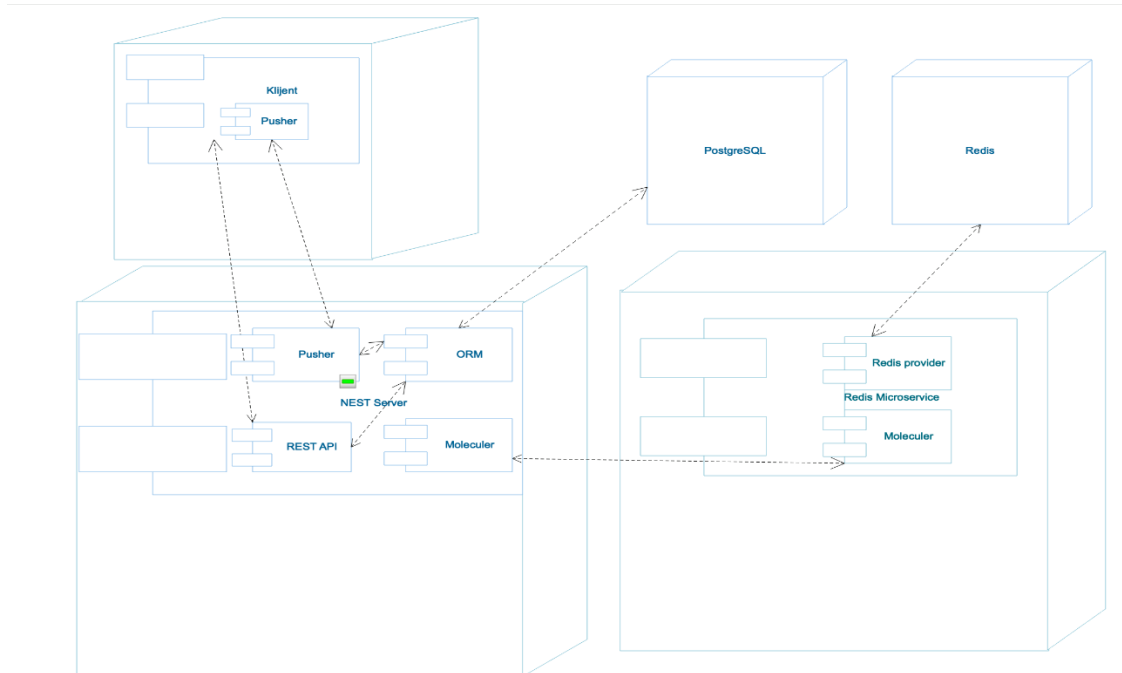
Logički pogled sistema prikazuje klasni dijagram sistema.



HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Fizički pogled

Fizički pogled sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema.



Procesi

PostgreSQL server

PostgreSQL Server je proces koji izvršava funkcionalnost PostgreSQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate.

Healthcare NEST mikroservis

Healthcare NEST mikroservis je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih od više Healthcare klijenata, uključujući i mobilne i Web klijente. Sav rad doktora i merenja i pregledi pacijenata šalju zahteve ovom serveru koji ih obrađuje, delegira i vraća odgovore. Healthcare NEST mikroservis takodje omogućava i real-time komunikaciju korišćenjem Pusher-a, dok sa ostalim mikroservisima komunicira putem moleculer framework-a.

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

Web klijent

Web klijent je proces koji izvršava funkcionalnosti aplikacije za prikaz korisničkog interfejsa doktorima. Pruža interfejs za prikaz postojećih i dodavanje novih pacijenata. Takođe putem njega moguće je pripisati novu terapiju, zakazati kontrole i pregledati chartove pacijenata. Deo biznis logike izvršava se na strani klijenta.

Mobile klijent

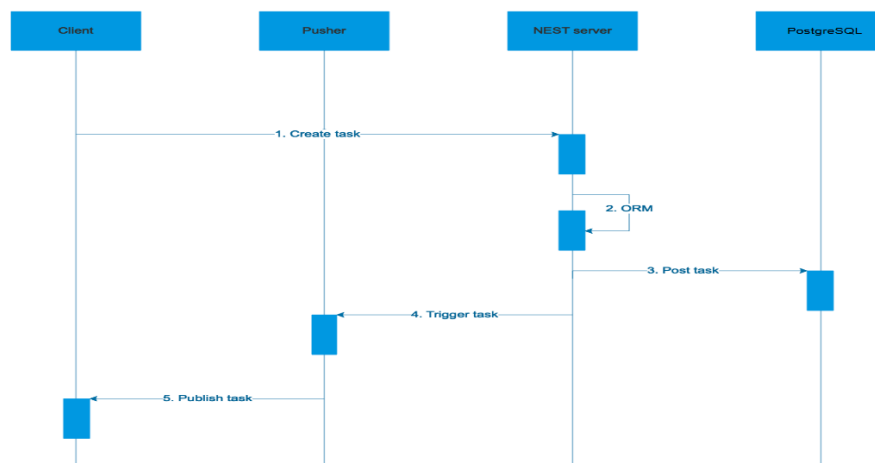
Mobile klijent je proces koji izvršava funkcionalnost aplikacije za prikaz korisničkog interfejsa pacijentima. Pruža mogućnost dodavanja osnovnih podataka, kao i najnovijih merenja osnovnih parametara kao što je krvni pritisak i drugi. Pacijent može videti i svoju trenutnu terapiju sa jasno definisanim dozama, i satnicom koju mora poštovati prilikom uzimanja lekova, kao i termin kontrolnog pregleda. Pacijent može takođe da pritiskom na Panic dugme obavesti ljude u određenom radijusu da je došlo do kritične situacije. Kao i kod web klijenta deo biznis logike prebačen je na strani klijenta.

Redis mikroservis

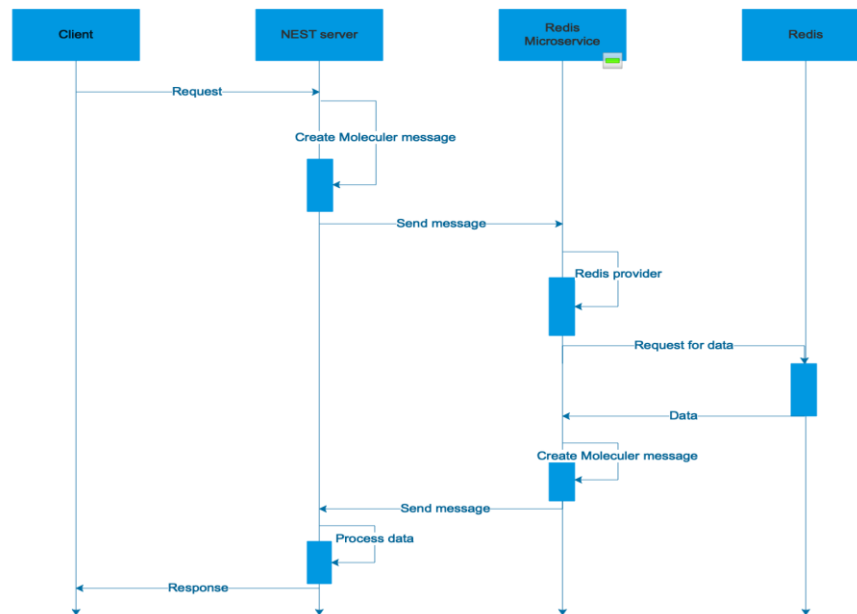
Redis microservice jeste proces koji u pozadini koristi Redis bazu podatka i koji funkcioniše kao keš memorija za najčešće prepisivane lekove. On vraća Healthcare NEST-u zatražen lek u trenutku kada doktor pripisuje terapiju, ili, ukoliko leka nema u Redis bazi, API pozivom pribavlja podatke sa Remote API servisa, i dodaje novi lek u Redis.

Redis server

Redis server jeste proces gde se nalazi baza sa izuzetno brzim vremenom pristupa odakle Redis microservice pribavlja lekove, ili upisuje nove.

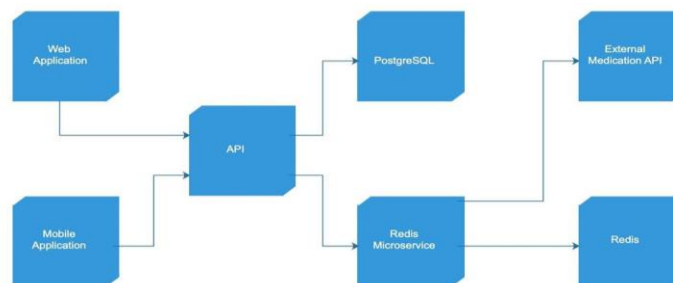


HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine



Deployment

Za process razvoja i deploymenta aplikacije koristiće se Docker.



API predstavlja NEST mikroservis koji se izvršava na jednom docker kontejneru, redis mikroservis se takođe izvršava na jednom kontejneru i oni komuniciraju putem molecular brokera. PostgreSQL i Redis baze podataka se izvršavaju takođe u posebnim kontejnerima. Sa njima komuniciraju NEST mikroservis i redis mikroservis.

Korišćene tehnologije

Za izradu klijenata korišćene su srodne React i React Native tehnologije.

Za izradu NEST mikroservisa korišćen je NEST Typescript framework koji omogućava ozbiljnu kontrolu nad podacima koristeći prednosti Typescripta, ali i fleksibilnost jezika Javascript u koji

HealthCare	Verzija: 1.0
Arhitekturni dizajn softverskog sistema	Datum: 05.12.2018. godine

se kod transpajlira. On koristi TypeORM za komuniciranje sa PostgreSQL bazom i Moleculer broker framework za komuniciranje sa Redis mikroservisom. Redis mikroservis jeste lightweight NodeJS mikroservis povezan sa Redis bazom, NEST serverom putem brokera i eksternim API servisom za lekove putem axios-a.

Za realtime komunikaciju, ažuriranje podataka na klijentskoj strain kao i za notifikacije koristi se Pusher.