

# Proof of Concept

**Internet softverske arhitekture**

Softversko inženjerstvo i informacione tehnologije

Školska 2020./2021. godina

Tim 1

Dimitrije Karanfilović, SW-39/2018

Marko Njegomir, SW-38/2018

Luka Šerbedžija, SW-32/2018

Dušan Erdeljan, SW-43/2018

# 1 UVOD

Zadatak ove aplikacije je da korisnicima olakša proces rezervacije lekova i zakazivanje pregleda kod farmaceuta i dermatologa. Imajući u vidu da aplikacija rukuje privatnim podacima korisnika, potrebno je čuvati integritet i privatnost korisničkih podataka. U ovo spadaju, na primer, istorija poseta farmaceutima i dermatolozima, istorija lekova i naravno lični podaci.

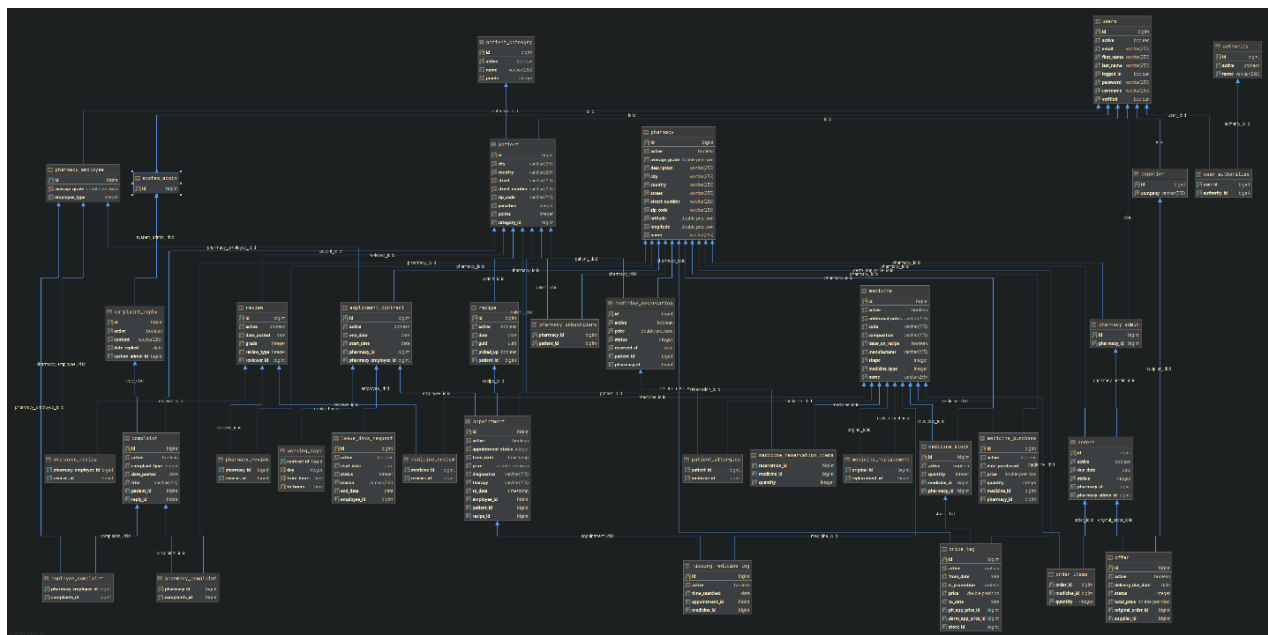
Dva najvažnija entiteta u sistemu su lekovi i pregledi. S toga je od izuzetne važnosti obezbediti konzistentnost količine leka na stanju u određenoj apoteci kao i dostupnost određenog pregleda.

Procena je da će u narednih 5 godina sistem biti korišćen od strane 200 miliona korisnika. Treba imati u vidu da broj od 200 miliona nije isto kao što bi to bilo na Facebook-u. Smatramo da će korisnici pristupiti ovoj aplikaciji kada moraju i da bi izvršili neki zadatak (rezervaciju leka i/ili zakazivanje pregleda). Shodno tome naša procena je da će korisnici koristiti aplikaciju u proseku 3 sata nedeljno. Takođe, smatramo da će se aplikaciji pristupiti najčešće u standardnom radnom vremenu, u zavisnosti od regije u kojoj se aplikacija koristi, najčešće od 9h do 17h.

Kako se korišćenje sistema eventualno završava fizičkim odlaskom na određenu lokaciju, bilo za potrebe preuzimanja leka ili za potrebe odlaska na pregled, smatramo da će korisnici najčešće pristupiti podacima apoteka koje su geografski bliske njihovoj lokaciji. Dominantna operacija u sistemu će biti čitanje, dok će se sesija uglavnom završavati nekom akcijom koja vrši izmenu sistema.

## 2 DIZAJN ŠEME BAZE PODATAKA

U nastavku je dat ER dijagram šeme baze podataka.



Slika 1 ER Dijagram baze podataka

## Predlog strategije za particionisanje podataka

Planirano je da se izvrši particionisanje na 3 načina:

- Horizontalno,
- Vertikalno,
- Funkcionalno.

Planirano je i da se vrši monitoring rada i opterećenja sistema, te da se shodno dobijenim podacima vrši rebalansiranje strategije za particionisanje.

### 2.1 HORIZONTALNO PARTICIONISANJE

Horizontalno particionisanje će se prvo vršiti po geografskoj lokaciji. To se radi zbog toga što su apoteke vezane za određenu lokaciju, a operacije kao što su zakazivanje pregleda i rezervacija lekova zahtevaju posećivanje date lokacije. Shodno tome, arhitekturu sistema bismo podelili na određeni broj data centara koji bi bili raspoređeni u skladu sa saobraćajem određenih regiona. Podaci o korisnicima će takođe biti raspoređeni po lokacijama sa kojih su se registrovali ili sa koje su poslednji put pristupili sistemu. Da bi se ovo postiglo, koristiće se hash funkcija koja će izabrati geografski najbližu particiju. Prilikom čitanja, hash funkcija će odrediti tačnu particiju odakle podaci treba da se čitaju, bez potrebe da se prolazi kroz sve particije.

Dodatan vid horizontalnog particionisanja bi moglo biti i particionisanje podataka kao što su termini pregleda i rezervacije lekova po mesecima. Ovo bi značilo da se prilikom generisanja izveštaja za jedan mesec, svi podaci nalaze u istoj particiji. Dodatno, kako se podaci prikazuju u mesečnom kalendaru, svi podaci koje treba dobiti bi opet bili u istoj particiji. Prilikom generisanja kvartalnih i godišnjih upita moguće bi bilo izvršiti paralelni upit na više particija, a zatim izvršiti agregaciju svih međurezultata u jedan rezultat.

### 2.2 VERTIKALNO PARTICIONISANJE

Što se tiče vertikalnog particionisanja, podelili bismo tabelu medicine stock na dve particije. Prva bi predstavljala read-only podatke kojima se često pristupa i koji bi se mogli keširati, kao što su ime, opis leka, sastav i proizvođač. U drugoj particiji bismo čuvali informacije o količini i ceni leka, jer se ovi podaci jedino i menjaju.

### 2.3 FUNKCIONALNO PARTICIONISANJE

Što se tiče funkcionalnog particionisanja, osetljivi podaci za svakog korisnika poput istorije pregleda, rezervisanih lekova i ličnih informacija se mogu particionisati na particije koje bi imale dodatnu sigurnosnu zaštitu. Takođe će biti replicirana polja sa cenama pregleda i leka u apoteci, da bi se izbegla nepotrebna izračunavanja.

### 3 PREDLOG STRATEGIJE ZA REPLIKACIJU BAZE I OBEZBEĐIVANJE OTPORNOSTI NA GREŠKE

---

Imali bismo konfiguraciju master-slave. Slave bi bile read only baze, gde bi se skladištile informacije kojima se često pristupa radi čitanja. Master baza dozvoljava i pisanje i propagirala bi izmene svim povezanim slave bazama.

Za svaku geografsku lokaciju bi se u periodu kada je najmanji intenzitet saobraćaja pravila replikacija najbitnijih podataka u tom delu sistema.

Povremeno bi se vršilo uklanjanje logički obrisanih podataka. Dodatno, podaci koji su zastareli bi se mogli migrirati u skladišta kojima se retko pristupa, uz nižu cenu skladištenja, kao što je na primer *Amazon S3 Glacier Deep Archive*.

### 4 PREDLOG STRATEGIJE ZA KEŠIRANJE PODATAKA

---

Na svim data centrima bi se keširale informacije o lekovima. Korišćenjem monitoring sistema bi se odredile apoteke kojima se najčešće pristupa i na osnovu tih informacija bi se ažurirala keš politika.

Koristili bismo CDN da se kešira statički sadržaj poput slika na početnim stranicama i slika najčešće posećivanih apoteka. Keširali bismo i js fajlove i css fajlove, koji su delovi biblioteka koje smo koristili u projektu, kao na primer Bootstrap i Material Design. Za CDN bismo mogli koristiti *Cloudflare*.

Za privatni keš bi se koristio Hibernate L1 keš.

Za deljeni keš bismo koristili Redis keš.

Lista lekova bi se mogla ubaciti unapred, dok bi se detalji apoteka keširali po zahtevu korišćenjem LRU politike. Dodatnim praćenjem saobraćaja bi se moglo ustanoviti koje apoteke bi se mogli keširati unapred, ukoliko im korisnici često pristupaju.

### 5 OKVIRNA PROCENA ZA HARDVERSKЕ RESURSE POTREBNE ZA SKLADIŠTENJE SVIH PODATAKA U NAREDNIH 5 GODINA

---

#### 5.1 REZERVACIJE LEKOVA

Broj rezervacija lekova na mesečnom nivou je 1000000 (milion). Rezervacije je predstavljena medicine\_reservation tabelom, gde jedan red u tabeli zauzima 53 bajta. Svaka rezervacija je povezana sa redom iz tabele medicine\_reservation\_item koje predstavlja količinu naručenog leka. Jedan red iz te tabele zauzima 37 bajtova. Kombinovano, jedna rezervacija zauzima 90 bajtova prostora. Odatle sledi, da je potreban kapacitet za mesec dana  $90b * 1000000 = 86mb$ , odnosno kroz period od pet godina to izađe  $86mb * 5 * 12 = 5gb$ .

## 5.2 KORISNICI

Broj korisnika je 200 000 000. Korisnik je predstavljen redom u tabeli user i dodatnim povezanim redom u tabeli patient. Red u tabeli user zauzima 1295 bajtova, dok povezani red u tabeli patient zauzima 1554 bajtova. Ovo zbirno daje 2849 bajtova. To znači da nam je za skladištenje pomenutog broja korisnika treba ukupno 530gb. Ako uzmemo da bi broj korisnika bio višestruko veći nego broj administratora i zaposlenih u apoteci, okvirna procena za kapacitet za skladištenje korisnika je 700gb.

## 5.3 PREGLEDI

Broj pregleda na mesečnom nivou je 1000000. Pregled je predstavljen jednim redom u tabeli appointment, koji zauzima 316 bajtova. To znači da nam je za 5 godina potrebno  $316b * 1000000 * 60 = 18gb$ . Ako uzmemo da je prosečan prepisan broj lekova posle pregleda 2, to je dodatnih  $53 + 37 + 37 = 127$  bajtova po pregledu. Kroz period od 5 godina, ovo je dodatnih  $127b * 1000000 * 60 = 7gb$  prostora.

Apoteke, zaposleni u apotekama i administratori se neće skalirati srazmerno broju korisnika. Na primer, za kreiranje porudžbenice, ne očekujemo prevliki porast redova u ovoj tabeli, već samo da se povećava količina leka koji se naručuje.

Imajući sve ovo u vidu, smatramo da bi nam za 5 godina bilo dovoljno, na primer, *Amazon S3 Standard* skladište podataka, od 50tb.

## 6 PREDLOG STRATEGIJE ZA POSTAVLJANJE LOAD BALANSERA

---

Load balancer bi bio ispred data centara, aplikativnih servera u data centru i ispred baza podataka. Load balancer ispred data centara bi bio takav da preusmerava korisnika na data centar koji je korisniku najbliži. Load balancer-i ispred aplikativnih servera i ispred baza podataka bi mogli raditi po round robin principu.

## 7 PREDLOG KOJE OPERACIJE KORISNIKA TREBA NADGLEDATI U CILJU POBOLJŠANJA SISTEMA

---

Nadgledali bismo koje apoteke korisnike najčešće posećuju. Takođe bismo gledali i u kom vremenskom periodu korisnici pristupaju lokaciji na određenoj geografskoj lokaciji.

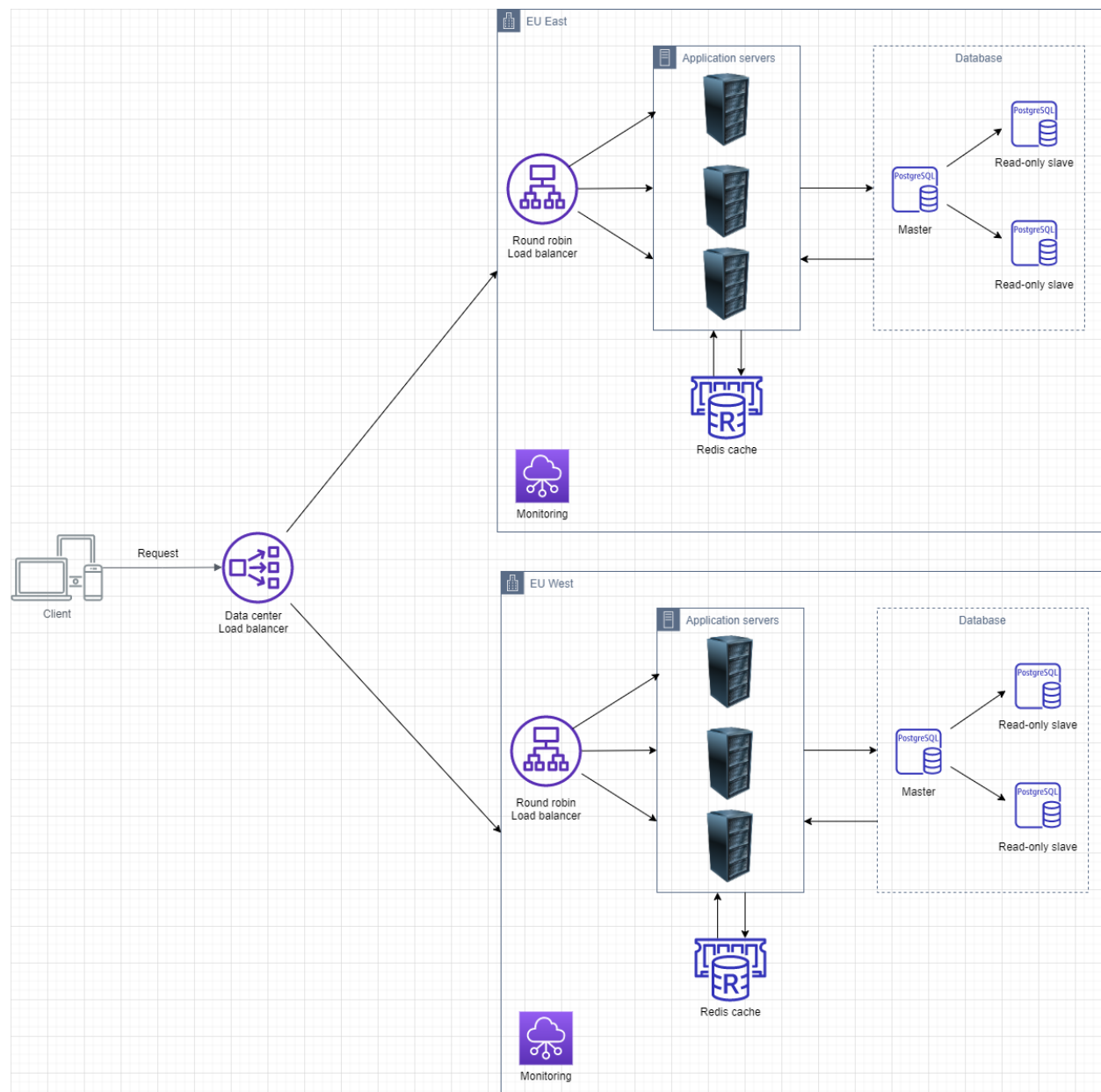
Pratio bi se intenzitet saobraćaja u određenim regionima, pa bi se shodno tome korigovao kapacitet data centara i politike keširanja i particionisanja.

Konkretno, za potrebe nadgledanja, bismo mogli koristiti *Google Analytics*.

## 8 KOMPLETAN CRTEŽ DIZAJNA PREDLOŽENE ARHITEKTURE (APLIKATIVNI SERVERI, SERVERI BAZA, SERVERI ZA KEŠIRANJE, ITD)

U nastavku je dat dizajn predložene arhitekture. Na dijagramu su prikazani data centri samo za EU West i EU East, ali bismo, u skladu sa saobraćajem iz određenih regiona, postavljali i korigovali data centre i na ostalim lokacijama.

Imajući u vidu da smo za praćenje korisničke sesije koristili JWT, a ne sesiju, ne bismo morali da vodimo računa o replikaciji korisničke sesije između servera. Podaci o sesiji bi se čuvali na klijentskoj strani, u okviru http-only kukija.



Slika 2 Dijagram predložene arhitekture