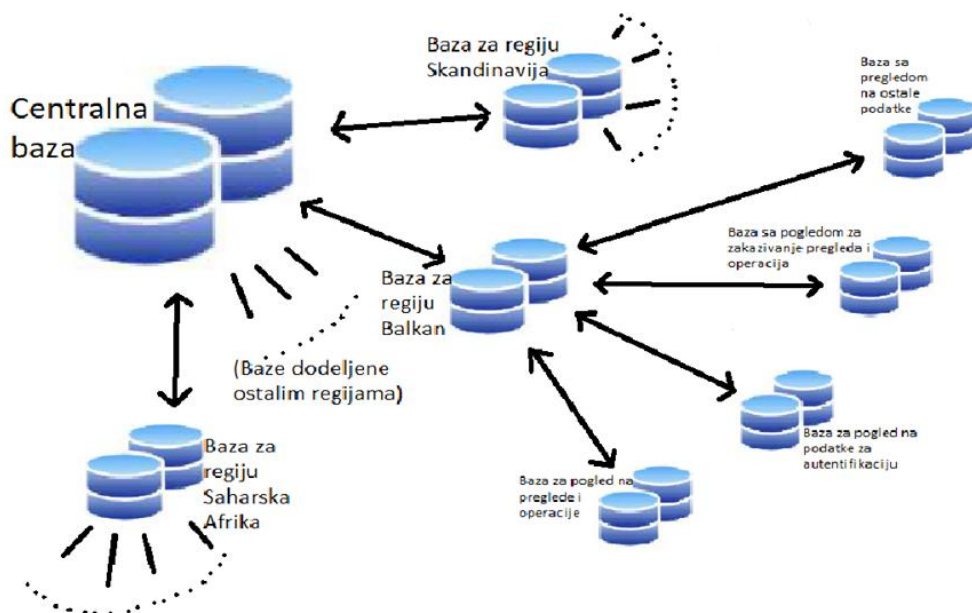
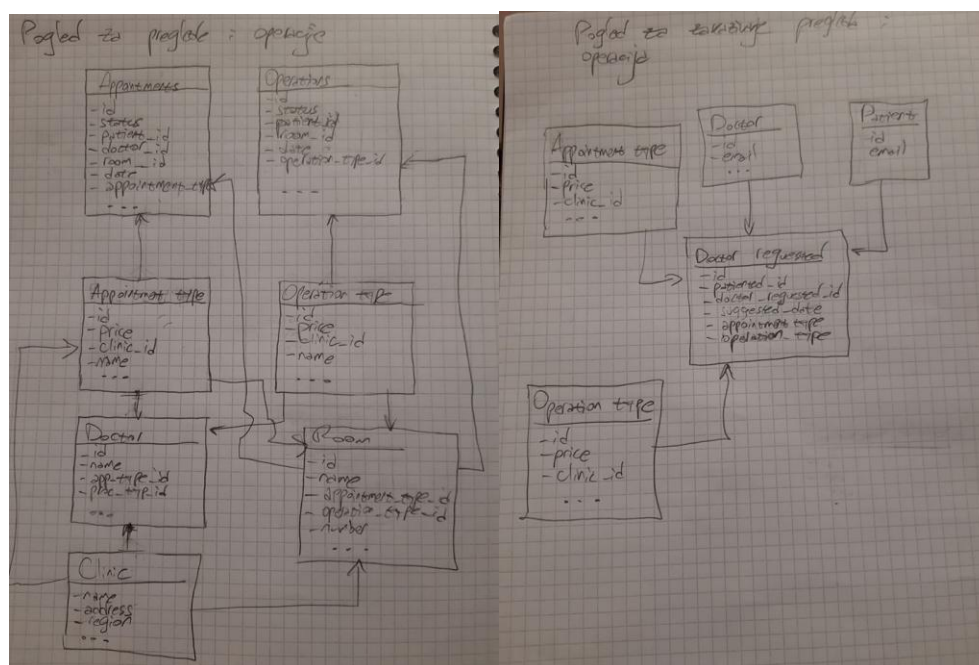


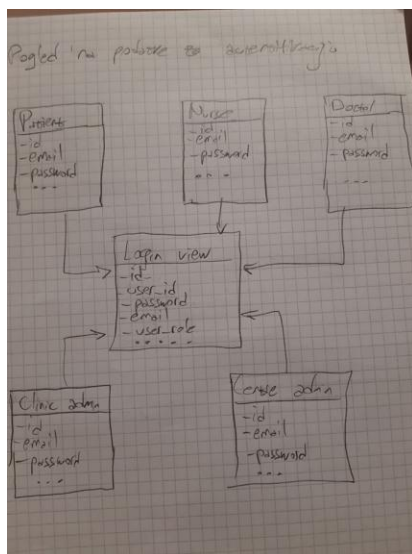
# Skalabilnost – Internet softverske arhitekture – Tim 7

## 1. Dizajn šeme baze podataka



slika 1.1 – Distribucija baza podataka.





slike 1.2, 1.3, 1.4 – Predlog pogleda šeme distribucije baze podataka.

## 2. Predlog strategije za particionisanje podataka:

Uzimajući u obzir da su najprometniji entiteti upravo pregledi i operacije, smisleno je particionisati ih po nekom kriterijumu. Pristupanje pregleda i operacija je daleko ređe ukoliko ti pregledi i operacije nisu vremenski bliski. Shodno tome, predlog za particionisanje podataka, prikazan na šemi, bi veoma ličio na particionisanje po vremenskom opsegu, gde bi se operacije i pregledi particionisali na osnovu datuma kreiranja, po segmentima meseci u kojem su kreirani. Time bi se ravnomerno distribuirali segmenti pregleda i operacija tokom cele godine.

Međutim, najaktuelniji segment će ujedno biti i najopterećeniji u svakom trenutku godine, jer medicinsko osoblje i pacijenti najviše pristupaju najskorijim ili neposredno vremenski bliskim pregledima i operacijama.

Jedna od strategija koja ovakvu prepreku ublažava jeste particionisanje na osnovu regiona u kojem se klinika nalazi.

Uzimajući u obzir činjenicu da su pregledi i operacije nužno vezani za kliniku, dok je klinika nužno vezana za geografsko područje, particionisanje bi se vršilo po regijama u kojima se klinike nalaze. Lista particija pregleda i operacija za npr. Oksford bi u sebi sadržala vremenske opsege od npr 4 godišnja intervala, u kojima bi se skladištili podaci.

Usled ovakve particije pregleda i operacija bi se upiti znatno poboljšali, gde bi kompozitna struktura omogućila ravnomerniju podelu podataka.

### **3. Predlog strategije za replikaciju i otpornosti na greške:**

Postojao bi distribuirani sistem sa redundantnim serverima na kojima bi se skladištili podaci i time obezbedili integritet podataka. Ukoliko deo servera prestane sa radom, drugi bi mogli da ih zamene. Svo pisanje odlazi direktno na primarni, dok sekundarni čitaju sa primarnog i prosleđuju podatke korisnicima.

### **4. Strategije keširanja**

Pošto zaposleni u sistemu mnogo češće pristupaju sajtu od pacijenata, i pošto je za efikasnost kojom vrše svoj posao važno da sajt brzo reaguje, smatramo da je najbolje keširati njihove podatke.

Pod pretpostavkom da lekari i sestre najčešće gledaju svoj radni kalendar, koji sadrži raspored svih pregleda koji ih čekaju, zajedno sa dodatnim informacijama o samim pregledima, bilo bi dobro keširati za medicinsko osoblje spisak svih pregleda koje imaju, npr. barem nedelju dana unapred.

Administratorima bi se mogli na sličan način keširati podaci o zahtevima koje imaju.

Administratori centra svaki dan treba da prate i prihvate ili odbiju zahteve za registraciju novih pacijenata, pa bi se ti zahtevi mogli keširati. Administratori pojedinačnih klinika moraju da vode evidenciju o zahtevima za preglede i operacije, i zahteve za odsustvo lekara i sestara.

Pošto smatramo da će zahteva za odsustvo biti za red veličine manje nego zahteva za preglede i operacije, nužno bi bilo samo keširanje zahteva za preglede i operacije. Podatke za pacijente ne smatramo nužnim za keširanje, pošto mnogo manje opterećuju hardverske resurse aplikacije.

### **5. Procena hardverskih resursa za skladištenje 5 godina podataka**

#### **Prepostavke:**

- Ukupan broj korisnika je 200 000 000.
- Broj zakazanih pregleda ili operacija na mesečnom nivou je milion.

Recimo da je 20% zaposleno i koristi aplikaciju u proseku 10 puta na dan, radnim danima, od toga 4% administratori, i po 8% lekara i sestara.

80% su pacijenti, koji koriste aplikaciju u proseku jednom nedeljno.

Dakle,  $40.000.000 * 50 + 160.000.000 * 1 = 2.160.000.000$  nedeljnih korisnika, tj  $\sim 310.000.000$  dnevnih korisnika.

Pretpostavimo da sistemom na mesečnom nivou bude zakazano 1.000.000 pregleda i operacija.

Objekat pregleda u bazi:  $8 * \text{long} + 1 \text{ boolean} + 1 \text{ varchar}(255) + \text{timestamp} + \text{float8}$ .

Imamo  $4B * 8 + 1B + 4B * 255 + 8B + 8B = 1069$  bajta ili  $\sim 1.1\text{KB}$  po pregledu.

Objekat operacije u bazi:  $8 * \text{long} + 1 \text{ boolean} + \text{varchar}(255)*4B + \text{timestamp}$ .

Dakle ukupno  $4B * 8 + 1B + 4B * 255 + 8B + 5*4B = 1081B \sim 1.1\text{KB}$  po operaciji.

Recimo da aproksimiramo veličinu operacije na veličinu pregleda.

To znači da imamo  $1100 * 1.000.000 = 1.100.000.000$  bajta pregleda mesečno, tj  $\sim 1049 \text{ MB}$  tj  $\sim 1.1 \text{ GB}$  novih podataka dnevno, na osnovu pregleda i operacija.

Dalje, pretpostavimo da lekari i sestre u proseku 4 puta godišnje odsustvuju sa posla,  $4 * 0.16 * 200.000.000 = 128.000.000$  zahteva za odsustvo godisnje, ili  $\sim 350.000$  zahteva dnevno.

Zahtev za odsustvo je veličine:  $3 * \text{long} + 2 * \text{timestamp} + \text{varchar}(255)*4B$ , tj.  $3 * 8B + 2 * 8B + 255 * 4B = 1060B \sim 1.1\text{KB}$ .

$350.000 * 1100 = 104.000 \text{ B} / \text{dan}$ , tj  $\sim 368 \text{ MB}$  dnevno, na osnovu zahteva za odsustvo.

Pošto je priliv ostalih podataka manji od priliva novih operacija i pregleda, i pošto smo prilikom njihovog računanja u skoro svakom koraku zaokrugljivanja zaokrugljivali na više, pretpostavimo da je dnevni priliv novih podataka  $\leq 1.5 \text{ GB}$ .

U kombinacij sa podacima potrebnim da vodi evidenciju o 200.000.000 korisnika, i pod pretpostavkom da je klinika u sistemu prosečno jedna na 100 zaposlenih, tj  $40.000.000 / 100 = 400.000$ , i da ovi podaci zajedno čine glavninu naših podataka, za korisnike je potrebno  $16 * \text{varchar}(255)*4 + 2 * \text{boolean} + 3 * \text{long} + 1 * \text{timestamp}$  prostora u bazi, tj.  $16 * 255B*4 + 2 * 1B + 3 * 8B + 1 * 8B = 16 \text{ KB}$  podataka po korisniku, ako ih sve aproksimiramo kao lekare, koji su najveći pojedinačni korisnik, što za sve korisnike iznosi  $200.000.000 * 16.000 = 822.800.000.000b \sim 3 \text{ TB}$ .

Dalje, memorijski prostor potreban da skladišti klinike je  $1 * 8 + 5 * \text{varchar}(255)$ , što je  $8B + 5 * 4B*255 = 5\text{KB}$  podataka po klinici, tj.  $5.000 * 400.000 = 2.000.000.000B \sim 1.9 \text{ GB}$  za sve klinike na nivou sistema, ali recimo pesimistično da je  $2 \text{ GB}$ .

Dakle, glavnina trajnih podataka zauzela bi do  $3.02 \text{ TB}$  podataka, dnevno bismo dodavali jos po  $1.1\text{GB}$  zbog pregleda i operacija, i za 5 godina bi nam trebalo  $3.02\text{TB} + 365 * 1.1\text{GB} * 5 \approx 3.4\text{TB}$ , recimo da zaokruglimo na pesimističnih  $4\text{TB}$ , i aproksimirali smo koliko bi nam trajnog memorijskog prostora bilo potrebno.

Iz tačke 4, pod pretpostavkom da je neophodan minimalan kapacitet keš memorije za skladištenje pregleda, operacija, podataka o zahtevima upućenim ka administratorima, barem nedelju dana unapred, dobijamo  $7 \cdot (1.1\text{GB} + 1.1\text{GB}) + 7 \cdot (350\text{MB}) + 7 \cdot (1.1\text{GB} + 1.1\text{GB})$  gde je treći deo sabirka najgori slučaj nedeljnih zahteva za preglede i operacije od strane doktora, dobijamo 55.3 GB neophodne keš memorije.

Zarad obezbeđivanja sistema od otkaza, uvešćemo redundantna skladišta podataka, kojim se učetvorostručuje potreban trajni memorijski prostor. Time dolazimo do  $4 \cdot 4\text{TB}$  neophodne trajne memorije.

Na osnovu svega navedenog, neophodno je  $\sim 16\text{TB}$  trajne memorije u sistemu, i  $\sim 54\text{ GB}$  keš memorije.

## 6. Strategija load balansera

Pošto najveći deo podataka koji prolaze kroz sistem čine podaci o pregledima, izdvojili bismo te podatke, i to samo za čitanje, u posebne njima dodeljene servere.

Pored toga, podaci o kredencijalima za prijavljivanje bi mogli da se takođe izdvoje u manji broj sopstvenih servera, pošto se podaci za prijavljivanje, osim prilikom prijavljivanja retko čitaju i menjaju u bazi.

Dalje, zahtevi za operacije i preglede često se upisuju i modifikuju, i važno je da im je pristup efikasan. Ali, mali udeo korisnika im ima pristup, pa bi se i oni mogli serverski odvojiti od ostatka podataka na sistemu.

Dakle, imali bismo posebne servere dodeljene za čitanje podataka o pregledima, kredencijalima za prijavu, zahteve za operacije i ostali saobraćaj, i load balanser, koji bi korisnike raspoređivao između njih. Raspoređivanje bi moglo da se vrši putem Round Robin algoritma, gde serveri u krug dobijaju po jedan zahtev korisnika, za vrstu zahteva za koju su osposobljeni. Serveri iz jedne grupe bi u tom slučaju bili ravnomerno opterećeni. Pored toga, ako jedan server iz grupe izađe iz funkcije, jednostavno je prestati sa slanjem zahteva u njega i malo više opteretiti druge, što ne bi trebalo značajno da uspori rad sistema.

## 7. Operacije za gledati radi boljeg rada sistema

Jedna od ovih operacija je klijentsko zahtevanje ručno kreiranih pregleda. Kreiranje pregleda zahteva podatke o već zakazanim pregledima, lekarima i klinikama. Moguće bi bilo jednu grupu servera dodeliti isključivo za čitanje ovih podataka, da bi pacijenti brže mogli da filtriraju klinike i lekare.

Druga složena operacija predstavlja dodelu termina i sobe za operaciju administratoru kliničkog centra. Operacija zahteva podatke o svim pregledima, lekarima i sobama na određenoj klinici. Mogao bi

se na nivou klinike napraviti pogled na sistem, koji prati sve podatke koje bi administratoru za operaciju mogli da zatrebaju, i koji bi ubrzao proces.

Dalje, pod pretpostavkom da se većina poslovanja obavlja u prvoj smeni (08:00 - 16:00), nešto manje u drugoj (16:00 - 00:00) i značajno manje u trećoj (00:00 - 08:00), broj servera koji opslužuje sistem ne mora biti isti svih 24 časova dnevno. Noću bi se deo servera mogao delimično ili potpuno ugasiti, ili bi se njihovo vreme moglo iznajmiti nekoj drugoj kompaniji. Pretpostavimo da je saobraćaj između 20:00 i 07:00 značajno manji nego van tog intervala, te da se tada deo servera može rasteretiti.

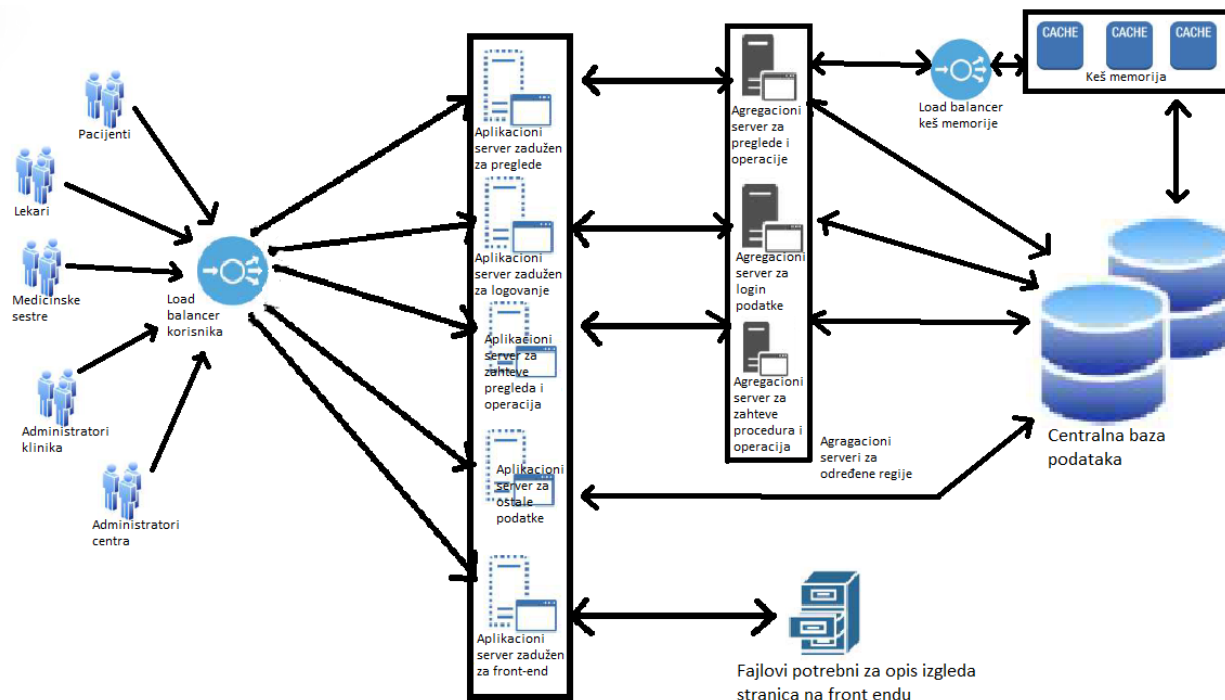
U letnjim mesecima se da pretpostaviti da će manje ljudi ići u bolnicu. Posle par godina rada sajta mogla bi se izvući statistika koja ruši ili potvrđuje ovu pretpostavku. Ako se uspostavi jasan šablon, gde su neki meseci drastično zauzetiji od ostalih iz godine u godinu, mogli bismo preduzeti jedan od sledećih koraka:

1) Scenario gde je 2-4 meseca godišnje značajno prometnije od ostataka godine: U ovim mesecima zakupiti dodatni serverski prostor, kako bi se osigurao pouzdan rad sistema.

2) Scenario gde su 2-4 meseca godišnje značajno manje prometni od ostatka godine: U ovim mesecima naš serverski prostor iznajmiti drugoj kompaniji.

Ova mera mogla bi se preduzeti samo ako se praksom ispostavi da je saobraćaj sistema iz godine u godinu u datom vremenskom intervalu drastično drugačiji nego ostatkom godine.

## 8. Kompletan crtež dizajna predložene arhitekture



Autori:

- Arsenijević Nikola RA 34-2016
- Tapai Tibor RA 75-2016
- Nikolić Dušan RA 187-2016