

MindMaps

Aplikacija za kolaborativno crtanje mapa uma

Jelena Živković 16581
Marija Stanković 16350

Sadržaj

[Sadržaj](#)

[Opis sistema](#)

[Funkcionalni zahtevi:](#)

[Nefunkcionalni zahtevi:](#)

[Arhitektura](#)

[Arhitekturni obrasci](#)

[Layered \(Slojeviti\) obrazac](#)

[Repository \(Skladište\) obrazac](#)

[Message broker](#)

[Strukturni dijagram](#)

[Bihevioralni pogled](#)

[Arhitektura klijentske komponente](#)

Opis sistema

MindMaps je Web aplikacija za kolaborativno crtanje mapa uma za vizuelno predstavljanje ideja, rešenja, znanja vezanih za neku temu u vidu dijagrama.

Funkcionalni zahtevi:

- registovanje i logovanje korisnika korišćenjem usernama i šifre
- kreiranje nove "sobe"
- dodavanje i brisanje članova "sobe"
- kreiranje novog dokumenta¹ unutar sobe
- ažuriranje (dodavanje elemenata) i brisanje dokumenta
- deljenje dokumenta sa ostalim članovima "sobe", pri čemu svaki od članova može da menja dokument
- razmena poruka među članovima iste sobe

Nefunkcionalni zahtevi:

- Brz odziv: Svaka promena koju jedan korisnik unese mora biti vidljiva ostalim korisnicima tog dokumenta u što kraćem vremenskom roku.
- Lakoća korišćenja: Sistem treba da bude intuitivan i lak za korišćenje
- Pouzdana konekcija tokom rada na projektu

¹ jedan dokument predstavlja jedan dijagram mape uma

Arhitektura

Arhitekturni obrasci

Layered (Slojeviti) obrazac

U aplikacije je implementiran Layered obrazac.

Višeslojna klijent-server arhitektura koja se sastoji od slojeva:

- Klijent
- Server:
 - Komunikacija sa klijentskom komponentom
 - Poslovna logika
 - Pristup bazi
- Baza: Sloj u kome se sladište podaci.

Repository (Skladište) obrazac

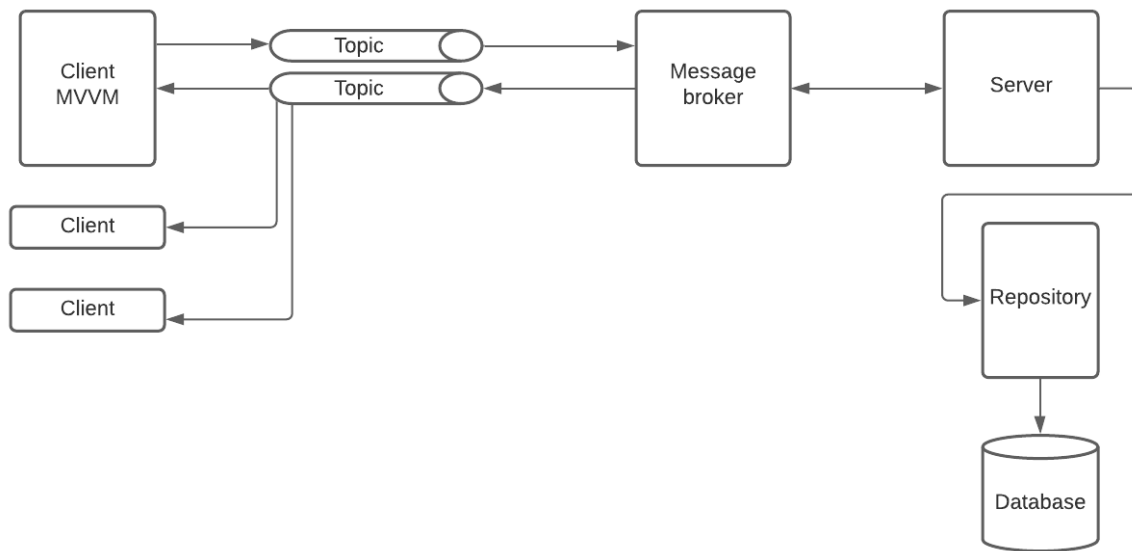
Za deo aplikacije koji se bavi pristupom i sladištenjem podataka koristimo Repository obrazac. Obezbeđuje izvršavanje osnovnih CRUD operacija nad podacima u bazi podataka. Izbegava direktan pristup bazi podataka iz poslovne logike.

Message broker

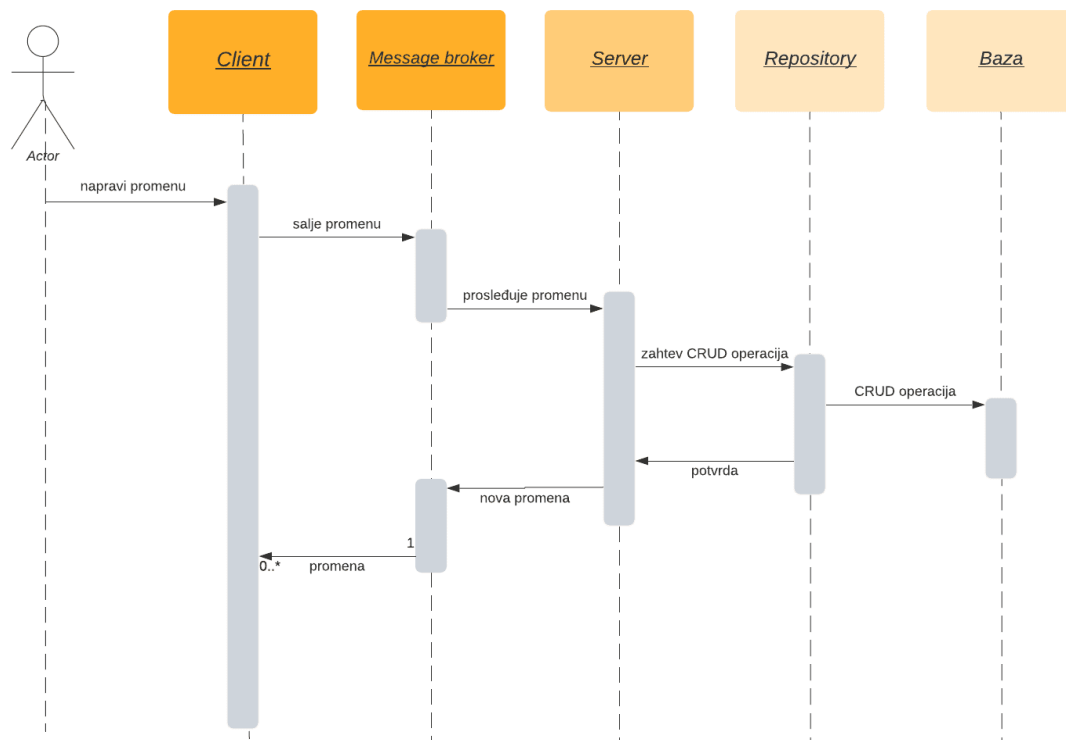
Za komunikaciju između servera i klijenta koristi se Message broker, koji prilagođava format poruke u zavisnosti kome je namenjena.

Kada korisnik unese neku promenu u dokument, server se obaveštava o toj promeni i tu promenu propagira ostalim zainteresovanim korisnicima po principu publish - subscribe.

Strukturni dijagram



Bihevioralni pogled



Arhitektura klijentske komponente

Za aplikacioni okvir koji koristimo za implementaciju klijentske komponente izabrale smo Angular. Angular implementira MVVM obrazac koji je varijanta MVC obrasca.

Glavne komponente u MVVM obrascu su *Model*, *View* i *ViewModel*. Cilj ovog obrasca je da se razdvoje *View* i *Model*. Komponenta *ViewModel* ima ulogu da selektuje i transformiše podatke iz *Modela* i prosledi ih u *View* i obratno. *View* služi za jednostavi prikaz, dok *Model* služi za čuvanje podataka.

View bind-uje *ViewModel* što znači da se promena na *View* automatski prosleđuje na *ViewModel*. *ViewModel* dalje utiče na promenu u *Modelu*. Isto tako, ukoliko se desi promena u *Modelu*, automatski se ažurira *ViewModel* što direktno izaziva promenu na korisničkom interfejsu (*View*).

