

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

Roko Fumić

# Studiu - Forum za studente

**Dokumentacija**

Kolegij - Programsko inženjerstvo

Nositelj - doc. dr. sc. Nikola Tanković

Pula, Travanj 2022.

<b>Sažetak</b>	<b>2</b>
<b>Uvod i motivacija</b>	<b>2</b>
<b>Razrada funkcionalnosti</b>	<b>3</b>
Use case dijagram sustava	3
Prototip sučelja	5
Klasni dijagram	7
<b>Implementacija</b>	<b>9</b>
Registracija	9
Prijava na račun	11
Pregled svih foruma	12
Pregled foruma	14
Pregled objava i komentara	16

## 1. Sažetak

Studiu je forum namijenjen studentima i svima zainteresiranim za teme vezane uz određene kolegije. Omogućuje objavljivanje u forume vezane uz relevantni kolegij te razgovaranje s drugima unutar komentara. Administratori imaju i prilagođeno sučelje sa jednostavnim funkcijama za vođenje stranice.

Studiu je izrađen pomoću Vue.js Javascript okvira, sučelje je napravljeno koristeći TailwindCSS, dok se Firebase koristi za autentifikaciju i bazu podataka.

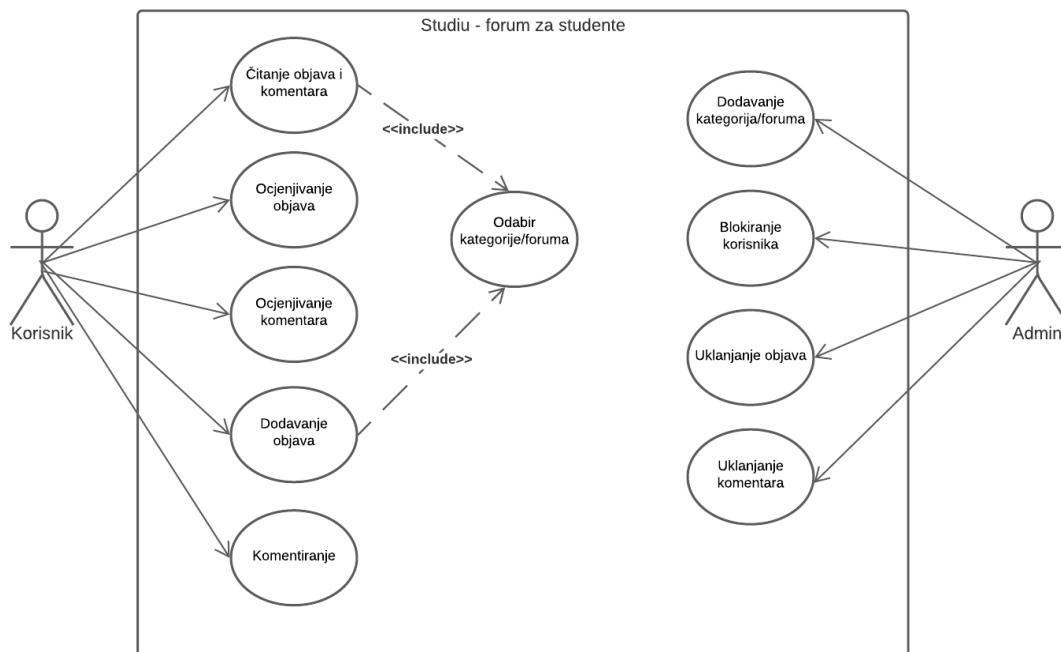
## 2. Uvod i motivacija

Studiu je web aplikacija, zamišljena kao tekstualni forum, namijenjena za diskusiju o sveučilišnim kolegijima i ostalim predmetima u obrazovanju. Ciljano tržište aplikacije su studenti fakulteta u Republici Hrvatskoj (i susjednim zemljama), ali i ostali zainteresirani za teme o kojima se raspravlja.

Svrha aplikacije je olakšati komunikaciju studenata o konkretnim temama vezanih uz kolegije koji ih interesiraju, kako bi uz međusobnu pomoći lakše shvatili gradivo koje pokušavaju naučiti.

### 3. Razrada funkcionalnosti

#### Use case dijagram sustava



Glavne funkcionalnosti za studente, odnosno, obične korisnike su:

- Odabir foruma - korisnik mora moći odabrati forum/kategoriju za koju je zainteresiran
- Čitanje objava - korisnik mora moći čitati objave na željenom forumu te znati tko je i kada napravio objavu
- Čitanje komentara - korisnik mora moći otvoriti željenu objavu i pročitati sve komentare napisane za nju
- Ocjenjivanje objava - korisnik mora moći pozitivno ili negativno ocijeniti bilo koju objavu (like/dislike), te vidjeti ukupni broj ocjena iste. Pojedini korisnik može objavu ocijeniti pozitivno ili negativno, ali ne oboje.

- Ocjenjivanje komentara - korisnik mora moći pozitivno ili negativno ocijeniti bilo koji komentar (like/dislike), te vidjeti ukupni broj ocjena istog. Pojedini korisnik može komentar ocijeniti pozitivno ili negativno, ali ne oboje.
- Dodavanje objava - korisnik mora moći dodati tekstualnu objavu na bilo koji forum. Objava mora sadržavati naslov, čime je ukratko opisuje. Objava može, ali ne mora, imati sadržaj u kojem je korisnik detaljnije opisuje.
- Dodavanje komentara - korisnik mora moći komentirati na bilo koju objavu željeni broj puta. Komentar imati tekstualni sadržaj.

Glavne funkcionalnosti za administratore su:

- Dodavanje kategorija/foruma - administrator mora moći dodati nove kategorije na Studiu. Dodavanje podrazumjeva unos naziva foruma/kategorije, što korisnicima omogućuje objavljivanje.
- Uklanjanje objava - administrator mora moći ukloniti bilo koju objavu na bilo kojem forumu, čime se ona uklanja iz baze podataka te postaje nevidljiva korisnicima.
- Uklanjanje komentara - administrator mora moći ukloniti bilo koji komentar na bilo kojoj objavi, čime se on uklanja iz baze podataka te postaje nevidljiv korisnicima.
- Blokiranje korisnika - administrator mora moći blokirati pristup Studiu bilo kojem korisniku. Blokiranim korisnikom smatra se onaj koji nema mogućnost prijave na svoj korisnički račun te čije korisničko ime više nije vidljivo drugim korisnicima.
- Korisničke funkcionalnosti - administrator mora moći koristiti sve funkcionalnosti koje imaju obični korisnici (objavljivanje, komentiranje, ocjenjivanje)

## Prototip sučelja





# KATEGORIJE

## STRUKTURE PODATAKA I ALGORITMI

Nauči sve o algoritmima, listama i Big O notaciji

## EKONOMIJA

Brojke, novac, excel...

## STATISTIKA

Diskusije o statistici

## MEHANIKA FLUIDA

Bernoulli?

## LATINSKI JEZIK

In vino veritas

## FUNKCIJSKO PROGRAMIRANJE

Currying, lambde i sve ostalo...

## ETIKA

Sve o filozofiji morala

## EVOLUCIJSKA PSIHOLOGIJA

Nauči sve o evolucijskoj psihologiji

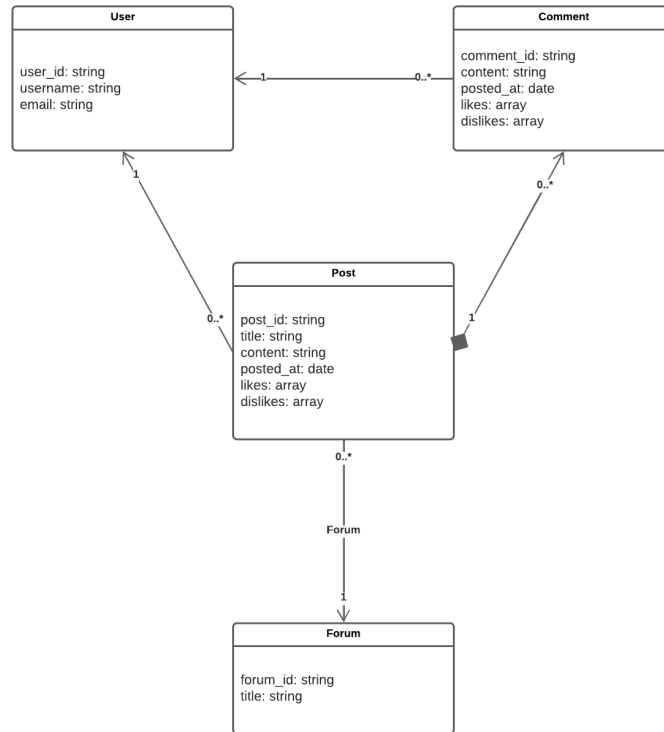
## STRUKTURE PODATAKA I ALGORITMI

		<b>Algoritam koji racuna jesu li dvije rijeci anagrami</b> Koja je najefikasnija implementacija?	ivan_ivanovic
25	7		
		<b>Zasto je ovo <math>O(n \log n)</math></b> Nije mi jasno zasto ovaj algoritam nije $O(\log n)$ ...	user1
25	7		
		<b>SPA literatura</b> Koje knjige preporucujete?	user112
25	7		
		<b>Stack u C++</b> Moze li mi netko objasniti kako napraviti stack u C++ koristeći klase?	mate123
25	7		
		<b>Koja je razlika između vezane liste i običnog polja?</b>	username12
25	7		
		<b>Jednostavna vizualizacija sorting algoritama u pythonu.</b>	user1
25	7		

## Klasni dijagram

Objekti koji se spremaju u bazu podataka su:

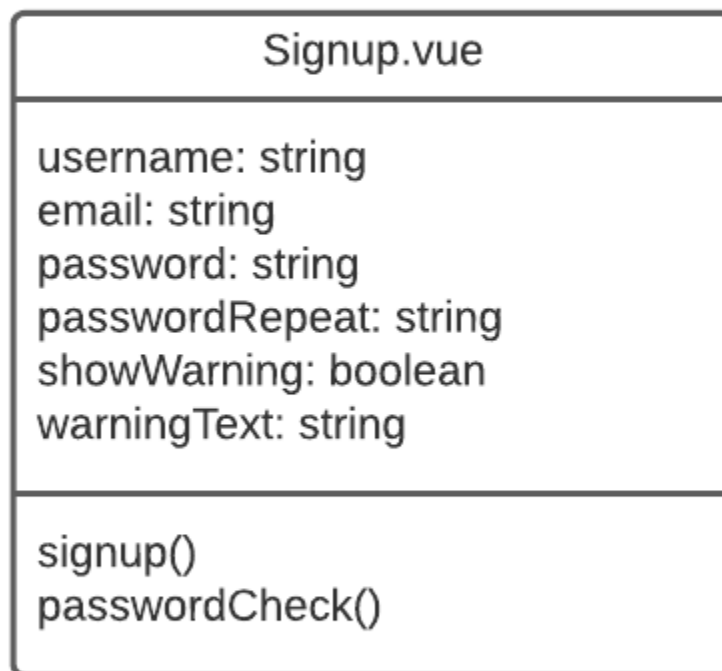
- Post - “glavni” objekt u bazi podataka; opisuje pojedinu objavu na forumu; sadrži polja za id, naslov, sadržaj, datum objave, likes, dislikes; referencira forum u kojem se nalazi objava, korisnika koji je napravio objavu, te sve komentare na objavi.
- Forum - objekt koji opisuje forum/kategoriju na Studios; sadrži polja za id i naziv foruma.
- User - objekt koji opisuje određenog korisnika; sadrži polja za id, korisničko ime, te email.
- Comment - objekt koji opisuje pojedini komentar na nekoj objavi; sadrži polja za id, sadržaj komentara, datum objave komentara, likes, dislikes; referencira objavu na koju se odnosi i korisnika koji je napisao komentar; Komentari se brišu iz baze podataka ukoliko se ukloni objava za koju su napisani.





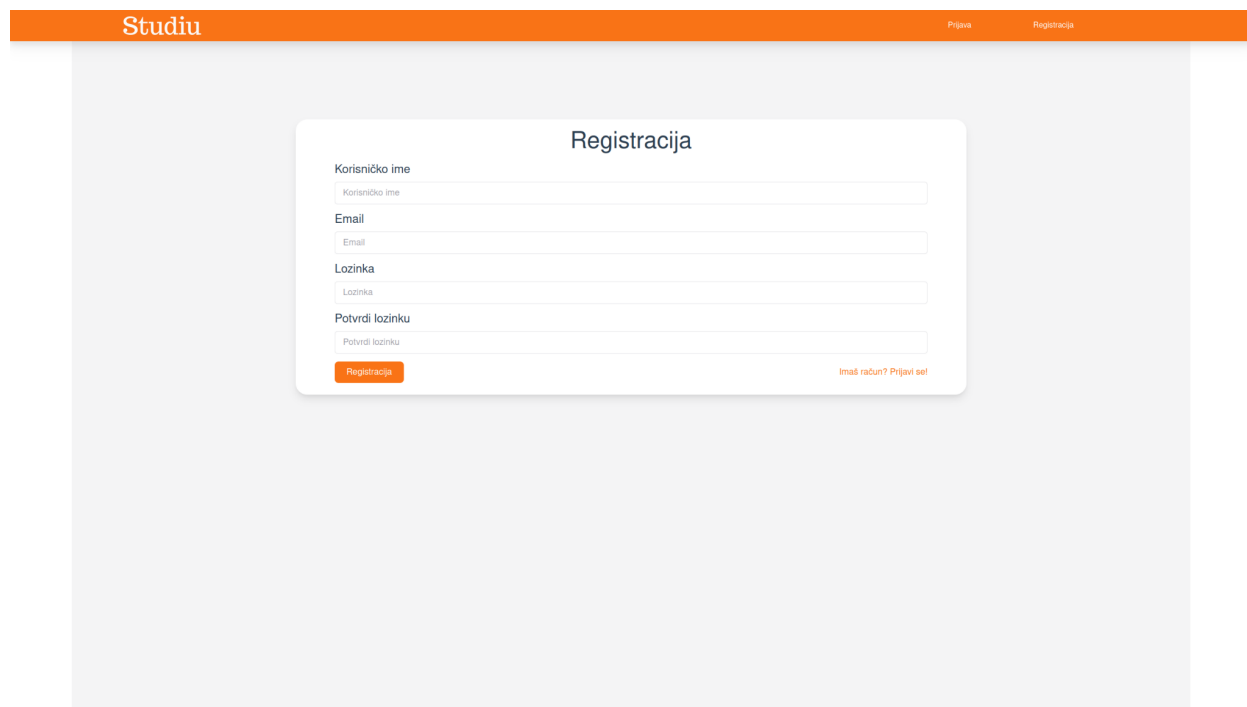
## 4. Implementacija

### Registracija



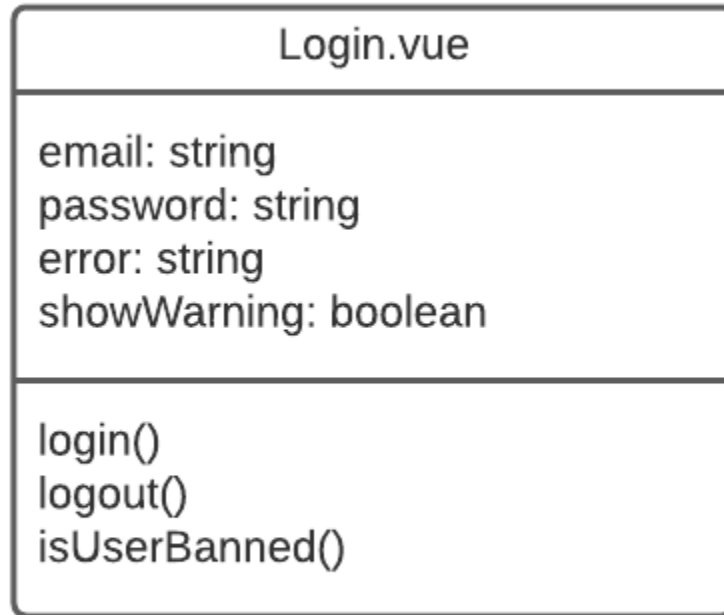
Cijeli proces registracije izvršava se unutar komponente “*Signup.vue*”. Ako je korisnik upisao korisničko ime, email i lozinke, pritiskom na gumb “Registracija” ili tipku enter, poziva se funkcija “*passwordCheck*” koja izvršava razne provjere vezane uz registraciju, poput ispravnosti formata email adrese, dostupnosti korisničkog imena, dužine lozinke i sličnog. U slučaju da neki od podataka nisu ispravni, na komponenti se prikazuje upozorenje korisniku sa relevantnim informacijama. Ukoliko su svi

podaci ispravno upisani, poziva se funkcija “*signup*” koja pomoću firebase autentifikacije stvara novog korisnika, zatim ga dodaje u firestore kolekciju “users” gdje se sprema korisničko ime i status korisnika (isBanned) radi lakše provjere tih podataka u ostalim dijelovima aplikacije. Ukoliko je sve uspješno izvršeno, vue router preusmjerava korisnika na komponentu “*Forums.vue*”.



The screenshot shows a web application interface with an orange header bar. On the left of the header is the word "Stadiu" in white. On the right are two links: "Prijava" and "Registracija", with "Registracija" being the active link. The main content area has a light gray background. In the center is a white registration form titled "Registracija". The form contains four input fields: "Korisničko ime", "Email", "Lozinka", and "Potvrdi lozinku". Below the "Lozinka" field is an orange "Registracija" button. At the bottom right of the form is a link that says "Imaš račun? Prijavi se!".

## Prijava na račun



Proces prijave na korisnički račun odvija se u komponenti “*Login.vue*”. Nakon unosa lozinke i email adrese, pritiskom na gumb “prijava” ili tipku enter, poziva se funkcija “*login*” koja poziva firebase autentifikaciju te prijavljuje korisnika. Nakon prijave dohvaća se korisnikov token, koji se zatim koristi u funkciji “*isUserBanned*” koja u firestore kolekciji “*users*” provjerava varijablu “*isBanned*”. Ako je korisnikov račun blokiran na Studiu, poziva se funkcija “*logout*” kojom se korisnika odjavljuje te vraća na komponentu za prijavu s upozorenjem o statusu korisnikova računa.

Studiu

Prijava

Registracija

Prijava

Email

Email

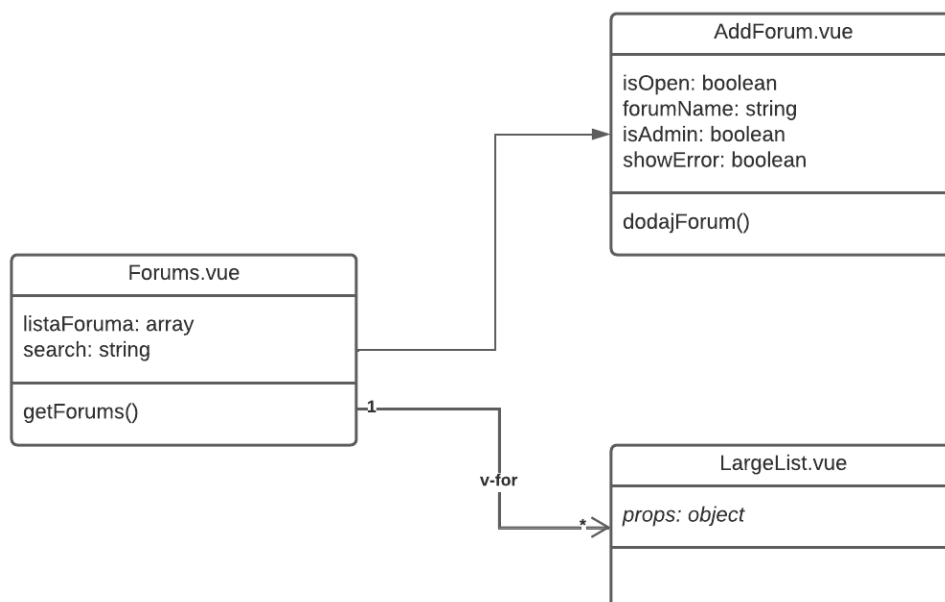
Lozinka

Lozinka

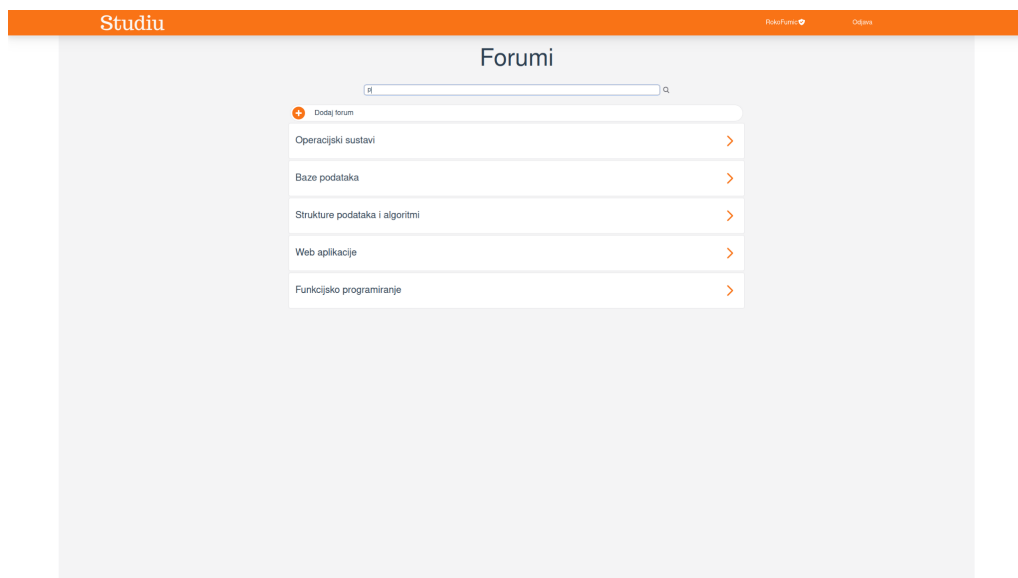
Prijava

Nemaš račun? Registriraj se!

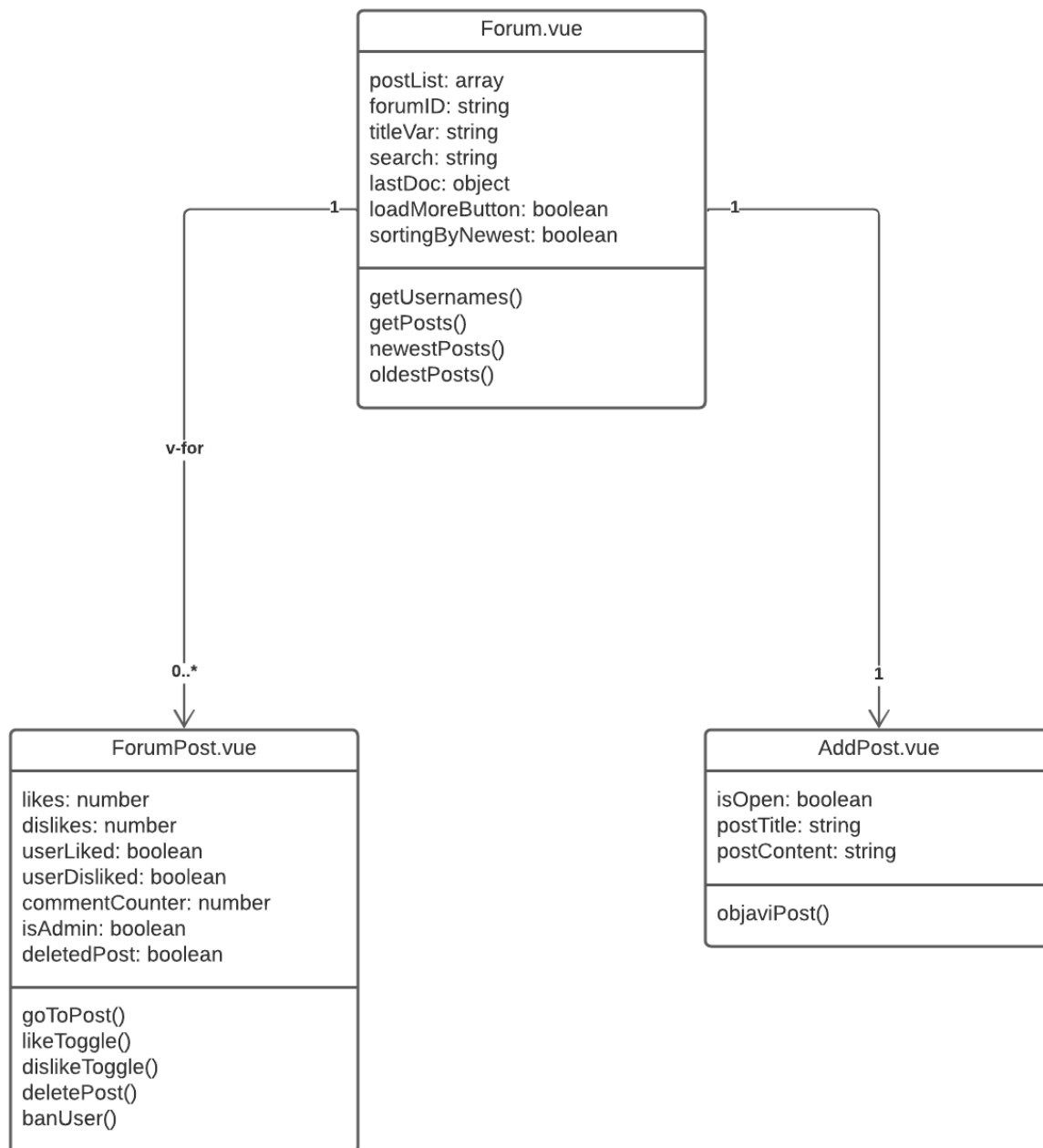
Pregled svih foruma



Komponenta/pogled “*Forums.vue*” sastoji se, uz dio za pretragu liste foruma, od dvije pomoćne komponente, “*LargeList.vue*” i “*AddForum.vue*”. Prilikom stvaranja komponente pokreće se metoda “*getForums*” kojom se iz polja “*forums*” u firestore bazi podataka dohvaćaju svi postojeći forumi te spremaju u polje “*listaForuma*”. Nakon što se popuni ta varijabla, koristeći v-for svojstvo komponente “*LargeList*” korisniku se prikazuju svi postojeći forumi. Kao argument za v-for koristi se computed svojstvo “*filteredForums*” koje služi za pretragu liste foruma. Komponenta “*LargeList.vue*” kao svoj *prop* parametar prima objekt koji se sastoji od naziva određenog foruma i njegovog id-a. Komponenta “*AddForum.vue*” provjerava je li korisnik administrator ili nije te ako je ona mu se prikazuje. Ukoliko je korisnik administrator, nakon što unese naziv novog foruma koji želi dodati pokreće se metoda “*dodajForum*” koja u firestore kolekciju “*forums*” dodaje novi forum.



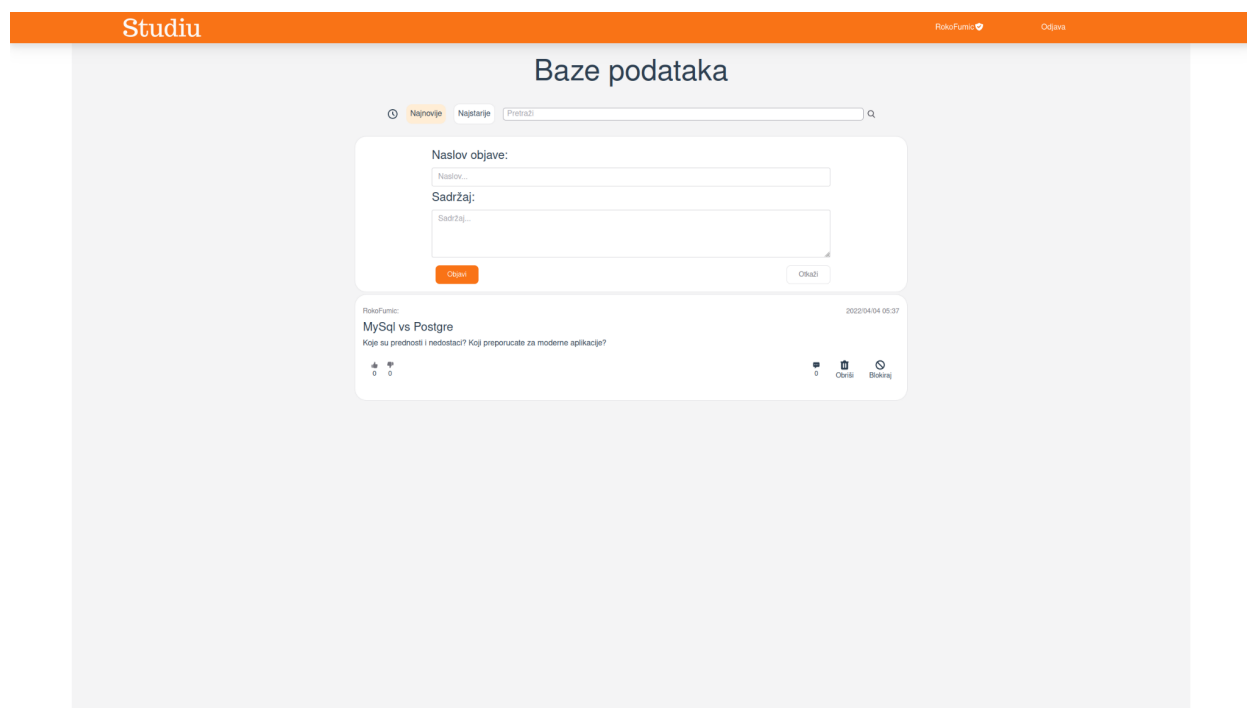
## Pregled foruma



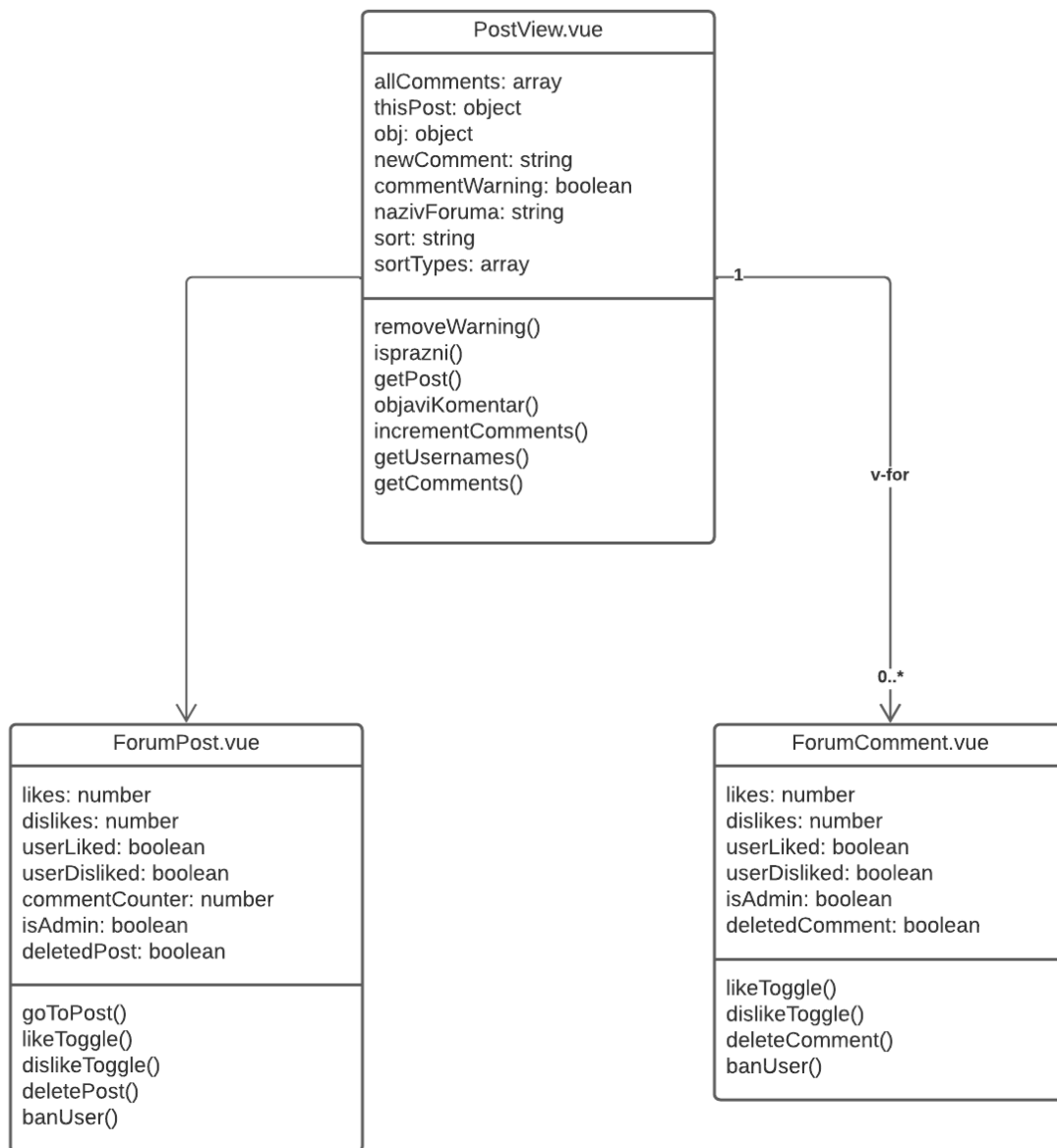
Za pregled objava na određenom forumu koristi se glavna komponenta “*Forum.vue*”, uz dvije pomoćne komponente “*ForumPost.vue*” i “*AddPost.vue*”. Ona kroz svoja props svojstva prima id foruma koji je korisnik odabrao, što se koristi u metodi “`getPosts`” koja dohvaća 10 najnovijih objava na forumu i unosi ih u polje “`postList`” koja se koristi kao `v-for` argument za komponentu “*ForumPost.vue*”. Na kraju izvođenja funkcije, referenca posljednjeg (desetog) učitano dokumenta pohranjuje se u varijablu `lastDoc`,

koja se koristi za učitavanje dodatnih objava ukoliko ih korisnik želi prikazati. Korisniku se omogućuje sortiranje objava po starosti koristeći funkcije “newestPost” i “oldestPost” koje pozivaju “getPosts” sa relevantnim argumentima sortiranja. Korisniku se prikazuje i komponenta “AddPost.vue” pomoću koje može dodati novu objavu, koristeći metodu “objaviPost”, u forum na kojem se nalazi.

Unutar komponente “ForumPost.vue” nalaze se razne bitne funkcionalnosti aplikacije, no više o njoj u nastavku.



## Pregled objava i komentara



Klikom na naslov objave koja ga interesira, korisnik dobiva detaljniji pregled objave zajedno sa njenim komentarima unutar komponente “*PostView.vue*”. U komponenti se nalazi jedna komponenta “*ForumPost.vue*” koja prikazuje detalje odabrane objave kao što su naslov, ime korisnika koji ju je objavio te omogućuje korisniku ocijenjivanje objave koristeći funkcije “*likeToggle*” i “*dislikeToggle*” koje svojim okidanjem dodaju id korisnika unutar polja “*like*” ili “*dislike*” u firestore kolekciju za odabranu objavu. Ako je korisnik administrator unutar komponente mu se omogućuje da ukloni tu objavu



ili blokira korisnika koji ju je objavio koristeći metode “deletePost” i “banUser”. Ispod objave nalazi se i polje za dodavanje komentara koje pomoću funkcije “objaviKomentar” sprema sadržaj komentara u podkolekciju “comments” koja se nalazi u firestore dokumentu od objave. Komponenta “*ForumComment.vue*” prikazuje se jednom za svaki komentar na objavi koristeći v-for. Unutar nje se nalaze metode za like i dislike te iste funkcije za admina kao i kod objave. Koristeći computed svojstvo “sortedComments” korisniku se omogućuje jednostavno sortiranje komentara.