Seminarski rad

Tema: Aplikacija za udomljavanje macaka

Sadržaj

[Verbalni opis 3](#_Toc103555558)

[Opis slučajeva korišćenja 4](#_Toc103555559)

[Opis procesa – dijagrami sekvenci 7](#_Toc103555560)

[Kreiranje posta 7](#_Toc103555561)

[Brisanje posta 9](#_Toc103555562)

[Konceptualni model 12](#_Toc103555563)

[Specifikacija REST API – ja 13](#_Toc103555564)

[Opis korišćenih tehnologija 16](#_Toc103555565)

[React 16](#_Toc103555566)

[Express 16](#_Toc103555567)

[Mysql 17](#_Toc103555568)

[TypeORM 17](#_Toc103555569)

[Reprezentativni delovi koda 18](#_Toc103555570)

[Link ka udaljenom repozitorijumu 20](#_Toc103555571)

# Verbalni opis

Potrebno je implementirati aplikaciju na kojoj je moguce postavljati objave o mackama kako bi sto pre nasle vlasnika.

Aplikacija treba da prepozna 2 vrste korisnika (regularan korisnik i administrator).

Korisnik može da postavljavlja postove, kreira i briše komentare, pretražuje ostale korisnike i dopisuje se sa njima.

Administrator treba da ima mogućnost pretrage korisnika kao i njihovo blokiranje. Blokiran korisnik se ne može ulogovati. Takođe, adminsitrator treba da ima uvid u određene statistike vezane za aplikaciju.

# Opis slučajeva korišćenja

U sistemu smo prepoznali sledeće slučajeve korišćenja:

* Prijava na sistem
* Registrovanje na sistem
* Odjava sa sistema
* Pretraga postova
* Kreiranje posta
* Brisanje posta
* Kreiranje komentara
* Brisanje komentara
* Pretraga korisnika
* Slanje poruka korisniku
* Pretraga poruka
* Blokiranje korisnika
* Pregled statistike



Slika 1 Dijagram slučajeva korišćenja

**SK1: Slučaj korišćenja – Kreiranje posta**

**Aktori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov: Sistem je uključen i korisnik je ulogovan pod svojom šifrom. Sistem prikazuje korsiniku formu za rad sa postom.**

**Osnovni scenario SK**

1. Korisnik unosi vrednosti za kreiranje posta. (APUSO)
2. Korisnik poziva sistem da kreira post sa datim podacima. (APSO)
3. Sistem kreira post sa datim podacima. (SO)
4. Sistem prikazuje korsiniku poruku: „Sistem je usepšno kreirao post“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da kreira proizvod on prikazuje korsiniku objavu: „Sistem ne može da kreira post“.(IA)

**SK2: Slučaj korišćenja – Brisanje posta**

**Aktori SK**

Korisnik

**Učesnici SK**

Korisnik i sistem

**Preduslov: Sistem je uključen i korisnik je ulogovan pod svojom. Sistem prikazuje korsiniku formu za rad sa postom.**

**Osnovni scenario SK**

1. Korisnik vrednost po kojoj pretražuje post. (APUSO)
2. Korisnik poziva sistem da pronađe post po zadatoj vrednosti. (APSO)
3. Sistem pretražuje post po zadatoj vrednosti. (SO)
4. Sistem prikazuje korisniku post.(IA)
5. Korisnik poziva sistem da obriše post.(APSO)
6. Sistem briše odabrani post. (SO)
7. Sistem prikazuje korsiniku poruku: „Sistem je usepšno obrisao post“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da pronađe post on prikazuje korsiniku objavu: „Sistem ne može da pronađe post“.(IA) prekida se izvršenje SK
  2. Ukoliko sistem ne može da obriše post on prikazuje korsiniku objavu: „Sistem ne može da obriše post“.(IA)

# Opis procesa – dijagrami sekvenci

## Kreiranje posta



Slika 2



Slika 3

## Brisanje posta



Slika 4



Slika 5



Slika 6

# Konceptualni model



Slika 7 Konceptualni model

# Specifikacija REST API – ja

|  |  |
| --- | --- |
| Opis funkcije | Vrati sve postove |
| HTTP metoda | GET |
| URL | /post |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Primer:  [  {  "id":4,  "createdAt":"2022-04-10T20:22:52.807Z",  "postCategoryId":1,  "userId":7, "content":"<b>Sadrzaj</b>",  "comments":[],  "animalType": {  "id":1,  "value":"persian"  },  "user":{  "id":7,  "firstName":"a",  "lastName":"a",  "email":"a@test.com",  "admin":false,  "blocked":false, "imageUrl":"https://localhost:8000/img/babd2c9f-fc37-49a3-8b87-a0881e4e5b8e.png"}}  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Vrati korisnike |
| HTTP metoda | GET |
| URL | /user |
| URL parametri | Search (string), page(integer), size (integer) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Primer:  {  "totalElements":3,  "size":3,"page":0,  "content":[  {  "id":7,  "firstName":"a",  "lastName":"a",  "email":"a@test.com",  "admin":false,  "blocked":false, "imageUrl":<https://localhost:8000/img/babd2c9f-fc37-49a3-8b87-a0881e4e5b8e.png>  },  {  "id":8,  "firstName":"Milica",  "lastName":"Tomic",  "email":"m@test.com",  "admin":false,  "blocked":false, "imageUrl":<https://localhost:8000/img/32933ab5-a7c9-470d-a170-0440976cb498.png>  }  ]  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiraj post |
| HTTP metoda | POST |
| URL | /post/ |
| URL parametri | (nema) |
| HTTP body parametri | Primer:  {  "animalTypeId":1,  "content":"<b>Sadrzaj</b>"  } |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Primer:  {  "id":4,  "createdAt":"2022-04-10T20:22:52.807Z",  "postCategoryId":1,  "userId":7,  "content":"<b>Sadrzaj</b>",  "comments":[],  " animalType": {  "id":1,  "value":"persian"  },  "user":{  "id":7,  "firstName":"a",  "lastName":"a",  "email":"a@test.com",  "admin":false,  "blocked":false, "imageUrl":"https://localhost:8000/img/babd2c9f-fc37-49a3-8b87-a0881e4e5b8e.png"}} |
| Format izlaznih parametara | application/json |

## Opis korišćenih tehnologija

Prilikom izrade aplikacije korišćene su sledeće tehnologije:

* React – kreiranje poslovne logike na strani klijenta
* Express – kreiranje servera
* Typeorm – objektno – relacioni maper
* Mysql – baza podataka

## React

React predstavlja biblioteka za kreiranje korisničkog interfejsa i poslovne logike na klijentskoj strani. React služi za kreiranje SPA web aplikacija odnosno aplikacija koje imaju samo jednu html stranu i kojima glavnu ulogu ima javascript.

Glavni koncept u React – je komponenta.

Komponenta predstavlja nešto što može da se iscrta kao jedna celina.

Celokupni prikaz aplikacije dobija se kompozicijom raznih elemenata u okviru jedne glavne komponente koja se, najčešće, naziva App.

Komponente mogu definisane kao funkcija i kao klasa.

Svaka komponenta ima svoje atribute koje je dobila komponente roditelja i svoje stanje koje je ona sama kreirala. Komponenta može da menja svoje stanje ali ne može da menja svoje atribute. Prilikom promene stanja poziva se posebna funkcija koja to radi nakon čega se prikaz na ekranu automatski menja na svim mestima koja su se iscrtavala na osnovu tog stanja.

## Express

Predstavlja „framework“ za izradu serverske strane u Node.js okruženju. Za razliku od većine ostalih servera, Express ne podržava višenitno programiranje. Problem konkurentnosti on rešava pomoću asinhronog programiranja i „event loop“ – a.

Express aplikacije su zamišljene da rade kao REST servisi, a obrada zahteva klijenta vrši se na sledeći način:

* Pristiže zahtev klijenta
* Izvršavaju se funkcije koje su definisane da se izvrše nad svakim zahtevom („middleware“)
* Zahtev se rutira na osnovu svog URL – a
* Izvršava se funkcija pridodata za dati URL
* Šalje se odgovor korisniku

## Mysql

Relaciona baza podataka koja je besplatna za korišćenje.

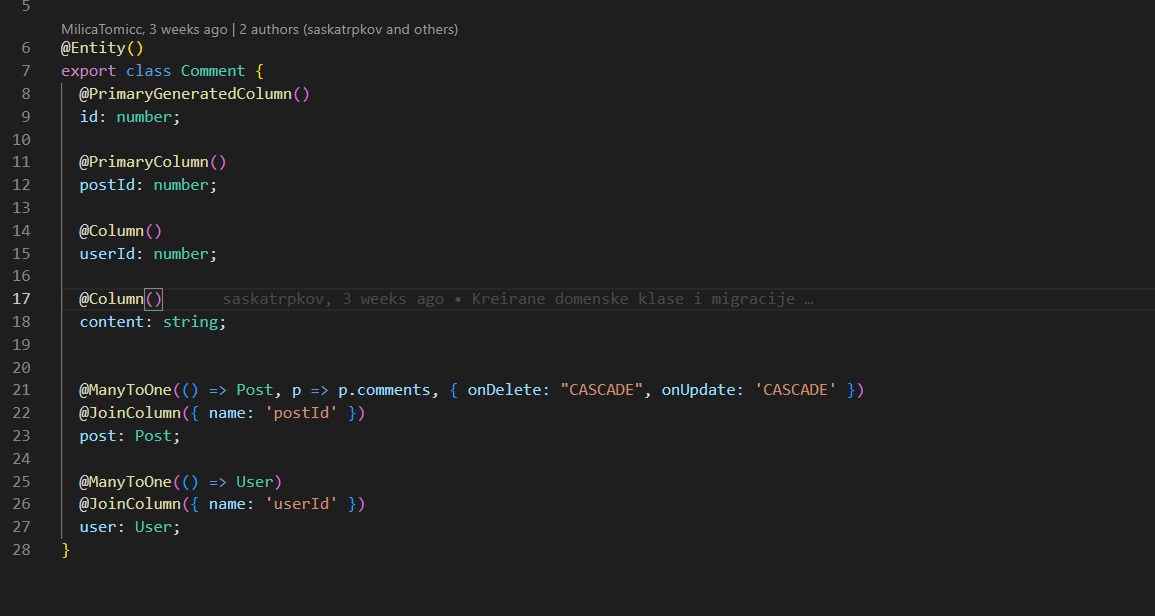
## TypeORM

Objektno – relacion maper u javascript – u. Pravila prevođenja vrše se anotacijom klasa u programskom kodu. Sva konfiguracija nalazi se fajlu ormconfig.json. TypeORM omogućava rad sa bazom na 2 načina:

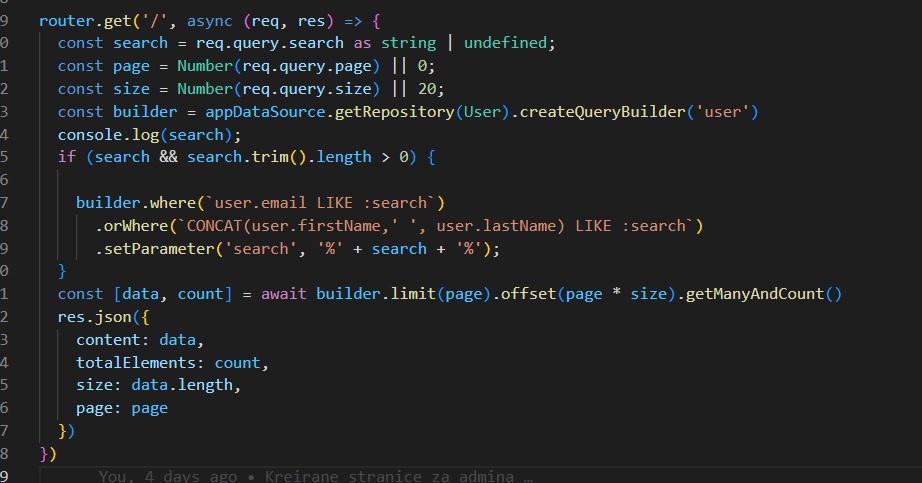
* Preko repozitorijuma (Repository pattern)
* Preko menadžera

U prvom slučaju olakšan je rad sa bazom ali su mogućnost kontrole toka i prilagođavanje rada ograničeni dok je u drugom omogućena mnogo veća kontrola i kastomizacija na račun komplikovanijeg koda.

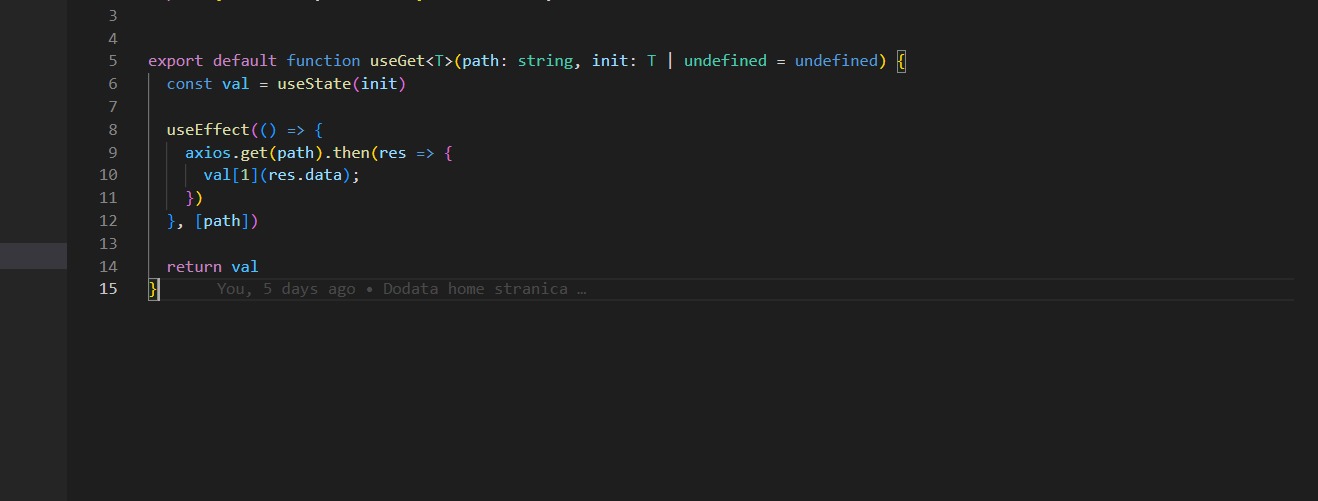
# Reprezentativni delovi koda



Slika 9 Domenska klasa Komentar



Slika 11 Pretraga korisnika



Slika 12 AJAX poziv

# Link ka udaljenom repozitorijumu