

*Baza podataka poslodavaca I radnika****.***

*IT350 – Baze podataka*

*Projektna dokumentacija*

***Student****:*

Aleksa Cekić 4173

***Asistent****:*

Tamara Vukadinović

# *Apstrakt Projektnog Zadatka*

## Opis problema

Potrebno je napraviti aplikaciju sa bazom podataka koji će raditi kao veza između poslodavaca I radnika (koje taj poslodavac angažuje). Sajt mora da podržava tri vrste korisnika: administratore, poslodavce i radnike.

Administrator je zadužen za održavanje aplikacije, a po potrebi može da kreira naloge korisnicima (ukoliko nisu vični tome).

Poslodavci postavljaju poslove od kojih svaki ima naziv, opis, maksimalni budžet, dužinu trajanja, broj potrebnih radnika I ključne reči (markira se kao “Aktivan”). Radnici, na objavljeni posao, postavljaju svoju ponudu za rad na tom poslu, koja mora da uključuje kratak opis () I predloženu cenu. Takođe, radnici mogu da postavljaju file-ove vezane za projekte (skice, prethodne radove, …). Jedan radnik može biti angažovan samo na jednom poslu. Ukoliko se za posao angažuje više radnika, oni se udružuju u tim I imaju svog predstavnika (vođu radova) koji pregovara sa poslodavcem.

Na osnovu prikupljenih ponuda, poslodavac bira kome će dodeliti posao I sa njim počinje pregovore (markira se “kao Pregovori”). Poslodavci i radnici su u stanju da razmenjuju poruke, poruke između poslodavca I radnika mogu da budu vezane za projekat. Nakon završetka pregovora ,ukoliko su pozitivno završeni, posao se dodeljuje je radnicima (markira se kao “Radovi u toku”). Ukoliko nije došlo do dogovora, posao se vraća na listu aktivnih.

Nakon završetka radova, posao se markira kao “Završen”, a poslodavac može da oceni kvalitet radova i radnike.

## SQL upiti vezani za projekat

a. Sve trenutno aktivne poslove na kojima se radi građevina

b. Poslodavce, uređene rastući po broju projekata koji su postavili ili po imenu poslodavca (moguće je selektovati koje uređenje želimo – da li po imenu ili broju poslova)

c. Izveštaj o poslodavcima sa poslovima u svim stanjima, aktivnim poslovima, poslovima u pregovorima, poslovima u toku I završenim poslovima (moguće je kroz aplikaciju selektovati sve ili jedno stanje).

d. Radnike, uređene po broju poslova koje su završili ili po imenu (moguće je selektovati koje uređenje želimo)

e. Sve radnike koje koji trenutno nisu angažovani, ali I radnike koji rade na nekom poslu (selektovati kroz aplikaciju šta želimo da prikažemo)

f. Projekte poređane po broju poruka koji su na njima razmenjeni

g. Radnike sa najvećom zaradom u svakom mesecu u godini h. Radnike sa najvećom ocenom od strane poslodavaca

# *Uvod*

## Cilj

Cilj ovog projekta je da se dizajnira sistem koji prati komunikaciju između poslodavca koji postavlja poslove I radnika koje te poslove izvršava.

Sistem takođe mora omogućiti lak i efikasan način izmene kao i dodavanje novih korisnika bilo kog tipa (administrator, poslodavac ili radnik).

Sama baza podataka mora biti maksimalno efikasna, sa što manje redudatnih informacija kao i bespotrebnih tabela i view-a.

## Realizacija

Baza podataka će biti kreirana u **MySQL-u** na **Windows** sistemu uz pomoć SQL upita i phpMyAdmin alata.

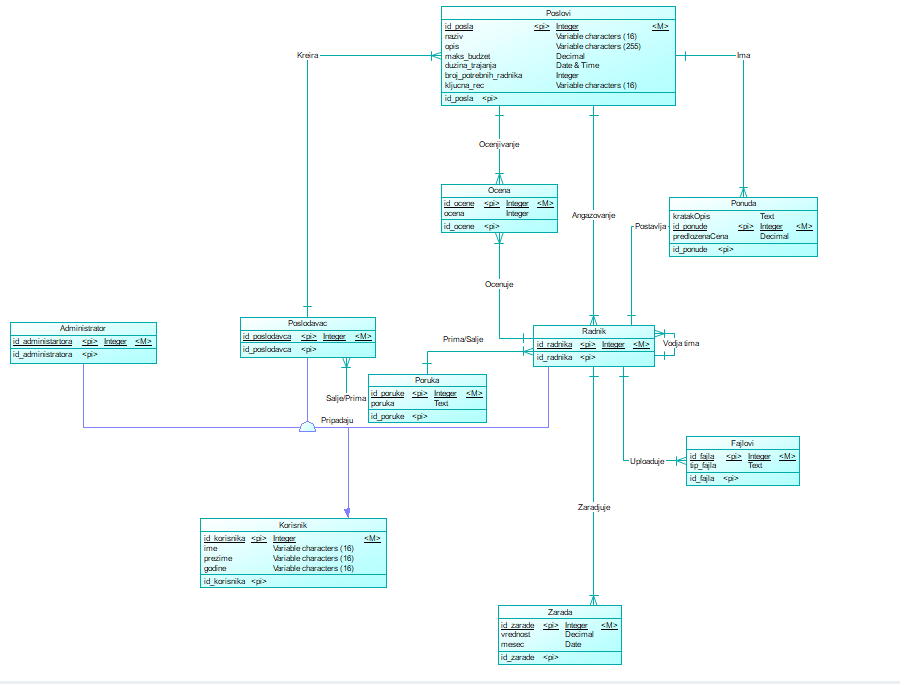
Dizajniranje konceptualnog modela se izvršava uz pomoć **Powerdesigner-a**.

Za izradu SQL upita koristićemo **XAMPP phpMyAdmin** aplikaciju. **CRUD aplikacija** će biti napravljena u **Angular** framework za prikaz podataka na aplikaciji, a funkcionalnost I samo povezivanje sa bazom radi se na Java framework pod imenom **Spring.**

# *Projektno rešenje*

## Klasni model baze podataka

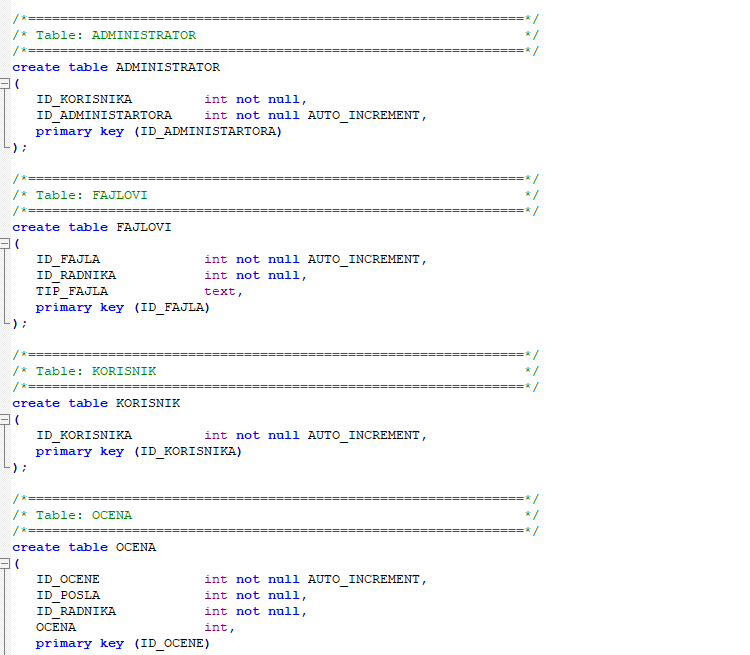
Klasni model predstavlja prvi najniži nivo apstrakcije problema koji je dobijen iz detaljnog opisa projektnog zadatka.



*Slika 1 Konceptualni model baze podataka*

## Kreiranje baze

Sledeći korak je generisanje fizičkog modela od već postojaćeg klasnog modela. Odakle možemo da generišemo SQL kod koji ćemo da koristimo za kreiranje same baze podataka tako što ćemo je importovati sql fajl unutar phpMyAdmin.



*Slika 2 Generisanje Baze podataka uz pomoć phpMyAdmin-a*

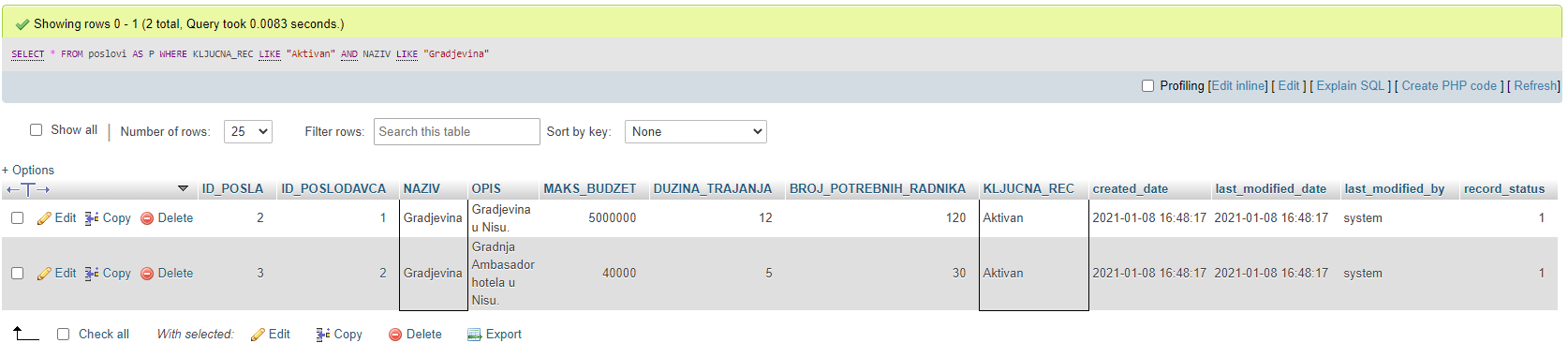
Nakon toga unosimo nekoliko redova u svajok tabeli baze podataka kako bismo uspešno istestirali sve upite.

Unošenje informacija se može izvršiti preko insertovanje SQL fajla, korišćenjem grafičkog korisničkog interfejsa na phpMyAdmin-u ili direktno kucanjem SQL upita preko konzole na phpMyAdmin-u.

# *Rešenja SQL upita*

## Upit *a*

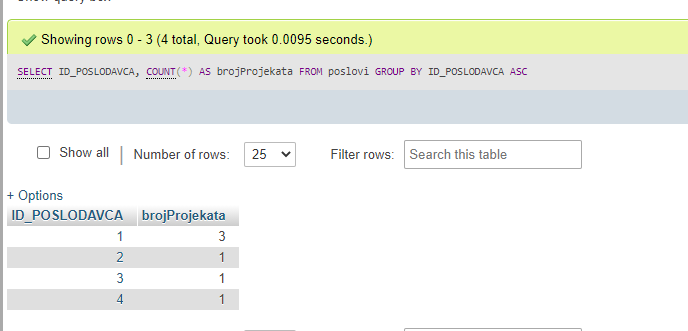
Sve trenutno aktivne poslove na kojima se radi građevina



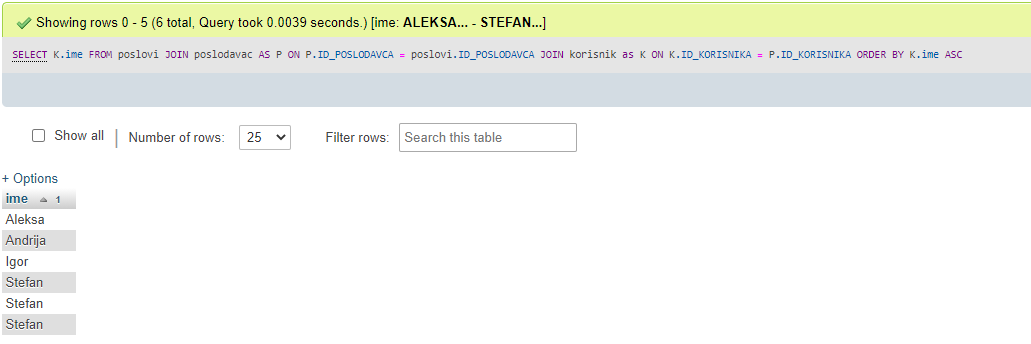
## Upit *b*

Poslodavce, uređene rastući po broju projekata koji su postavili ili po imenu poslodavca (moguće je selektovati koje uređenje želimo – da li po imenu ili broju poslova)

1.

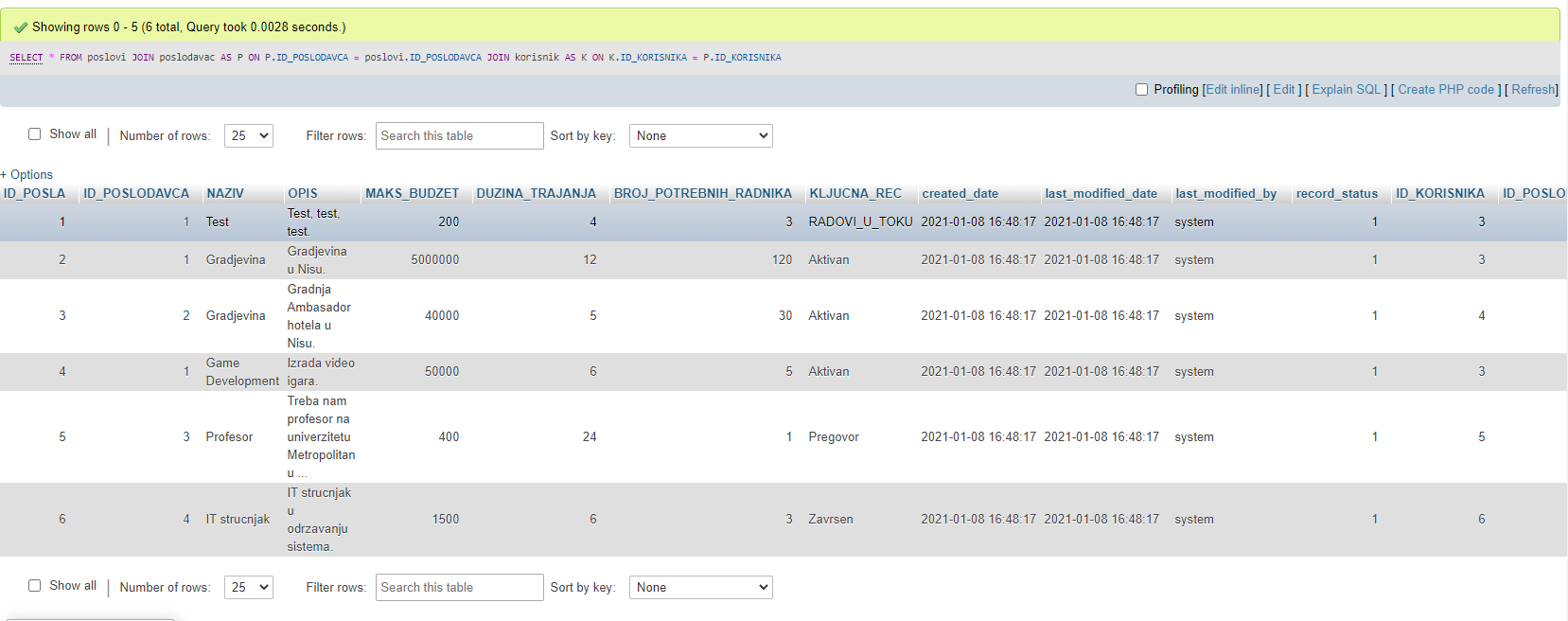


2.



## Upit *c*

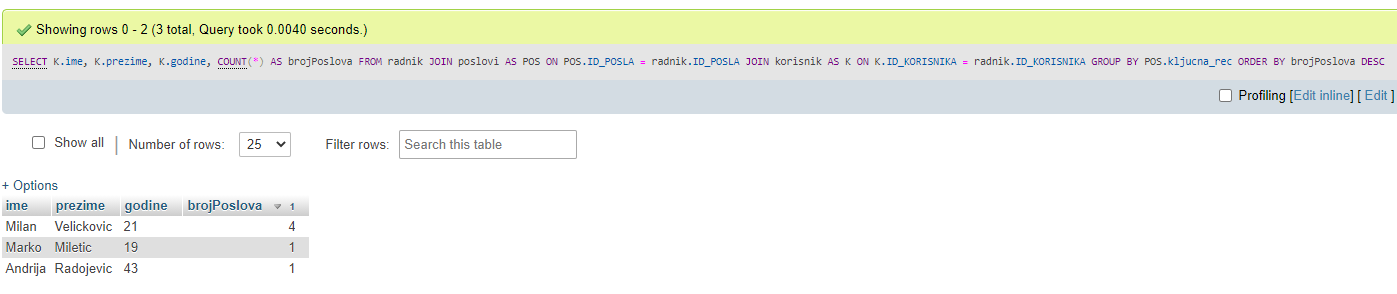
Izveštaj o poslodavcima sa poslovima u svim stanjima, aktivnim poslovima, poslovima u pregovorima, poslovima u toku I završenim poslovima (moguće je kroz aplikaciju selektovati sve ili jedno stanje).



## Upit d

Radnike, uređene po broju poslova koje su završili ili po imenu (moguće je selektovati koje uređenje želimo).

1.



2.



## Upit *e*

Sve radnike koje koji trenutno nisu angažovani, ali I radnike koji rade na nekom poslu (selektovati kroz aplikaciju šta želimo da prikažemo)

1.

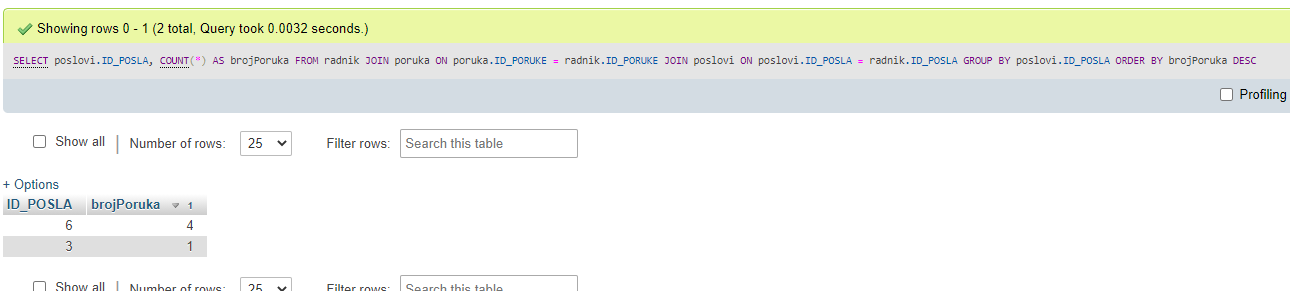


2.



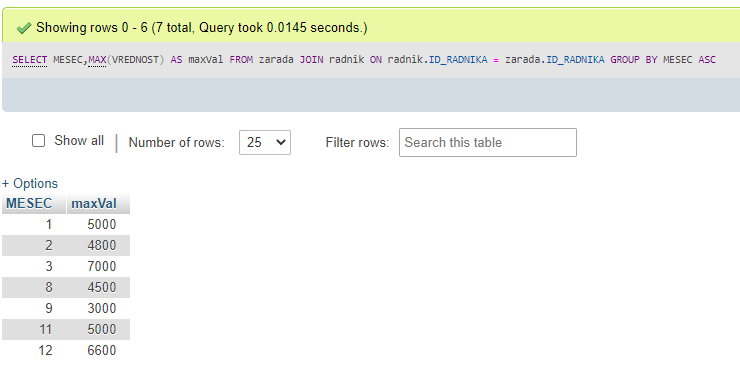
## Upit *f*

Projekte poređane po broju poruka koji su na njima razmenjeni



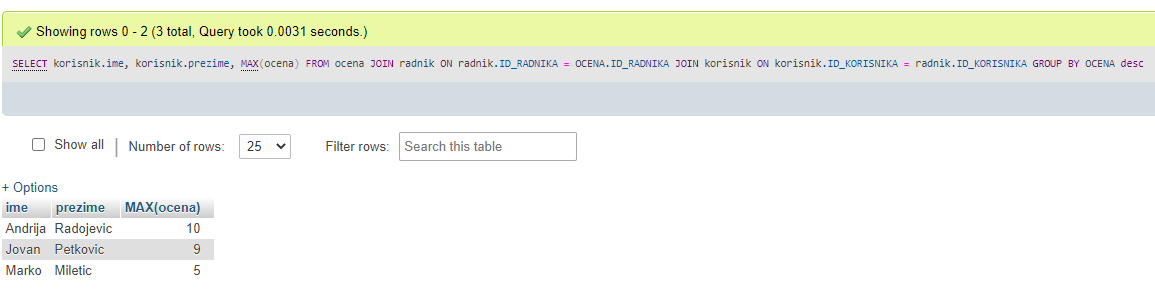
## Upit g

Radnike sa najvećom zaradom u svakom mesecu u godini



## Upit h

Radnike sa najvećom ocenom od strane poslodavaca



# *Aplikacija*

Aplikacija predstavlja izgled I funkcionalnost CRUD aplikacije. Aplikacija izgleda jako prosto, ima jedan navbar na kome se nalazi input pomoću koga možete da pretražite korisnika iz baze po imenu, prezimenu ili broju godina. Takođe na sredini aplikacije nalazi se jedna tabela koja izvlači podatke I prikazuje ih. Takođe nalaze se par opcija koje vrše CRUD funkcionalnosti.

Aplikacija je napravljena u framework-u Angular na front end-u I u Spring framework-u na back end-u.

Aplikacija je napravljena kao neki Administratorski interfejs za kontrolu korisnika u bazi podataka. Izgled je napravljen u Angular frameworku korišćenjem Angular Material-a. Na back-endu generisan je ceo spring projekat iz eksportovane sql baze.

U nastavku ćemo videti kako itgleda CRUD stranica za tabelu “Korisnik”.

## Glavna strana

*Slika 10 Izgled glavne strane*

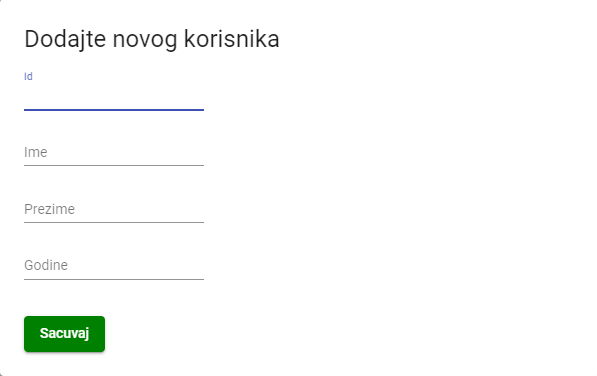
## Pregled korisnika (READ)



*Slika 11 Strana za pregled korisnika*

Klikom na dugme mоgu se videti podaci korisnika. Ti podaci su upisani u polja koja nisu editable, što znači da se oni ne mogu menjati.

## Dodaj/napravi korisnika (CREATE)

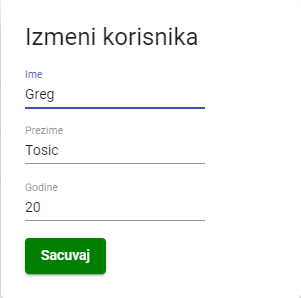


*Slika 12 Strana za kreiranje novog korisnika*

Ovde se nalaze forme potrebne za kreiranje novog korisnika. Id sam ostavio zbog toga što kada sam generisao back-end, generisao sam ga bez auto\_increment jer ga nije htelo izgenerisat sa njime.

Kada se sva polja popune, korisnik se dodaje u bazi.

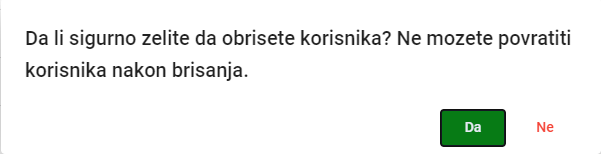
## Ažuriraj korisnika (UPDATE)



*Slika 13 Izgled ažuriranja korisnika*

Klikom na dugme za ažuriranje, podaci se izvlače u polja i mogu se menjati. Nakon toga izmenjeni podaci se čuvaju na mesto tog korisnika u bazi.

## Brisanje korisnika (DELETE)



*Slika 14 Izgled brisanje korisnika*

Korisnik se briše tako što se klikne na dugme za brisanje, i kada se to desi, izađe poruka potvrđivanja brisanja. Ukoliko korisnik klikne na da, selectovani korisnik se briše iz baze.

# *Zaključak*

Aplikacija realizuje osnovne CRUD operacije nad definisanom bazom podataka. Aplikacija pored osnovnih CRUD funkcija dodatno omogućava pretraživanje korisnika po imenu, prezimenu ili godinama.

# *Literatura*

LAMS literatura, lekcije DZ01-DZ13 : <http://lams.metropolitan.ac.rs:8080/lams/index.do>

Angular dokumentacija: https://angular.io/docs

Angular materijali: <https://material.angular.io/>

Spring dokumentacija: https://docs.spring.io/spring-framework/docs/current/reference/html/