

Univerzitet u Beogradu  
Fakultet organizacionih nauka  
Laboratorija za veštačku inteligenciju

Seminarski rad iz predmeta Alati i metode softverskog inženjerstva

TEMA: Veb aplikacija za ocenjivanje predavača

Mentor:

Đurić dr Dragan

Student:

Katarina Jerinić 2019/3704

Beograd, 2020.

## Sadržaj

1. Verbalni opis .....	3
2. Model slučajeve korišćenja .....	3
3. Korisničko uputstvo.....	4
4. Implementacija.....	7
5. Korišćenje aplikacije .....	9

## 1. Verbalni opis

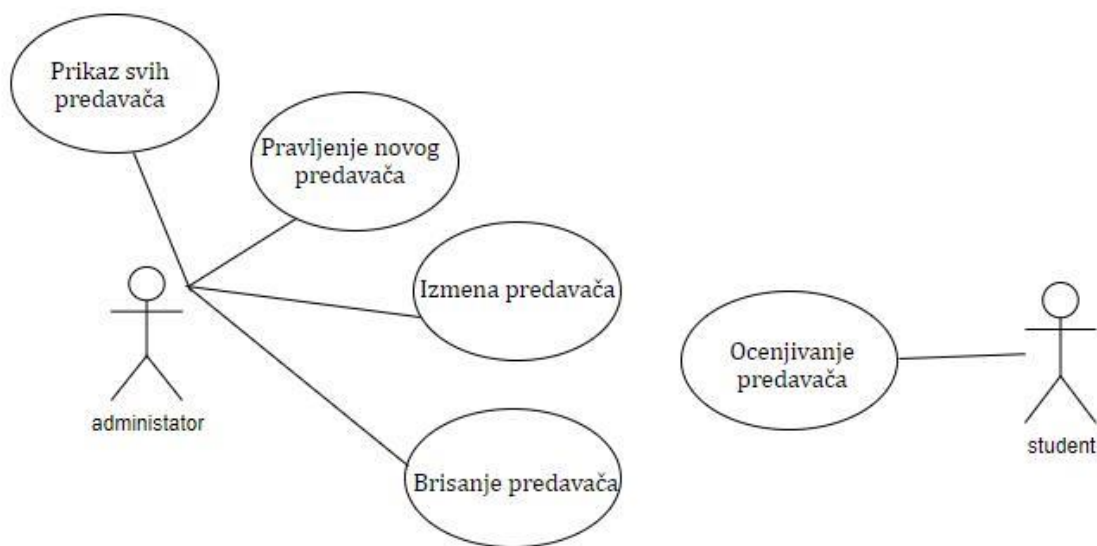
Aplikacija je razvijena kako bi student anonimno mogao ostaviti ocenu i komentar za odabranog predavača. Ona takođe nudi mogućnost adminskog pristupa i pravljenja novog predavača, kao i izmene i brisanja postojećeg predavača.

## 2. Model slučajeva korišćenja

Na osnovu verbalnog opisa identifikovano je 5 slučajeva korišćenja:

1. SK1: Prikaz svih predavača
2. SK2: Pravljenje novog predavača
3. SK3: Izmena predavača
4. SK4: Brisanje predavača
5. SK5: Ocenjivanje predavača

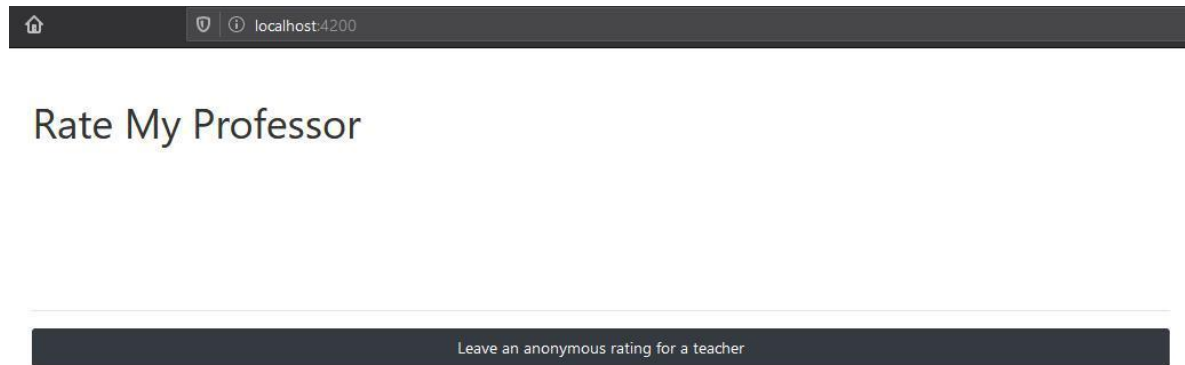
Oni su prikazani na dijagramu slučajeva korišćenja na slici 1.



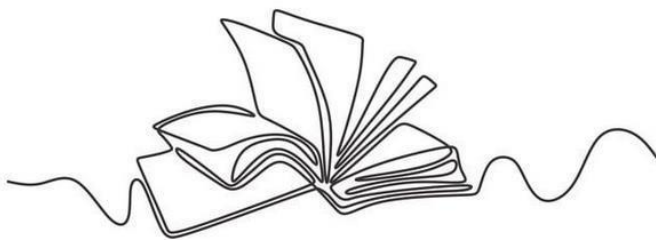
Slika 1

### 3. Korisničko uputstvo

Početni interfejs aplikacije prikazan je na slici 2.

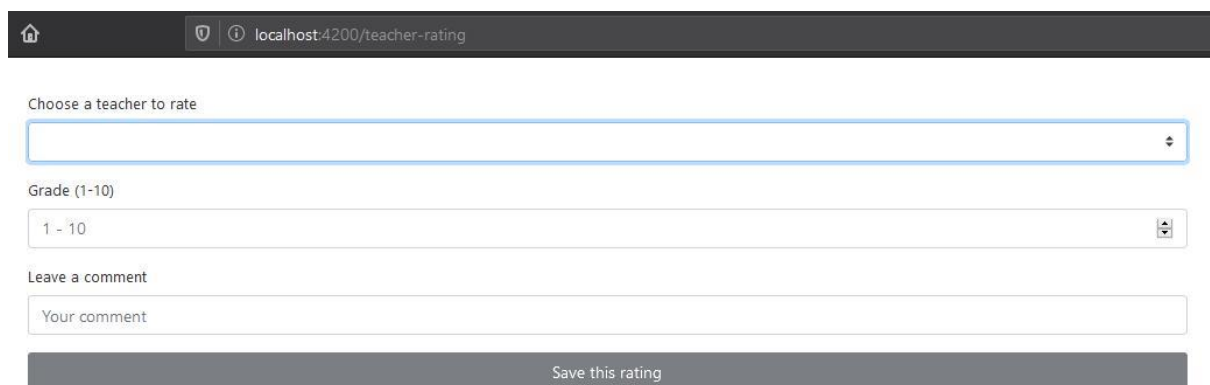


The screenshot shows a web browser window with the address bar displaying 'localhost:4200'. The main heading is 'Rate My Professor'. Below the heading is a dark grey button with the text 'Leave an anonymous rating for a teacher'.



Slika 2

Klikom na dugme se ide na formu za ocenjivanje predavača, prikazanu na slici 3. Svako od polja je obavezno, a polje za unos ocene prihvata samo brojčane vrednosti od 1 do 10, dok polje za unos komentara prihvata bilo kakvu tekstualnu vrednost.

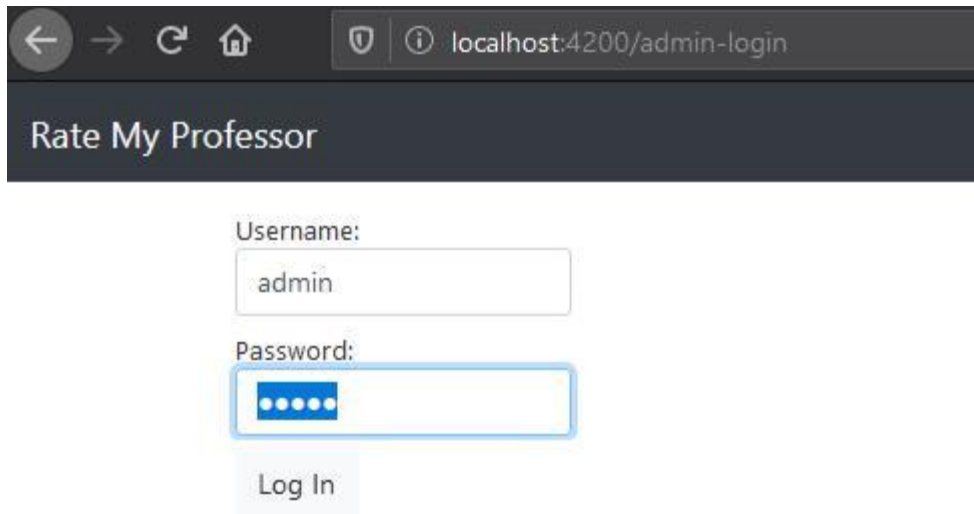


The screenshot shows a web browser window with the address bar displaying 'localhost:4200/teacher-rating'. The form contains the following elements:

- A label 'Choose a teacher to rate' above a dropdown menu.
- A label 'Grade (1-10)' above a numeric input field containing the value '1 - 10'.
- A label 'Leave a comment' above a text input field containing the placeholder text 'Your comment'.
- A dark grey button at the bottom with the text 'Save this rating'.

Slika 3

Ako administrator želi da se prijavi kao takav kako bi mogao da pravi novog predavača u sistemu i slično, treba da ode na URL /admin-login (slika 4). Aplikaciji se dodaje jedan sloj bezbednosti time što samo administratori znaju kako glasi putanja za formu za prijavljivanje. Trenutno je jedini administrator onaj čije je korisničko ime „admin“ i čija je lozinka takođe „admin“.



The screenshot shows a web browser window with the address bar displaying 'localhost:4200/admin-login'. The page has a dark header with the text 'Rate My Professor'. Below the header, there is a login form with two input fields: 'Username:' containing the text 'admin' and 'Password:' containing five blue dots. A 'Log In' button is located below the password field.

Slika 4

Po uspešnom prijavljivanju administrator biva preveden na formu za rad sa predavačem na URL-u „/teacher“ (slika 5), gde ima mogućnost da napravi novog, izmeni ili obriše postojećeg predavača. Svaka uspešna operacija daje poruku o tome (slika 6) i osvežava listu svih predavača novim podacima.

Rate My Professor Admin Teachers Log out

Full name  
Pera Peric

Title

Create new teacher

Show all teachers

Full name	Title		
Pera Peric	vanredni profesor	Edit	Delete
Ana Janjic	redovni profesor	Edit	Delete
Sanja Savic	docent	Edit	Delete

Slika 5

Rate My Professor Admin Teachers Log out

Full name  
Pera Peric

Title  
docent

Show all teachers

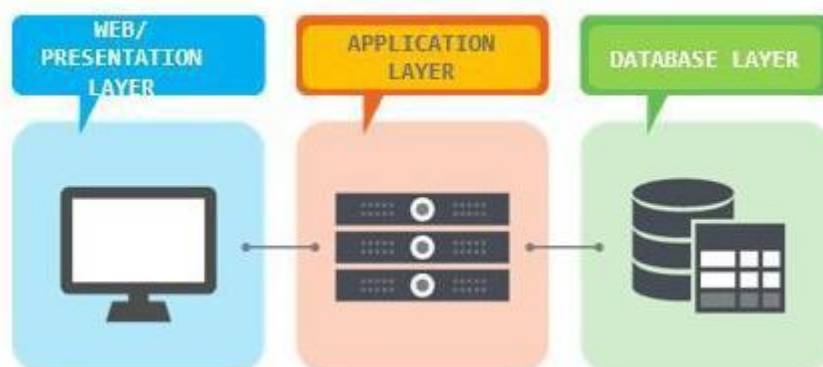
Saved.

OK

Slika 6

## 4. Implementacija

Za implementaciju verbalnog opisa napravljena je veb aplikacija troslojne arhitekture (slika 7).



Slika 7

Na prezentacionom nivou korišćen je okvir *Angular* u verziji 9. Jezik za rad sa *Angular-om* je *TypeScript*.

Napravljeno je više *Angular* komponenti koje čine ovu aplikaciju na jednoj stranici (eng. single page app), i svaka od njih komunicira sa aplikacionim slojem preko *Angular* servisa koji koriste HTTP protokol. Kada korisnik želi da uradi neku od operacija za koje je potreban administratorski pristup, on najpre mora biti ulogovan, a zatim u svakom pozivu ka aplikacionom sloju mora postaviti svoje korisničko ime u heder HTTP zahteva kako bi aplikacioni sloj dozvolio te operacije (kod 1).

Kod 1

```
; Angular funkcija koja šalje HTTP zahtev za dodavanje novog predavača, stavljajući u body podatke o predavaču, a u header korisničko ime addNewTeacher(teacher: Teacher):
Observable<any> {
    const params = new HttpHeaders().set('username',
this.adminService.getAdminUsername());
    return this.httpClient.post<any>(this.getAPIUrl('teachers'),
    {
        fullName: teacher.fullName,
        title: teacher.title }, {headers:
        params});
}
```

Na aplikacionom sloju napravljen je REST server koji podržava 7 pristupnih tačaka. Napisan je u jeziku *Clojure* koristeći njegove biblioteke *Ring* i *Compojure*, kao i *SQLKorma* za komunikaciju sa slojem podataka. U Kodu 2 dat je primer jedne pristupne tačke (eng. endpoint) koja uzima gorepomenuto korisničko ime administratora iz header-a klijentskog zahteva kako bi se u funkciji `qrs/add-teacher username` izvršila provera administratorskog statusa pre izvršenja tražene operacije.

Funkcije koje izvršavaju operacije koje klijent traži od endpoint-a nazivaju se „handler“ funkcije. Primer jedne takve je funkcija `qrs/add-teacher username`.

Kod 2

```
; Clojure endpoint za dodavanje novog predavača, koja uzima korisničko iz header-a zahteva
i podatke o novom predavaču iz body-ja
(POST "/teachers" []
  :header-params [username :- schema/Str] :body [teacher-
data qrs/NewTeacher]
  (let [{:keys [fullName title]} teacher-data]
    (ok (qrs/add-teacher username fullName title))))
```

Na sloju podataka koristi se MySQL baza podataka. U okviru tabele sa korisničkim imenom i šifrom za administratora, šifra se čuva hešovana algoritmom sha256. Server proverava da li hešovana vrednost šifre poslate u zahtevu za logovanje odgovara hešovanoj vrednosti u tabeli admins (kod 3).

Kod 3

```
; Clojure funkcija koja hešuje argument prosleđen u parametru string
(defn sha256 [string]
  (let [digest (.digest (MessageDigest/getInstance "SHA-256") (.getBytes string "UTF-8"))]
    (apply str (map (partial format "%02x") digest))))

; provera hešovane vrednosti šifre pre odobrenog prijavljivanja
(defn login-admin [username password]
  (first (select admins
    (fields :username)
    (where {:username username :password (sha256 password)}))))
```



*Clojure* server takođe nudi opciju testiranja endpoint-ova preko biblioteke Swagger, koja pravi interfejs za njihovo istraživanje prikazan na slici 8.

## Simplified Rate My Professor

Compojure API example

default	Show/Hide	List Operations	Expand Operations
GET	/api/teachers		
POST	/api/teachers		
PUT	/api/teachers		
DELETE	/api/teachers/{id}		
GET	/api/teachers/{id}		
POST	/api/teachers/rating		
POST	/api/login		

[ BASE URL: / , API VERSION: 0.0.1 ]

## 5. Korišćenje aplikacije

Aplikacija se trenutno može koristiti samo na lokalnoj mašini, prema uputstvima datim u README fajlovima u direktorijumu ratingFront za *Angular* i ratingBack za *Clojure* server.