

ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U SARAJEVU

Web aplikacija za nutricionizam i fitnes

Članovi tima:

Merima Čišija
Aida Kanlić
Adna Tahić
Adin Velić

Sarajevo, april 2017.

1. Opis teme

Ova aplikacija će pružati različite programe, kao i samu interakciju sa trenerima, zarad poboljšanja ličnog zdravlja, kao i promjena radnih navika naših klijenata. Aplikacija se sastoji iz 3 modula:

- Modul za upravljanje korisnicima
- Modul za kreiranje plana i programa za klijente i komunikaciju s trenerom
- Modul za statistiku

1.1. Upravljanje korisnicima

Korisnički interfejs omogućava jednostavnu i efikasnu komunikaciju sistema sa korisnicima. Komunikacija je ostvarena putem GUI-a, čime se pospješuje efikasnost i uspješnost obavljanja operacija.

Postoje tri vrste korisničkih računa:

- Administrator (stručno lice/nutricionista) - sa svim privilegijama nad sistemom
- Fitnes trener - sa svim privilegijama nad sistemom osim privilegija koje se tiču konfiguracije aplikacije i dodavanja računa
- Klijent - osoba sa ograničenim pristupom

Različite privilegije za ove tri vrste korisničkih računa zahtjevaju različite interfejse samog sistema, u skladu sa funkcionalnostima koje su omogućene sa dodijeljenim pravima.

1.1.1 Korisnički interfejs za administratora

Korisnički interfejs za korisnike sa administratorskim privilegijama treba omogućiti pristup svim funkcionalnim zahtjevima koje sistem podržava. Sljedeće grupe funkcionalnih zahtjeva su omogućene računu sa administratorskim pravom:

- Upravljanje postavkama aplikacije
- Upravljanje svim korisničkim računima
- Pristup podacima korisnika usluga radi praćenja napretka
- Logovanje na sistem

1.1.2 Korisnički interfejs za fitnes trenera

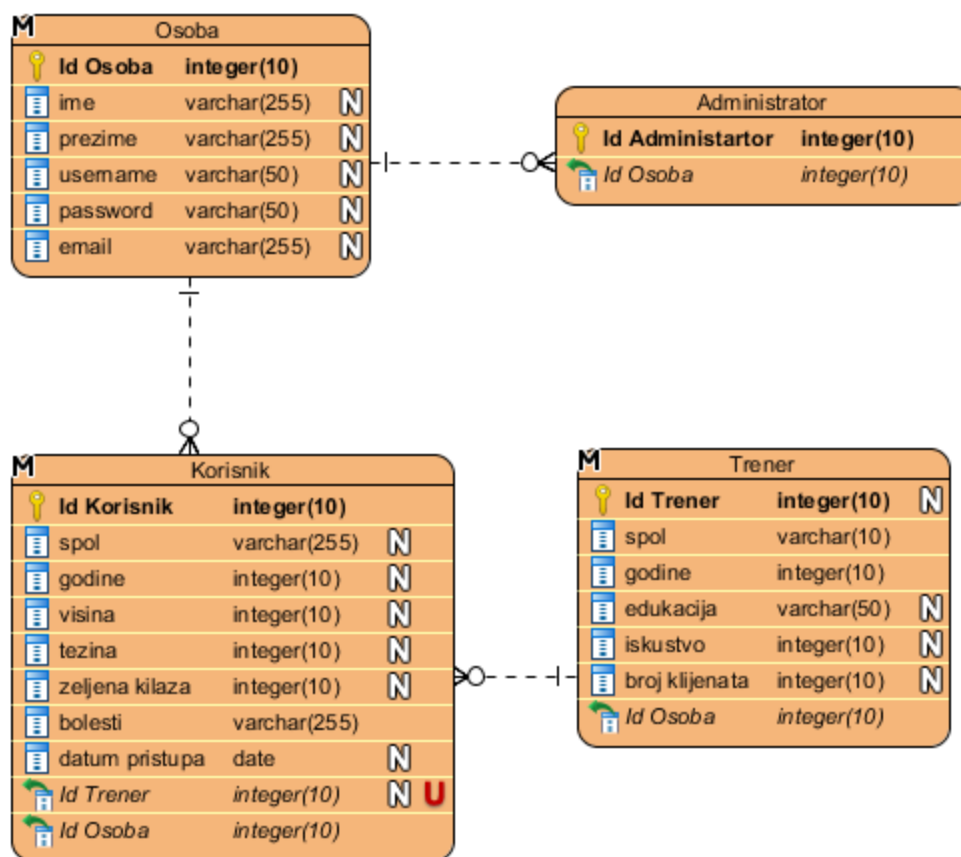
Korisnički interfejs za korisnike sa privilegijama supervizora ima pristup svim podacima i funkcionalnostima sistema osim onih koji bi na bilo koji način mogli narušiti željeni rad aplikacije:

- Mogućnost modifikovanja/dodavanja/brisanja fitnes ponuda i ponuda ishrane
- Pristup podacima korisnika radi praćenja napretka
- Logovanje na sistem

1.1.3 Korisnički interfejs klijenta

Korisnički interfejs za korisnike ima ograničen pristup funkcionalnim zahtjevima koje sistem pruža. Sljedeće grupe funkcionalnih zahtjeva su omogućene:

- Upravljanje sopstvenim korisničkim računom
- Pristup ponudama nutricioniste
- Praćenje vlastitog rada
- Komunikacija sa trenerom
- Logovanje na sistem



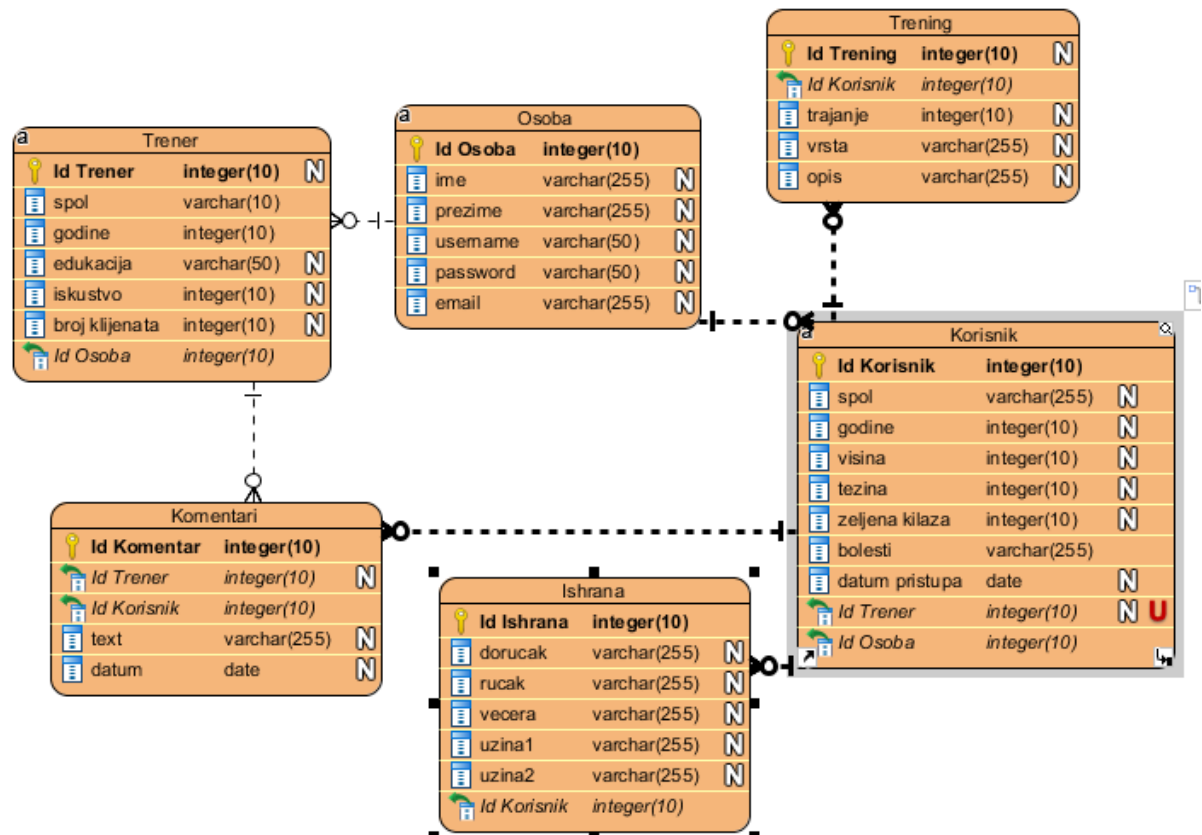
Slika 1: ERD za modul za upravljanje korisnicima

1.2 Modul za kreiranje plana i programa za klijente i komunikaciju s trenerom

Ovaj modul klijentu omogućava da unese svoje podatke (lične podatke, visina, težina, dosadašnju aktivnost, bolesti). Nakon toga na osnovu statističkih podataka koji će biti prikazani klijentu, on vrši odabir trenera kojeg želi, a koji je trenutno slobodan. Zatim, trener na osnovu unesenih podataka o klijentu, klijentu preporučuje plan i program kojeg se treba pridržavati, koje vježbe treba raditi i sl. Pored toga klijent i trener mogu komunicirati preko forme putem poruka koje šalju jedan drugome.

Time su omogućene sljedeće funkcionalnosti:

- Unos podataka o klijentima
- Odabir trenera od strane klijenta
- Prijedlog plana i programa klijentu od strane trenera
- Komunikacija klijenta i trenera



Slika 2: ERD za modul za kreiranje plana i programa za klijente i komunikaciju s trenerom

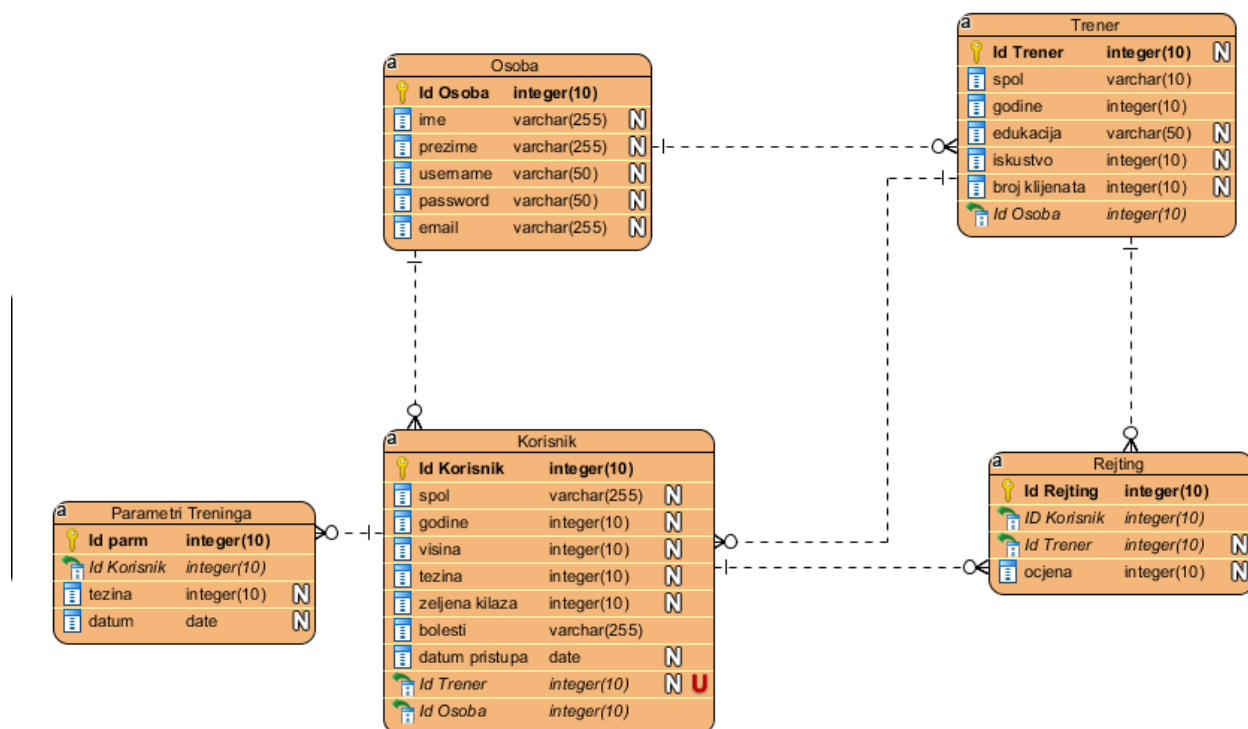
1.3 Modul za statistiku

Na osnovu podataka iz baze, prethodno unesenih kroz korištenje aplikacije, prikazuju se statistika prema određenim kriterijima. Korisnik bira kriterij prema kojem se kreira statistika. Postoji mogućnost izbora nekog od sljedećih: zauzetost trenera, odnos muškaraca i žena po trenerima, uspješnost programa.

Zauzetost trenera je kriterij koji će prezentirati procentualno prosječni udio korisnika sa kojima je radio svaki od trenera u toku jedne godine. Ova statistika omogućava budućim korisnicima da saznaju traženost svakog od trenera, kao potencijalnu mjeru kvalitete rada, te im pomogne u eventualnom odabiru vlastitog trenera. Važno je spomenuti da jedan trener u datom trenutku može raditi sa maksimalno 10 korisnika.

Odnos muškaraca i žena po trenerima kriterij koji prikazuje procentualnu podjelu korisnika po spolu i to za svakog trenera. Cilj ove statistike je dodatna pomoć pri odabiru trenera, obzirom da je moguće da je rad pojedinih trenera više orjentisan ka potrebama muškaraca ili ka zahtjevima žena.

Uspješnost programa prikazuje zadovoljstvo klijenata onako kako su oni to ocijenili ocjenama od 1 do 5. Mjera uspješnost je vrlo bitan faktor za privlačenje novih klijenata i za unapređenje rada organizacije.



Slika 3: ERD za modul za statistiku

2. Komunikacija između modula

Modul za upravljanje korisnicima je jedini modul koji može samostalno funkcionisati, a od kojeg zavise svi ostali moduli. Naime, u ovom modulu se vrši pravljenje korisničkih računa, klijenata i trenera te kao takav može bez problema funkcionisati kao zasebna cjelina.

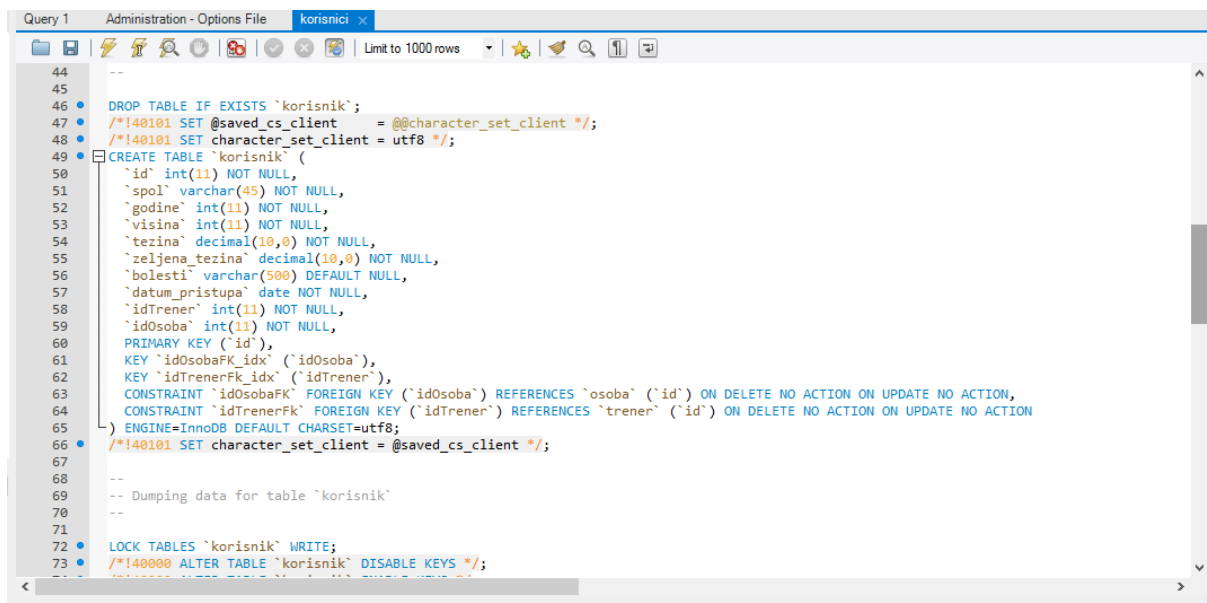
Modul za kreiranje plana i programa za klijente i komunikaciju s trenerom je u zavisnosti sa modulom za upravljanje korisnicima jer treba podatke o korisnicima i trenerima, kao i sa modulom za statistiku jer se na osnovu statističkih podataka vrši odabir trenera u ovom modulu.

Modul za statistiku zavisi od modula za upravljanje korisnicima jer treba podatke o trenerima i klijentima.

3. Implementirane funkcionalnosti sistema

3.1. Task 1

Na početku su kreirana tri odvojena projekta „KorisniciModulu“, „PlaniProgramModul“ i StatistikaModul“. Napravljene su i tri odvojene baze za svaki od ovih modula. Dio skripte za bazu koja pored tabele „korisnici“ ima jos tabele: administrator, osoba, trener.



```
44 --
45
46 DROP TABLE IF EXISTS `korisnik`;
47 /*!40101 SET @saved_cs_client = @@character_set_client */;
48 /*!40101 SET character_set_client = utf8 */;
49 CREATE TABLE `korisnik` (
50   `id` int(11) NOT NULL,
51   `spol` varchar(45) NOT NULL,
52   `godine` int(11) NOT NULL,
53   `visina` int(11) NOT NULL,
54   `tezina` decimal(10,0) NOT NULL,
55   `zeljena_tezina` decimal(10,0) NOT NULL,
56   `bolesti` varchar(500) DEFAULT NULL,
57   `datum_pristupa` date NOT NULL,
58   `idTrener` int(11) NOT NULL,
59   `idOsoba` int(11) NOT NULL,
60   PRIMARY KEY (`id`),
61   KEY `idOsobaFK_idx` (`idOsoba`),
62   KEY `idTrenerFk_idx` (`idTrener`),
63   CONSTRAINT `idOsobaFK` FOREIGN KEY (`idOsoba`) REFERENCES `osoba` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
64   CONSTRAINT `idTrenerFk` FOREIGN KEY (`idTrener`) REFERENCES `trener` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
65 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
66 /*!40101 SET character_set_client = @saved_cs_client */;
67
68 --
69 -- Dumping data for table `korisnik`
70 --
71
72 LOCK TABLES `korisnik` WRITE;
73 /*!40000 ALTER TABLE `korisnik` DISABLE KEYS */;
```

Slika 4: Baza mkorisnici:tabela korisnici

```

19 -- Table structure for table `ishrana`
20 --
21
22 DROP TABLE IF EXISTS `ishrana`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!40101 SET character_set_client = utf8 */;
25 CREATE TABLE `ishrana` (
26   `id` int(11) NOT NULL,
27   `dorucak` varchar(255) NOT NULL,
28   `rucak` varchar(255) NOT NULL,
29   `vecera` varchar(255) NOT NULL,
30   `uzina1` varchar(255) NOT NULL,
31   `uzina2` varchar(255) NOT NULL,
32   `id_korisnik` int(11) DEFAULT NULL,
33   PRIMARY KEY (`id`),
34   KEY `fk_korisnik3_idx` (`id_korisnik`),
35   CONSTRAINT `fk_korisnik3` FOREIGN KEY (`id_korisnik`) REFERENCES `korisnik` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
37 /*!40101 SET character_set_client = @saved_cs_client */;
38
39 -- Dumping data for table `ishrana`
40 --
41
42
43 LOCK TABLES `ishrana` WRITE;
44 /*!40000 ALTER TABLE `ishrana` DISABLE KEYS */;
45 /*!40000 ALTER TABLE `ishrana` ENABLE KEYS */;
46 UNLOCK TABLES;
47
48 --

```

Slika 5: Baza mplaniprogram: tabela ishrana

Pored nje u modulu „Plan i program“ su još table: komentari, korisnik, osoba, trener, terening.

```

76 /*!40000 ALTER TABLE `osoba` DISABLE KEYS */;
77 /*!40000 ALTER TABLE `osoba` ENABLE KEYS */;
78 UNLOCK TABLES;
79
80 --
81 -- Table structure for table `parametritreninga`
82 --
83
84 DROP TABLE IF EXISTS `parametritreninga`;
85 /*!40101 SET @saved_cs_client      = @@character_set_client */;
86 /*!40101 SET character_set_client = utf8 */;
87 CREATE TABLE `parametritreninga` (
88   `id` int(11) NOT NULL,
89   `id_korisnika` int(11) NOT NULL,
90   `tezina` decimal(10,0) NOT NULL,
91   `datum` date NOT NULL,
92   PRIMARY KEY (`id`),
93   KEY `fk_korisnik1_idx` (`id_korisnika`),
94   CONSTRAINT `fk_korisnik1` FOREIGN KEY (`id_korisnika`) REFERENCES `korisnik` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
95 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
96 /*!40101 SET character_set_client = @saved_cs_client */;
97
98 -- Dumping data for table `parametritreninga`
99 --
100
101
102 LOCK TABLES `parametritreninga` WRITE;
103 /*!40000 ALTER TABLE `parametritreninga` DISABLE KEYS */;
104 /*!40000 ALTER TABLE `parametritreninga` ENABLE KEYS */;
105 UNLOCK TABLES;

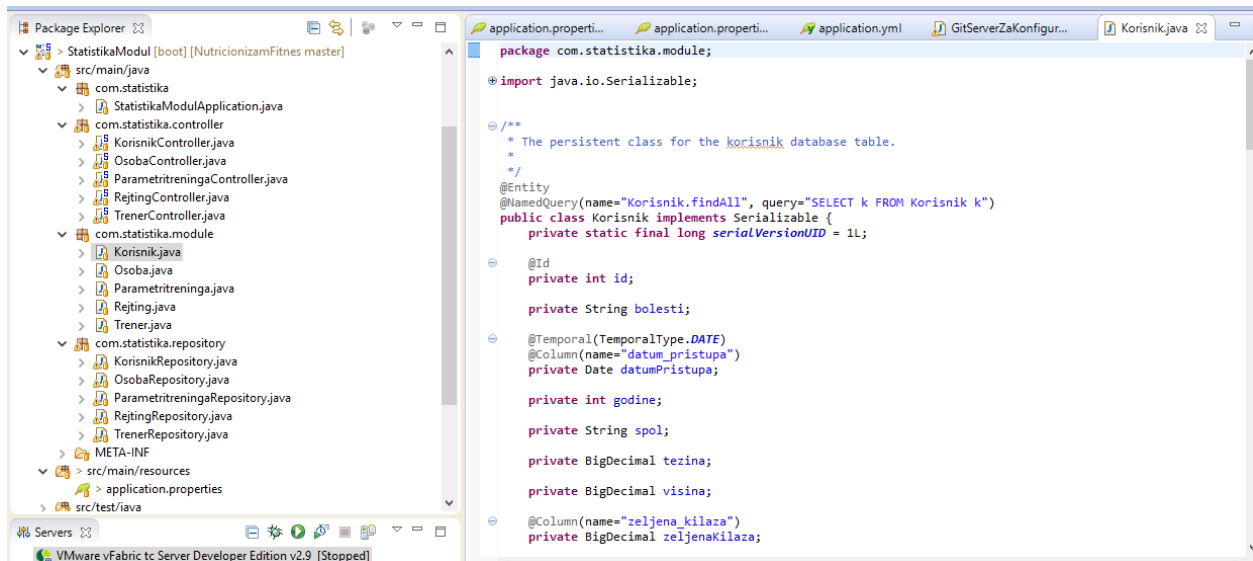
```

Sliak 6: Baza mstatistika: tabela parametri treninga

Pored nje u modulu „Statistika“ su još: korisnici, osoba, rejting, trener.

Nakon toga baze su napunjene podacima kako bi se mogle testirati funkcionalnosti aplikacije. U svakoj tabeli ima barem 4-5 redova.

Na osnovu tabela u bazi podataka kreirane su klase sa anotacijom `@Entity`. Na Slici 7 je prikazana klasa „Korisnici“ koja je kreirana na osnovu tabele „Korisnici“ u bazi podataka.

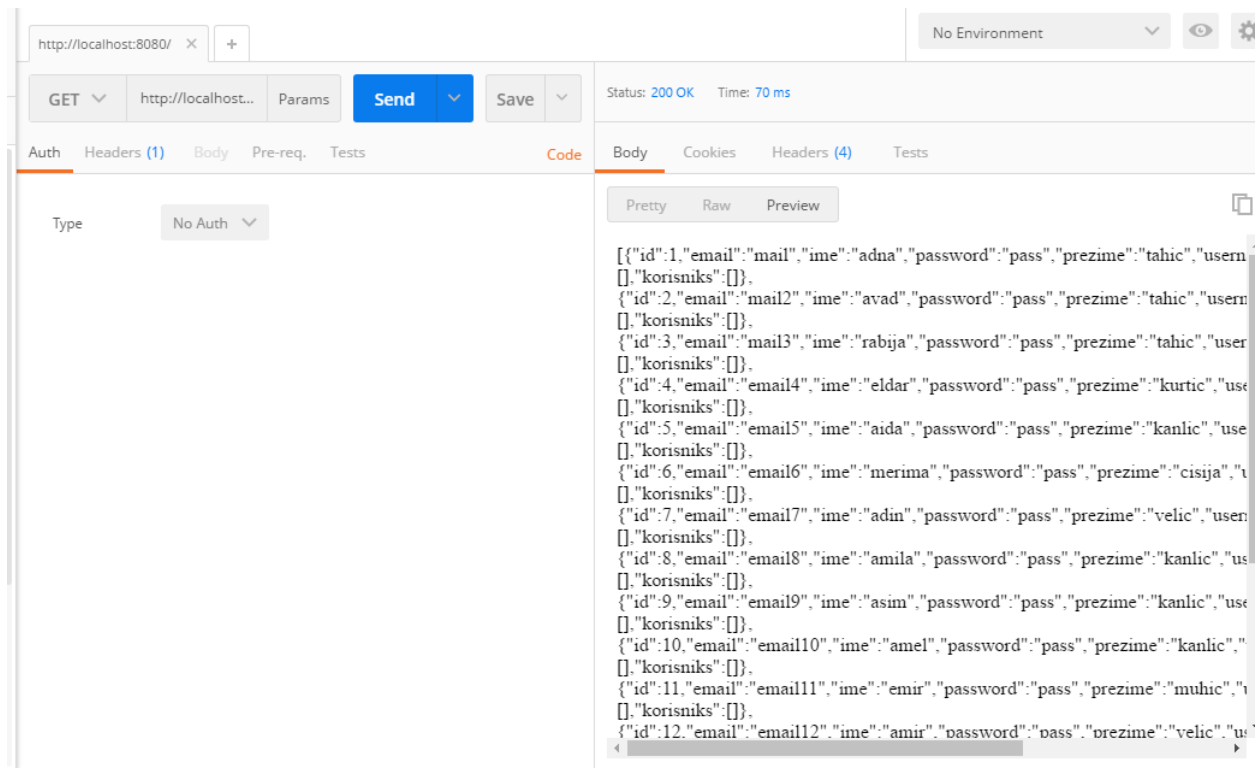


Slika 7: Tabela „Korisnici“ u „StatistikaModul“

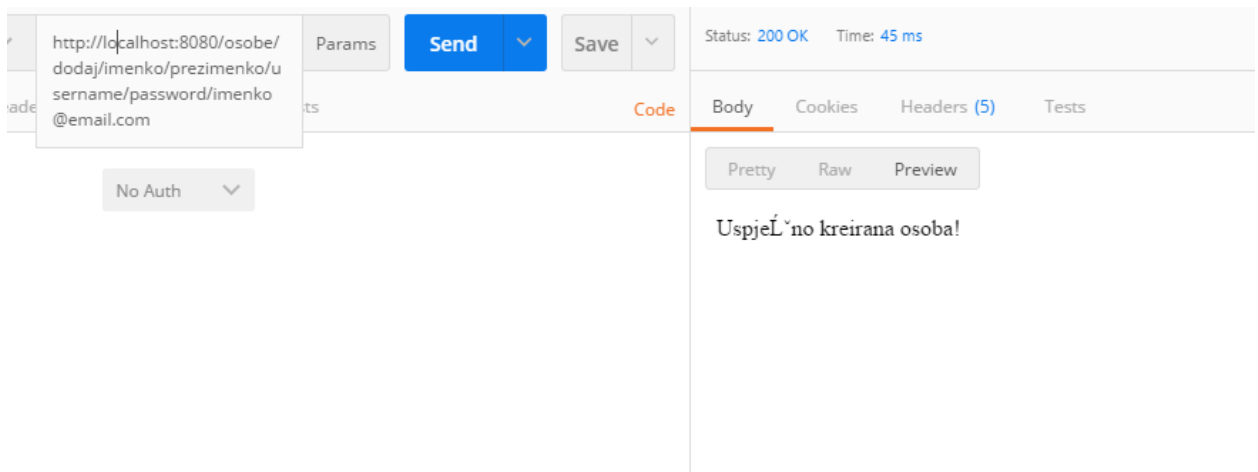
Analogno je urađeno za sve tabele u svim bazama.

Potom su napravljeni repozitoriji i u njima CRUD REST metode.

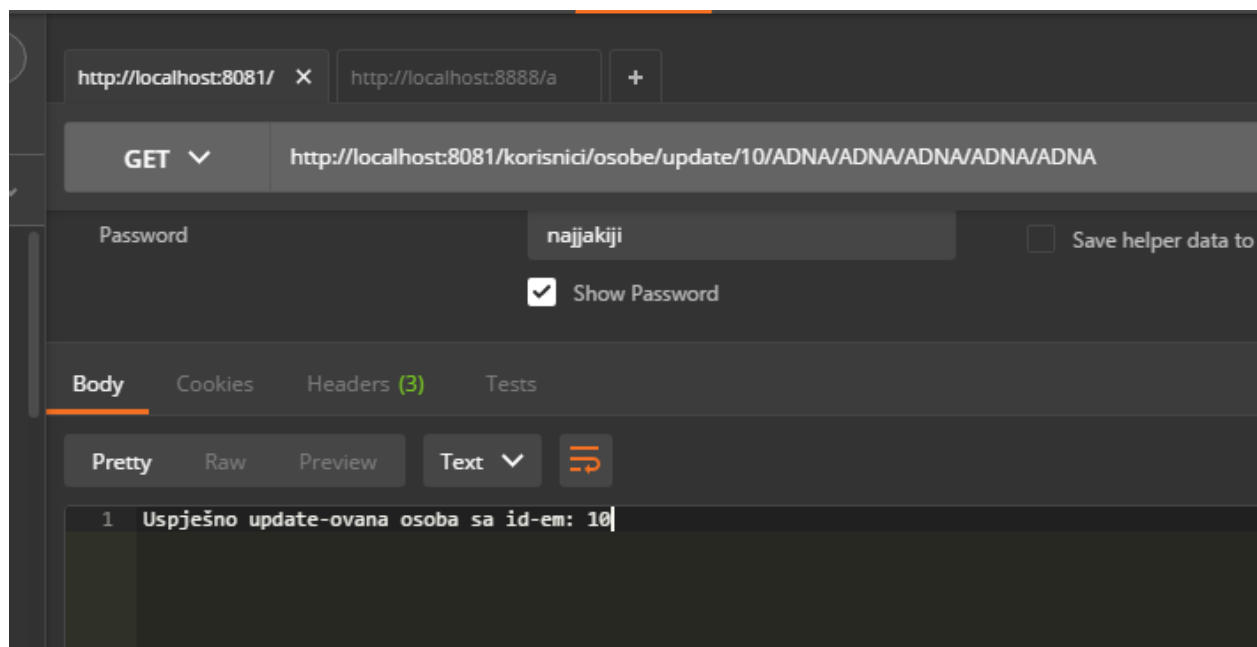
U Postmanu smo preko url-a: <http://localhost:8080/osobe/svi> izlistali sve osobe koje su bile u bazi. To je prikazano na Slici 8.



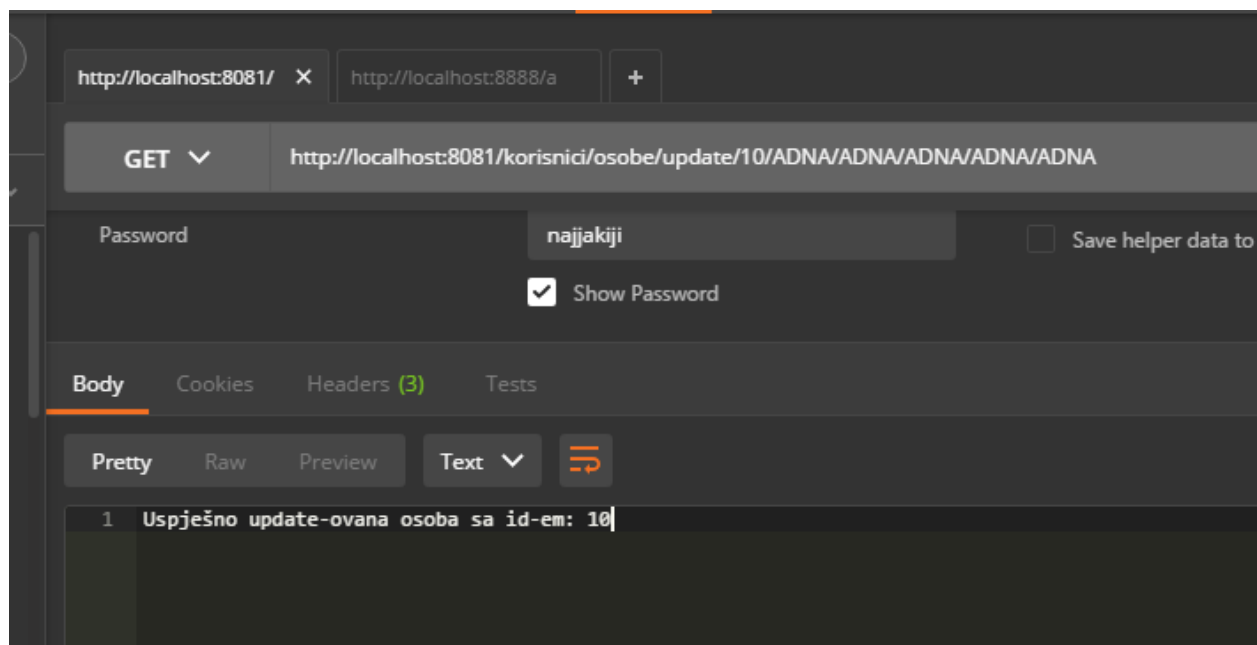
Slika 8: GET zahtjev za svim osobama u bazi



Slika 9: POST metod za dodavanje osobe



Slika 10: Brisanje korisnika s ID-em 10



Slika 11: Update osobe sa ID-em 10

3.2. Task 2

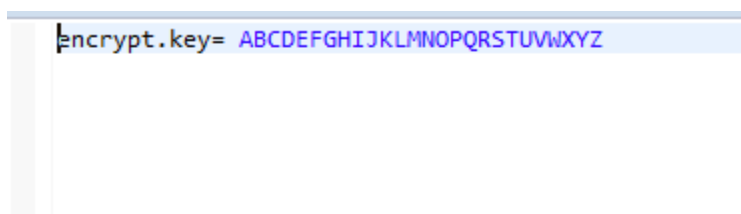
Korištenjem Spring Configuration Servera je napravljena centralizovana konfiguracija. Urađena je osnovna zaštita putem autentifikacije i enkripcija konfiguracije.

A screenshot of a code editor showing the content of an application.yml file. The file is a YAML configuration for a Spring application. It includes settings for the server port, security (basic authentication), and Spring Cloud configuration (Git repository, search paths).

```
---
server:
  port: 8888
security:
  basic:
    enabled: true
    user:
      name: najjakiji
      password: najjakiji
spring:
  cloud:
    config:
      server:
        git:
          uri: https://github.com/adnatahic/NutricionizamFitnes.git
          #username: uname
          #password: pass
          search-paths:
            - 'Konfiguracija*'
```

Slika 12: application.yml

Na Slici 12 je prikazana autentifikacija i centralizacija konfiguracije.

A screenshot of a code editor showing the content of a bootstrap.properties file. The file contains a single line of configuration for the encrypt key.

```
encrypt.key= ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Slika 13: Bootstrap.properties

Na Slici 13 enkripcija konfiguracije.

3.3. Task 3

Napravljen je Eureka server i tri klijenta(tri modula) sto je prikazano na Slici 12.

Environment	test	Current time	2017-04-13T13:20:39 +0200
Data center	default	Uptime	00:15
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	12
DS Replicas			
localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
KORISNICIMODUL	n/a (1)	(1)	UP (1) - Merima-PC:korisnicimodul:8084
PLANIPROGRAMMODUL	n/a (1)	(1)	UP (1) - Merima-PC:planiprogrammodul:8085
STATISTIKAMODUL	n/a (1)	(1)	UP (1) - Merima-PC:statistikamodul:8086
General Info			
Name	Value		
total-avail-memory	59mb		

Slika 14: Eureka server i klijenti: KorisniciModul, PlanIProgramModul i StatistikaModul

Urađena je komunikacije između mikroservisa preko RestTemplate-a.