

**Seminarski rad iz predmeta OOAD i OOP**

Fitness aplikacija

|  |  |
| --- | --- |
| **Studenti:** |  |
| Purić Ilija, PIN055/19 |  |
|  |  |
| Novi Sad*, 16.6.2021* | |

Sadržaj

[1. Opis procesa 1](#_Toc74756527)

[2. Dijagram slučajeva upotrebe 2](#_Toc74756528)

[3. Opis izabranih slučajeva upotrebe 3](#_Toc74756529)

[4. Nacrt korisničkog interfejsa 5](#_Toc74756530)

[5. Dijagram sekvenci 7](#_Toc74756531)

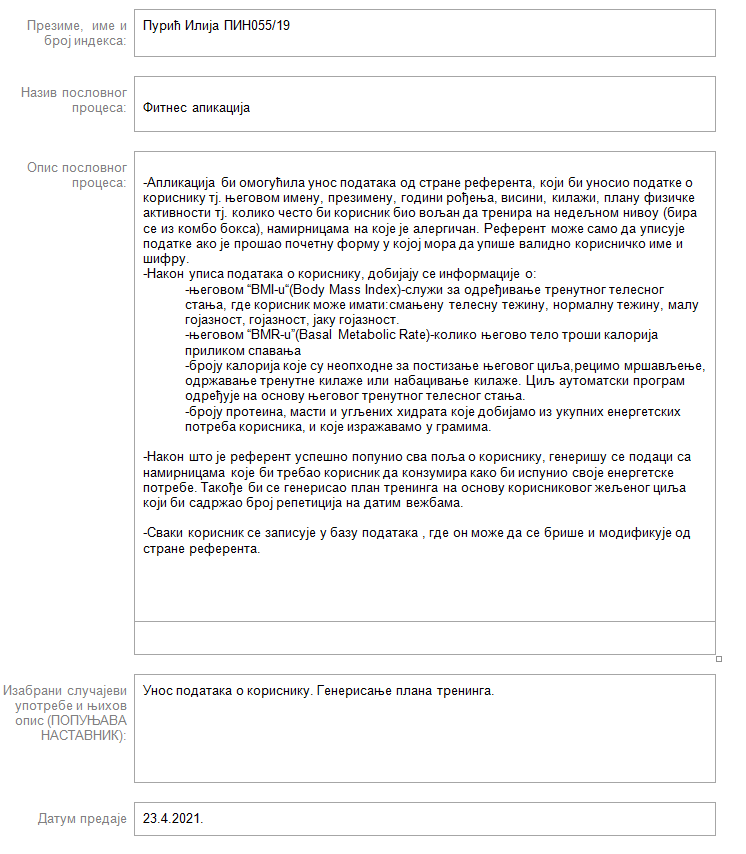
[6. CRC kartice 9](#_Toc74756532)

[7. Dijagrami klasa 11](#_Toc74756533)

[8. Generisani programski kod 12](#_Toc74756534)

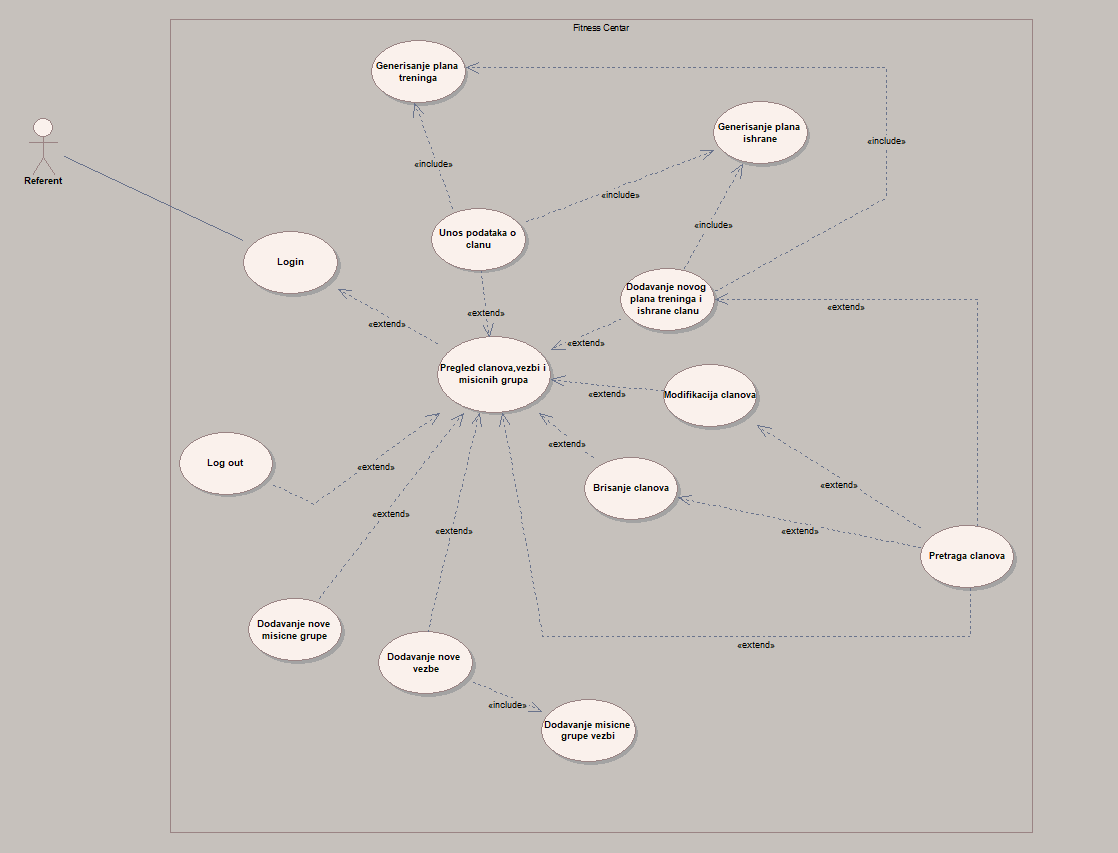
[9. Prikaz i opis gotove aplikacije 17](#_Toc74756535)

# Opis procesa



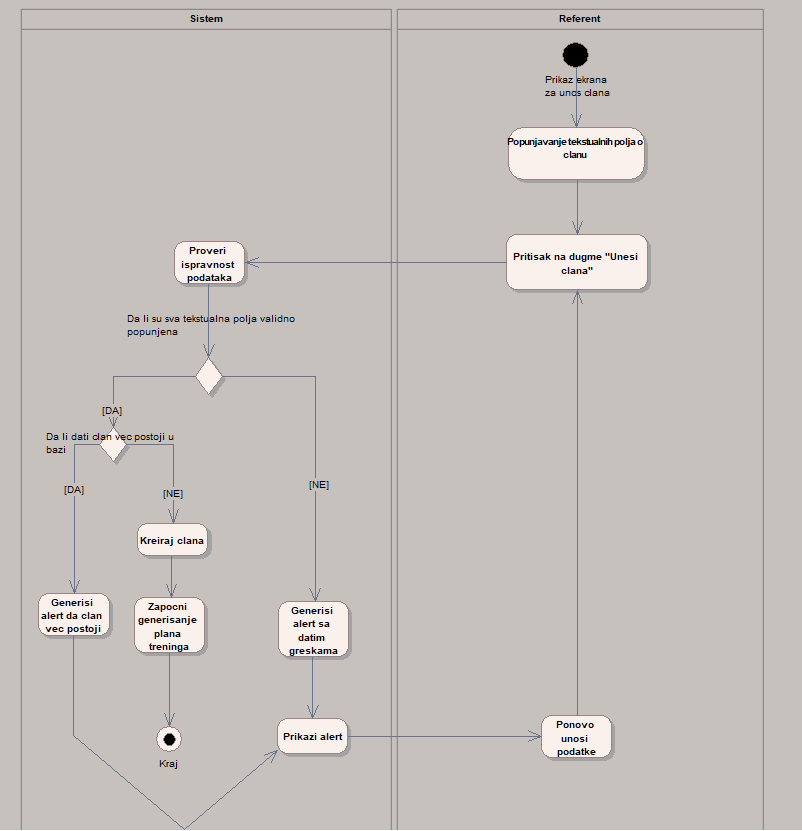
**Slika 1**.Opis procesa

# Dijagram slučajeva upotrebe



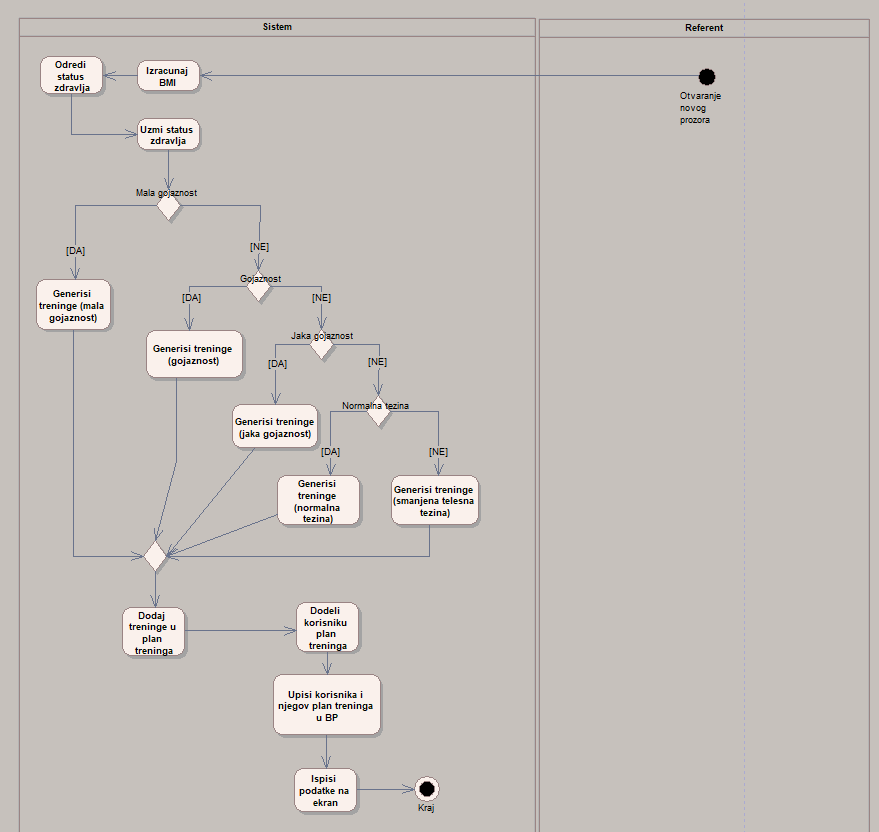
**Slika 2**.Dijagram slučajeva upotrebe

# Opis izabranih slučajeva upotrebe



**Slika 3.**Dijagram aktivnosti za slučaj upotrebe „Unos podataka o članu”

Referent dobija prikaz ekrana za unos člana, gde on unosi podatke o njemu. Podaci moraju biti popunjeni ispravno, kako bi korisnik bio upisan. Ukoliko su podaci pogrešno upisani generiše se alert sa greškama. Isto se alert može generisati u slučaju da se upisuje korisnik koji je već upisan u bazu podataka. U oba slučaja će se prikazati alert sa greškama/greškom i referent će ponovo unositi podatke i vršiće se opet ciklus provere. U drugom slučaju ako su sva polja validno popunjena, i ako ne postoji član upisan u bazi kreira se član i započinje se generisanje plana treninga.



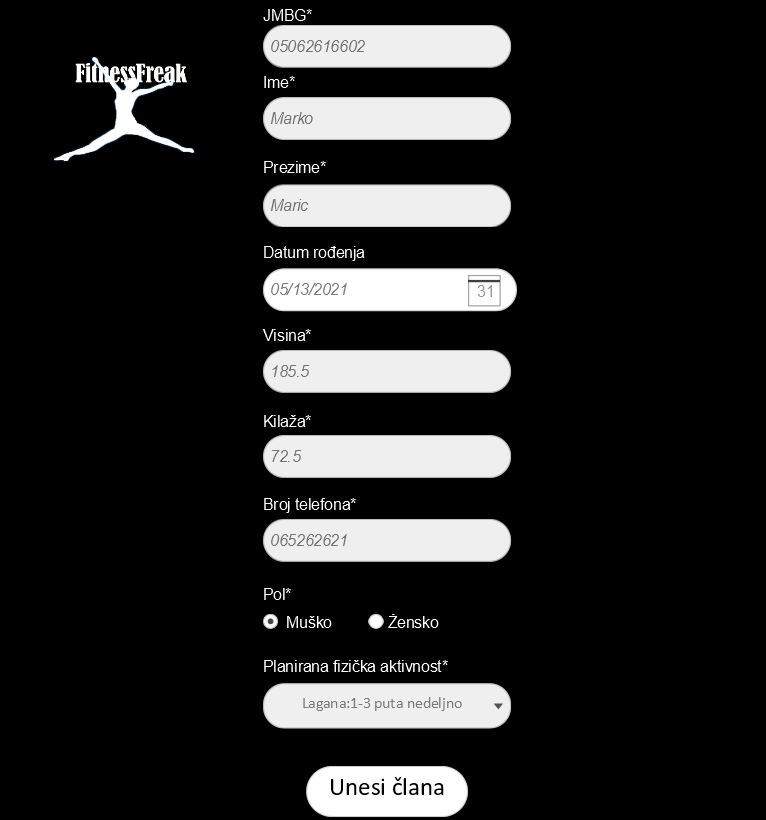
**Slika 4**. Dijagram aktivnosti za slučaj upotrebe „Generisanje plana treninga“

Nakon uspešnog unosa člana mora da se generiše plan treninga za njega. Plan treninga se generiše računanjem „BMI-a“ člana. Naspram njegovog „BMI-a“ možemo da odredimo status zdravlja koji može imati sledeće atribute:

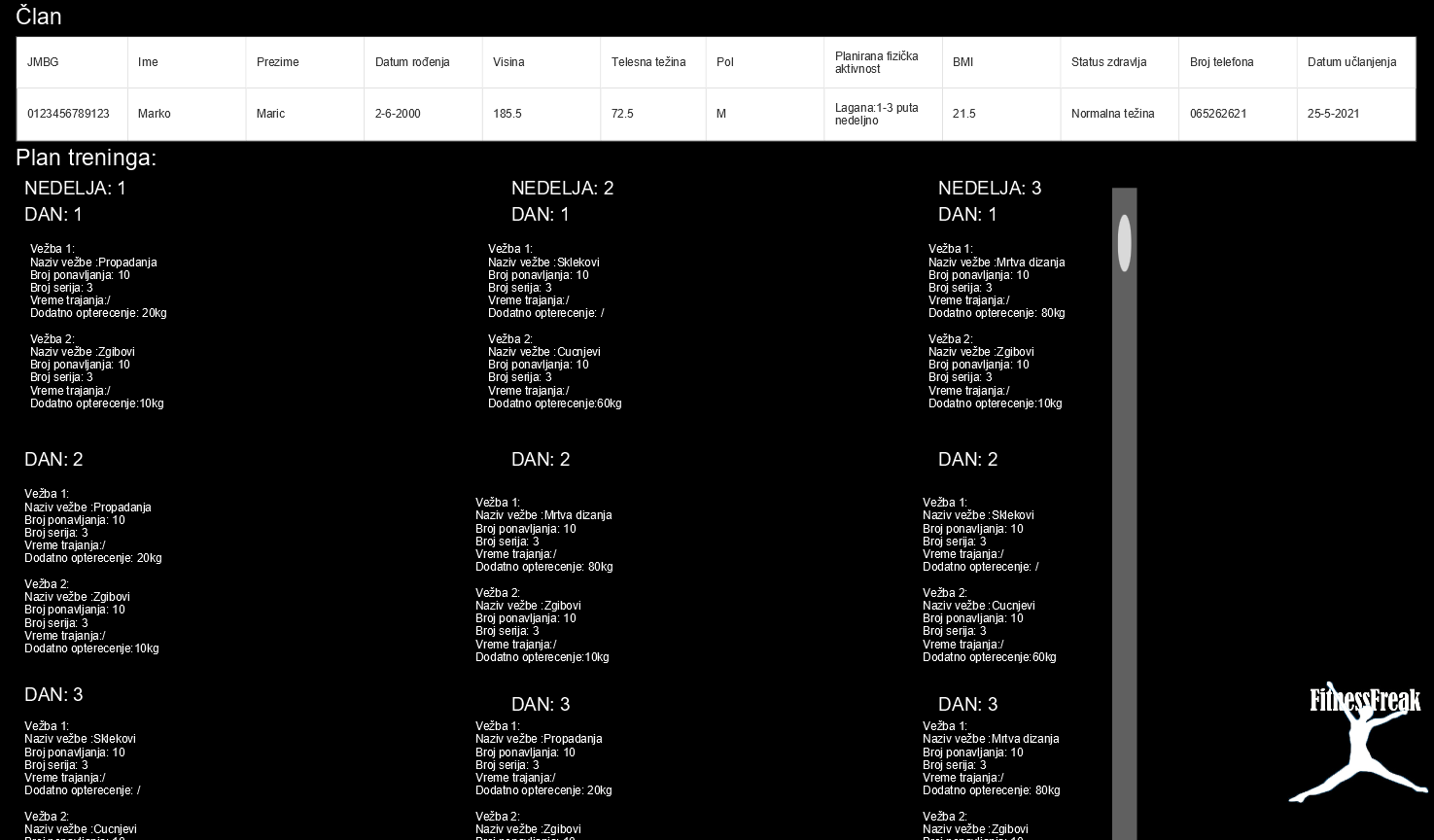
* Mala gojaznost
* Gojaznost
* Jaka gojaznost
* Normalna težina
* Smanjena telesna težina

Status zdravlja nam utiče na formiranje plana treninga , u smislu određivanja vežbi koje će biti dodeljene treningu, njihovog broja ponavljanja, broja serija, vremenskog trajanja i kilaže za dizanje u slučaju da je vežba ima. Nakon što je generisan plan treninga članu se isti dodeljuje. Za kraj se član upisuje u bazu podataka i njegovi podaci i plan treninga se ispisuje na ekran.

# Nacrt korisničkog interfejsa

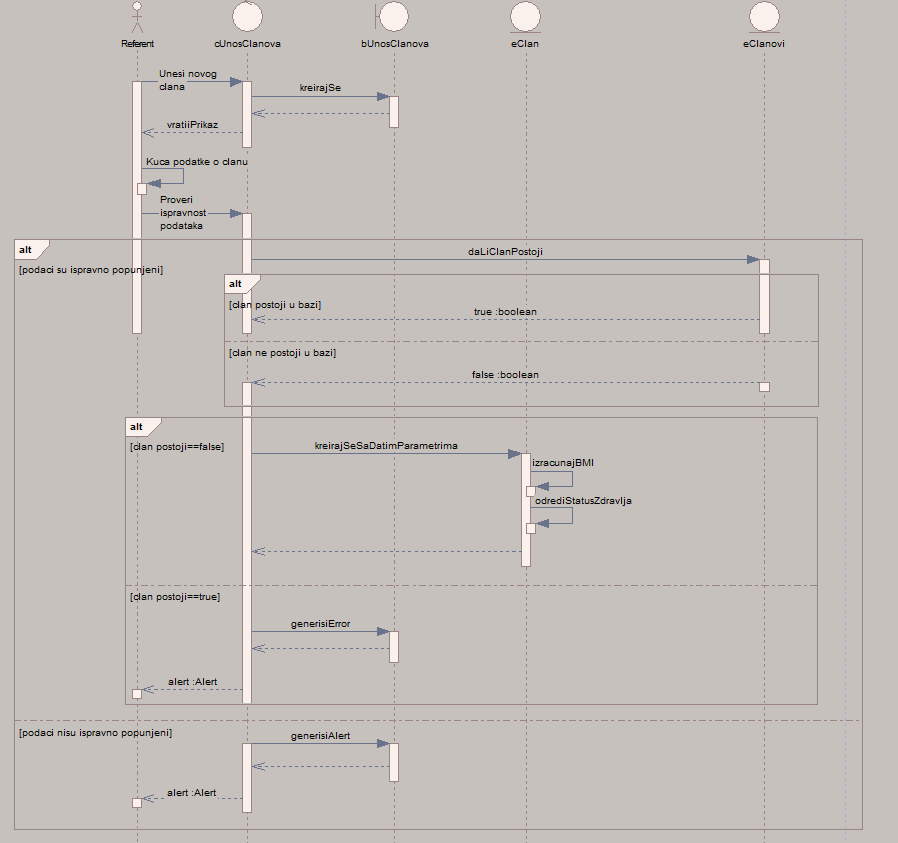


**Slika 5**. Nacrt korisničkog interfejsa za slučaj upotrebe “Unos podataka o korisniku”

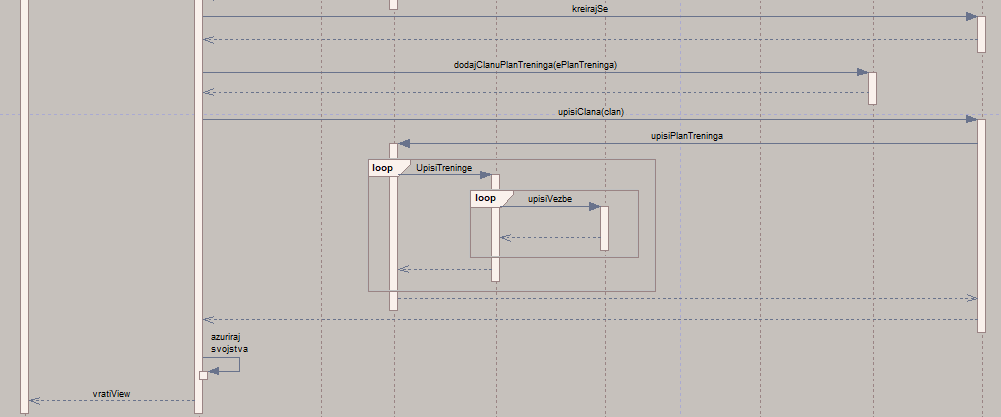
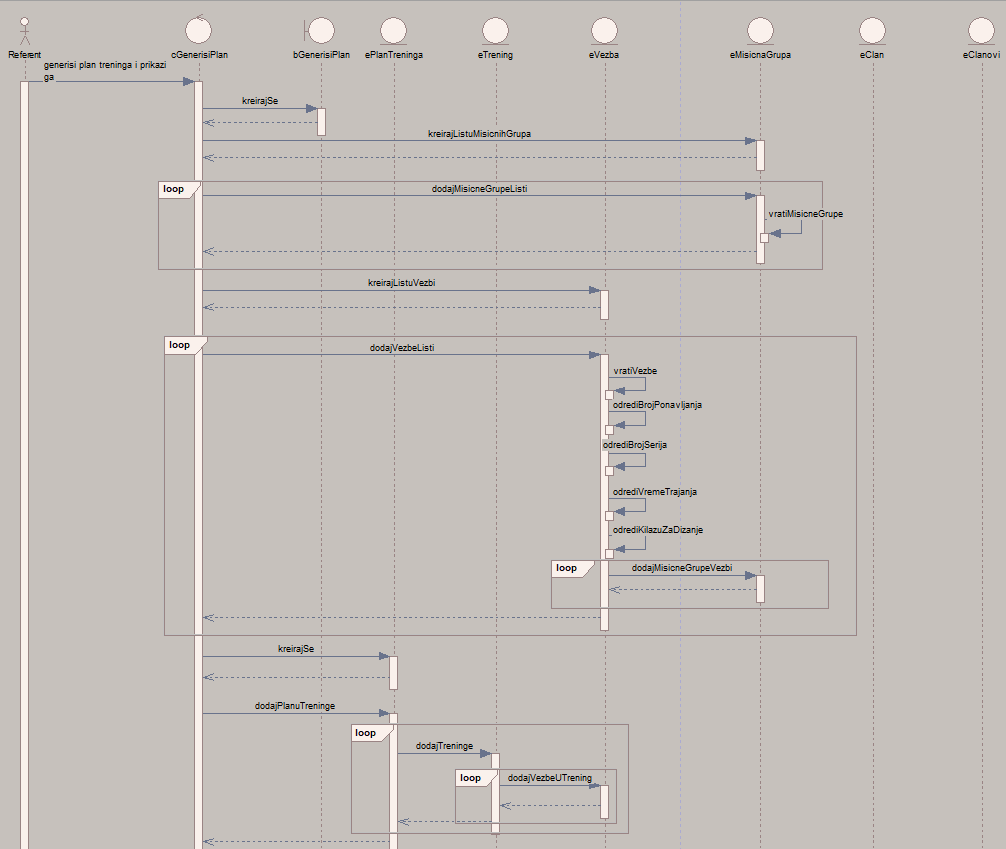


**Slika 6**. Nacrt korisničkog interfejsa „Generisanje plana treninga“

# Dijagram sekvenci

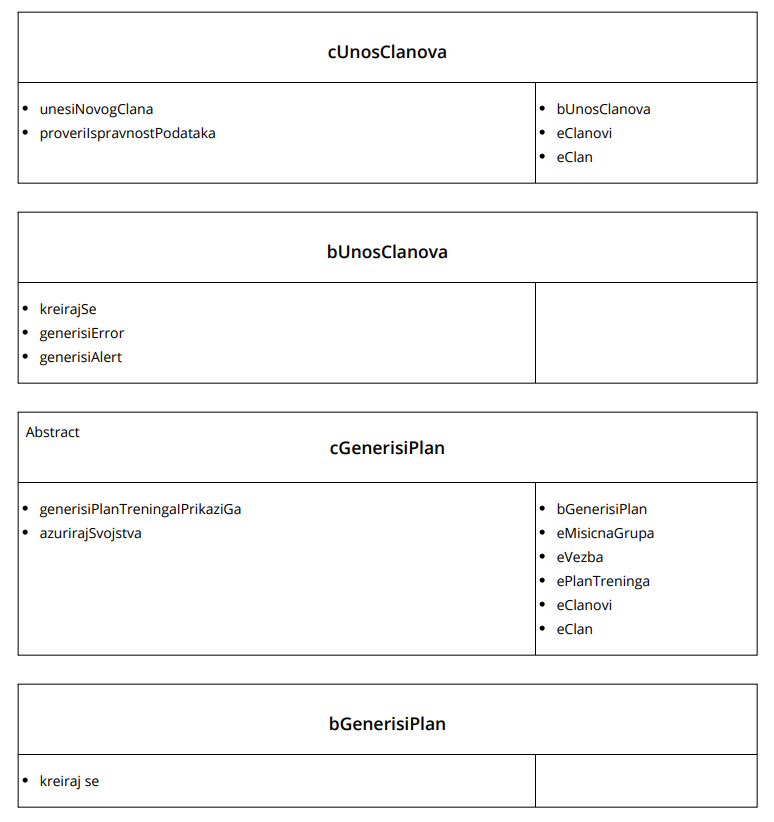


**Slika 7**. Dijagram sekvenci za slučaj upotrebe “Unos podataka o članu’

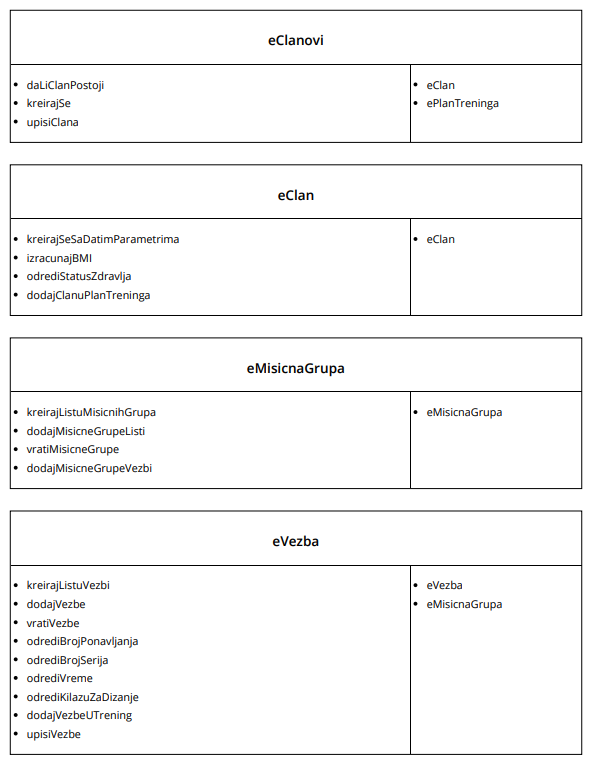


**Slika 8, 9.** Dijagram sekvenci za slučaj upotrebe „Generisanje plana treninga”

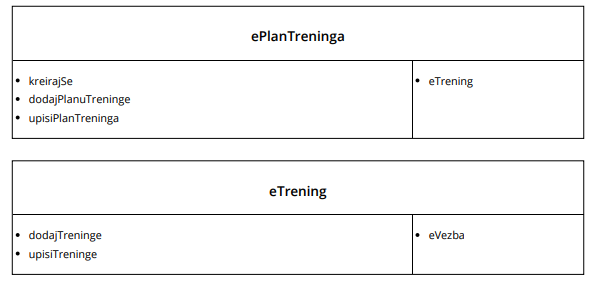
# CRC kartice



**Slika 10**. CRC kartice

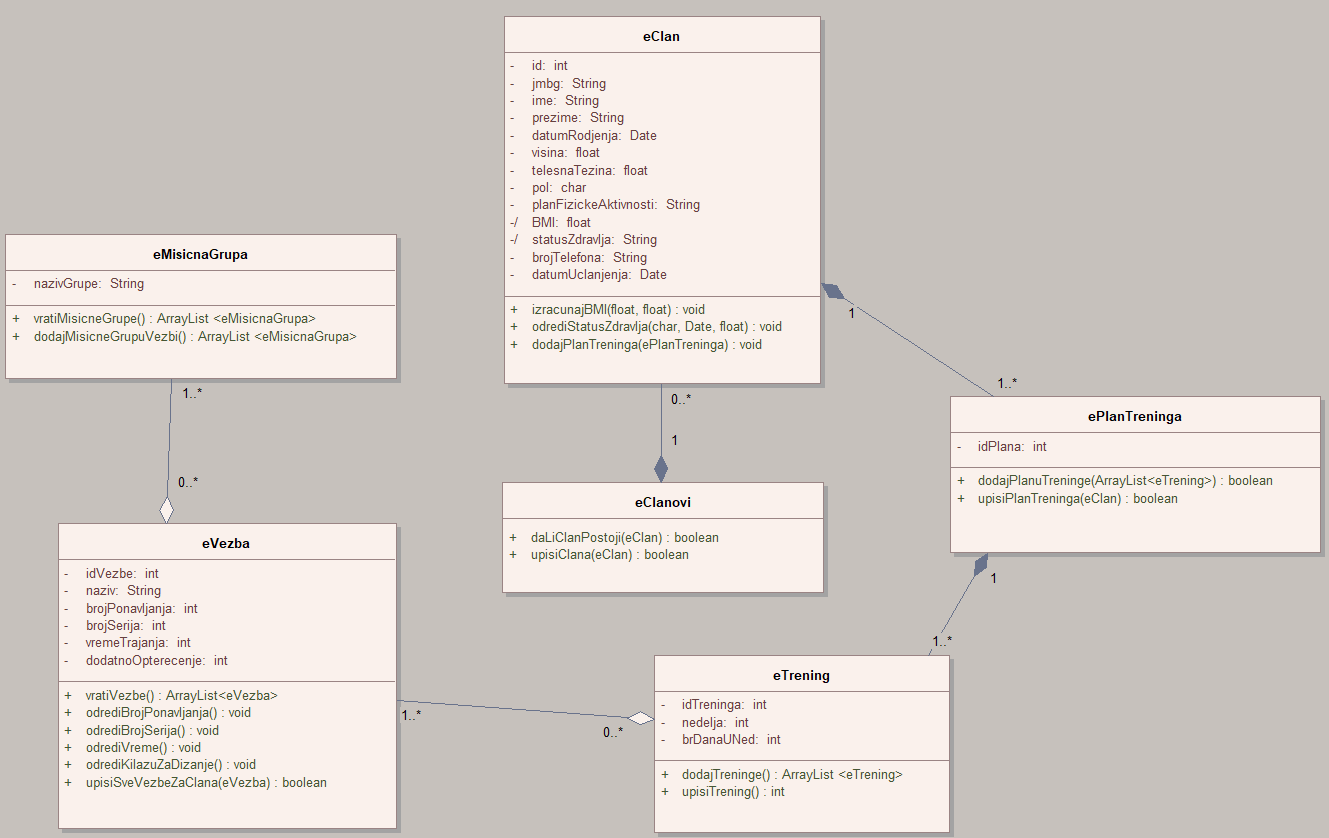


**Slika 11.** CRC kartice



**Slika 12.** CRC kartice

# Dijagrami klasa



**Slika 13**. Dijagram klasa

# Generisani programski kod

public class eClanovi {

public eClan m\_eClan;

public eClanovi(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param clan

\*/

public boolean daLiClanPostoji(eClan clan){

return false;

}

/\*\*

\*

\* @param clan

\*/

public boolean upisiClana(eClan clan){

return false;

}

}

public class eClan {

private int id;

private String jmbg;

private String ime;

private String prezime;

private Date datumRodjenja;

private float visina;

private float telesnaTezina;

private char pol;

private String planFizickeAktivnosti;

private float BMI;

private String statusZdravlja;

private String brojTelefona;

private Date datumUclanjenja;

public ePlanTreninga m\_ePlanTreninga;

public eClan(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param tezina

\* @param visina visina

\*/

public void izracunajBMI(float tezina, float visina){

}

/\*\*

\*

\* @param pol

\* @param datumRodjenja

\* @param BMI BMI

\*/

public void odrediStatusZdravlja(char pol, Date datumRodjenja, float BMI){

}

/\*\*

\*

\* @param ePlanTreninga

\*/

public void dodajPlanTreninga(ePlanTreninga ePlanTreninga){

}

}

public class ePlanTreninga {

private int idPlana;

public eTrening m\_eTrening;

public ePlanTreninga(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param trening

\*/

public boolean dodajPlanuTreninge(ArrayList<eTrening> trening){

}

/\*\*

\*

\* @param clan

\*/

public boolean upisiPlanTreninga(eClan clan){

return false;

}

}

public class eTrening {

private int idTreninga;

private int nedelja;

private int brDanaUNed;

public eVezba m\_eVezba;

public eTrening(){

}

public void finalize() throws Throwable {

}

public ArrayList <eTrening> dodajTreninge(){

return null;

}

public int upisiTrening(){

return 0;

}

}

public class eVezba {

private int idVezbe;

private String naziv;

private int brojPonavljanja;

private int brojSerija;

private int vremeTrajanja;

private int dodatnoOpterecenje;

public eMisicnaGrupa m\_eMisicnaGrupa;

public eVezba(){

}

public void finalize() throws Throwable {

}

public ArrayList<eVezba> vratiVezbe(){

return null;

}

public void odrediVreme(){

}

public void odrediBrojPonavljanja(){

}

public void odrediBrojSerija(){

}

public void odrediKilazuZaDizanje(){

}

/\*\*

\*

\* @param vezba

\*/

public boolean upisiSveVezbeZaClana(eVezba vezba){

return false;

}

}

public class eMisicnaGrupa {

private String nazivGrupe;

public eMisicnaGrupa(){

}

public void finalize() throws Throwable {

}

public ArrayList <eMisicnaGrupa> vratiMisicneGrupe(){

return null;

}

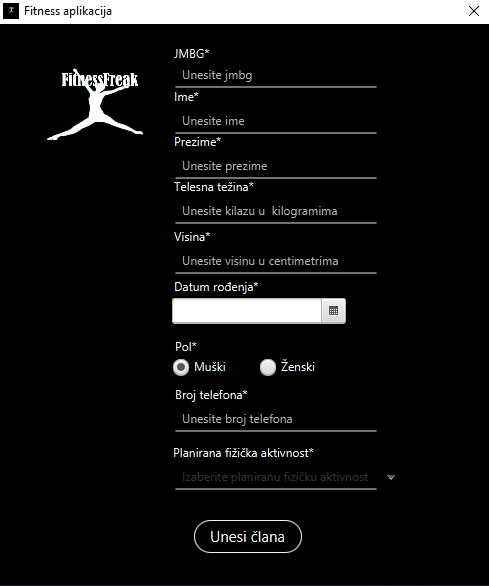
public ArrayList <eMisicnaGrupa> dodajMisicneGrupuVezbi(){

return null;

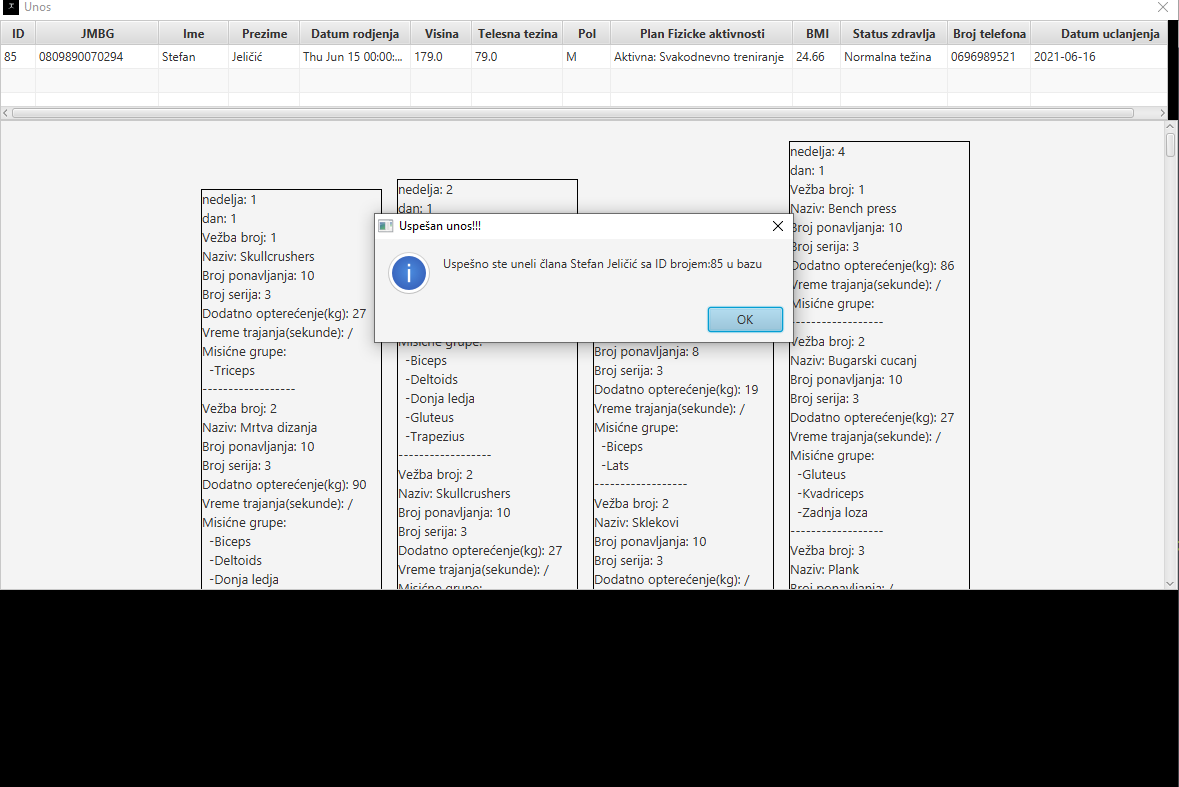
}

}

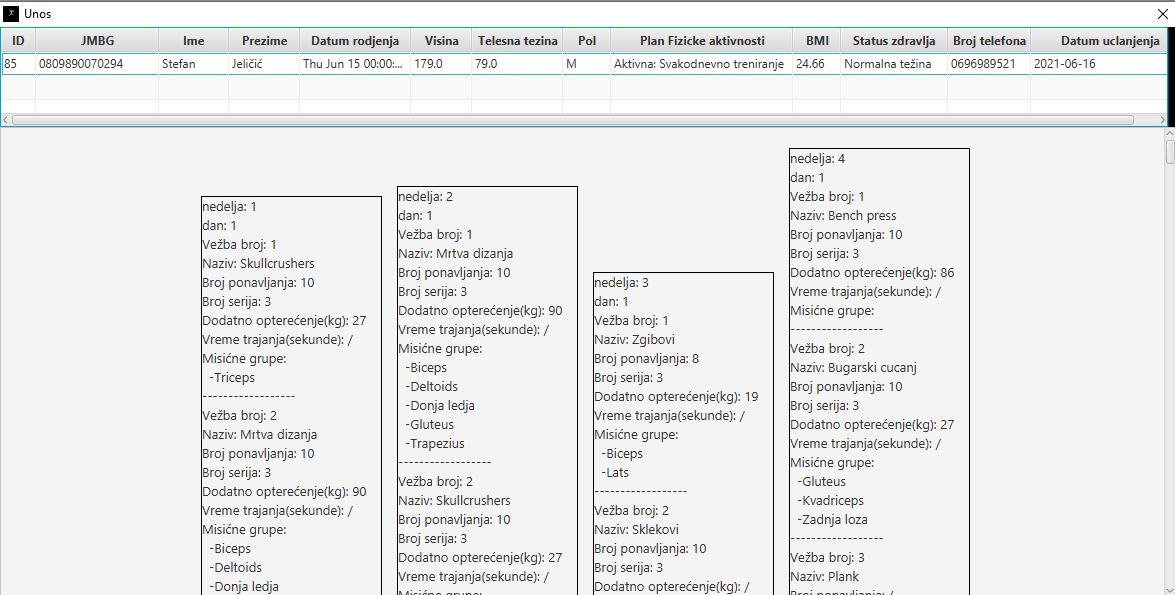
# Prikaz i opis gotove aplikacije



**Slika 14**. Forma za unos članova



**Slika 15.** Notifikacija o uspešnom unosu



**Slika 16.** Prikaz generisanog plana treninga

Pokretanjem programa se dolazi do prikaza početne forme za unos članova. U formi se unose podaci o članu, al od najvećeg značaja je njegov plan fizičke aktivnosti. Član ima mogućnost da bira koliko često planira da trenira na nedeljnom nivou. U zavisnosti od njegovog izbora vršiće se drugačiji vid generisanja treninga. Prlikom unosa člana u slučaju da podaci nisu ispravno popunjeni, izbacuje se alert sa datim greškama. Isto tako se izbacuje alert ukoliko član već postoji u bazi. U slučaju da su podaci ispravno popunjeni kreće se sa generisanjem plana treninga. Iz baze podataka se povlače vežbe i mišićne grupe, i one se stavljaju u treninge. Svaki trening je različit i formira se u odnosu na atribute korisnika, tj. Njegovog statusa zdravlja, telesne težine i starosnog doba. Svaki član dobija personalizovani plan treninga koji najviše njemu odgovara. Nakon što su se uspešno generisali treninzi oni se prikazuju preko “GridPane-a” koji se nalazi unutar “ScrollPane-a”. Na ekran se ispisuju treninzi, gde referent ima mogućnost listanja datog prikaza preko “scroll bar-a” koji se nalazi sa desne strane.

U slučaju da prilikom unosa podataka o članu ne postoje vežbe i mišićne grupe unutar baze podataka, tj da nisu upisane. Program će se prisilno zaustaviti pre započinjanja generisanja treninga.

Klikom na dugme unesi člana automatski se i čiste svi uneseni podaci iz tekstualnih polja. Tako da prilikom uspešnog generisanja treninga za člana i zatvaranja prozora, u kome je plan generisan. Mogu da se istog trenutka unose podaci o drugom članu.