



**Zimski semestar, 2019**

## **Aplikacija za unos kaloriskih vrednosti**

Mentor

Dr. Slobodan Jovanović

Kandidat

Miloš Atanacković 1542

## Sadržaj

<b>UVOD .....</b>	<b>3</b>
<i>Ključne reči .....</i>	3
<b>Opis odabranih tehnologija I alata .....</b>	<b>4</b>
<b>Specifikacija zahteva sistema .....</b>	<b>6</b>
Funkcionalni zahtevi sistema.....	6
<i>Korisnik – Klijent.....</i>	6
<i>Administrator .....</i>	7
Nefuknionalni zahtevi .....	7
<i>Performanse sistema .....</i>	7
<i>User friendly interfejs.....</i>	7
<i>Bezbednost I sigurnost sistema.....</i>	7
<i>Raspolozivost sistema .....</i>	8
<b>Modelovanje i dizajn sistema.....</b>	<b>9</b>
Dijagram slučajeva korišćenja aplikacije .....	9
<i>Use case dijagram.....</i>	9
<i>Scenariji slučajeva korišćenja korisnik .....</i>	10
<i>Scenariji slučajeva korišćenja administrator .....</i>	14
<i>Dijagram sekvenci .....</i>	15
<b>Arhitektura sistema .....</b>	<b>20</b>
Arhitektura angular.....	20
.....	20
<b>Kreiranje veb aplikacije u Web storm i Php storm-u, i podešavanje okruženja .....</b>	<b>21</b>
Kreiranje aplikacije.....	21
<i>Angular CLI.....</i>	22
<b>Kodna implementacija .....</b>	<b>28</b>
.....	34
<b>Testiranje sistema.....</b>	<b>36</b>
Funkcionalno testiranje.....	36
<i>Testovi slučajeva .....</i>	36
Jedinično(Unit) testiranje.....	43
<b>Korisničko uputstvo .....</b>	<b>46</b>
<b>Zaključak .....</b>	<b>52</b>
<b>Literatura.....</b>	<b>53</b>

## UVOD

U ovom projektu bice kreirana web aplikacija koja ce omoguciti korisniku unos kalorija. Za implementaciju aplikacije sa dinamičkim sadržajem korišćen je angular 5, a za izgled, grid i tabele je korišćen framework pod nazivom Bootstrap. Sistem se deli na administratorsku i korisničku stranu. Na stranici kalorijske tablice, nalazice se bootstrap tabela sa kaloriskom tablicom namirnica. Pored stranice kalorijske tablice, nalaze se stranica Unos kalorija, na kojoj se nalazi forma za unos namirnice i kalorija, i stranica iscitavanje unosa, preko koje cemo da prikazujemo podatke iz baze, dakle podatke sa stranice unos kalorija. Korisniku ce biti omoguceno logovanje na svoj akaunt, tek nakon toga ce mu biti omoguceno da upisuje kalorije.

### Ključne reči

Web aplikacija, merenje kalorija, baza podataka, bootstrap, php, angular2

## Opis odabranih tehnologija i alata

Veb system se zasnima i implementiran je korišćenjem sledećih tehnologija:

- HTML
- CSS
- Bootstrap
- PHP
- Angular 5
- MySQL baza podataka
- Apache WebServer
- NodeJS



Za razvoj back-end-a samog sistema, korišćen je PHP jezik koji se spaja sa front-end-om čija će funkcionalnost biti pisana u angular 5 jeziku. Forme i izgled celog sistema će se prikazivati na web pretraživaču stilizovano sa framework-om Bootstrap, koji je pisan u kombinaciji sa HTML, CSS I Jquery.. Cela aplikacija će biti u stalnoj interakciji sa MySQL bazom podataka, u koju se vrši upis, a I koristi se za iščitavanje podataka iz nje. Za optimalan rad cele aplikacije, potreban je server, u ovom slučaju korišćen je Apache WebServer.

Alati koji će se koristiti za implementaciju sistema su:

1. WebStorm
2. PHPStorm
3. XAMPP
4. Power Designer



## Specifikacija zahteva sistema

Zahtevi sistema, po definiciji se dele na funkcionalne, nefunkcionalne i dodatne zahteve.

### Funkcionalni zahtevi sistema

#### Korisnik – Klijent

Svaka osoba koja pristupi Login stranici preko web čitača je korisnik sistema. Administrator je rukovodilac sistema koji ima pristup podacima i opcijama sistema kojima običan korisnik ne može da pristupi, npr. da manipuliše korisnicima, briše određene unose. Ulogovani korisnik u sistem može da se opredeli da li će da upisuje broj kalorija, a upisom broja kalorija će morati da upiše sve podatke koji su potrebni kako bi zahtev bio potpun, ili će da upisuje broj glikemije i insulina koji takođe zahtevaju dodatne podatke kao što su vreme i datum kao i količina unešenog broja insulina ili glikemije.

#### Registracija

Nakon pristupa na sistem, korisniku se prikazuje početna stranica, i menu na kojem može da bira da li će da se prijavi na sistem, ili, ako je nov korisnik, da se registruje na sistem. Nov korisnik, da bi se registrovao na sistem, iz menija bira "Registruj se", te mu se dinamički prikazuje stranica sa formom za registraciju. U formi se nalaze polja za upis sledećih podataka: Ime, Prezime, Username i Password. Nakon registracije na sistem, korisnik je spreman da u potpunosti pristupi sistemu.

#### Prijava

Korisnik koji se uspešno registrovao na sistem, unosi podatke koje je uneo u poljima username i password na stranici za registraciju, zahtev se šalje na back-end, gde se proverava ispravnost unetih podataka, ako sa servera stigne potvrDNA informacija o unetim kredencijalima, korisnik pristupa sistemu.

#### Unos namirnica

Nakon uspešnog pristupa na sistem, korisnik može da bira koju vrstu unosa će da izvrši. Odabirom unos-a namirnica, korisnik unosi ime namirnice i broj kalorija. Ispod forme, nalaze se kalorijske tablice, kako bi korisnik mogao da zna koliku namirnica ima kalorijsku vrednost.

#### Unos kalorija

Nakon uspešnog pristupa na sistem, korisnik može da bira koju vrstu unosa će da izvrši. Odabirom unos-a kalorija, korisnik bira namirnicu iz padajućeg menija koju je predefinisao na stranici unos manirnica, bira vreme i datum unosa kalorija, kao i kalorijsku vrednost.. Ispod forme, nalaze se kalorijske tablice, kako bi korisnik mogao da zna koliku namirnica ima kalorijsku vrednost.

## **Unos insulina**

Nakon uspešnog pristupa na sistem, korisnik može da bira koju vrstu unosa će da izvrši. Odabirom unos-a insulina, korisnik iz padajućeg menija može da bira tip insulina: brzodelujući i sporodelujući. Pored tipa insulina, korisnik bira vrednost insulina i vreme i datum unosa.

## **Unos glikemije**

Nakon uspešnog pristupa na sistem, korisnik može da bira koju vrstu unosa će da izvrši. Odabirom unos-a glikemije, korisnik upisuje vrednost glikemije kao i vreme i datum unosa.

## **Dnevnik**

Na stranici dnevnik, korisnik ima kompletan prikaz unosa glikemije, inulina, kalorijske vrednosti po danima mesecima i godinama. U svakom vremenskom intervalu, pojavljuje se tablica sa unosima, kao i chartovi koji prikazuju određeni unos.

## **Administrator**

Administrator ima pristup korisnicima, klikom na dugme obriši korisnika, briše sve vezano za korisnika, njegov "profil" na sistemu.

## **Nefuknionalni zahtevi**

### **Performanse sistema**

Sistem je dostupan svakom korisniku ko može da pristupi internetu. Sistem će se nalaziti na domenu [www.merenjekalorija.com](http://www.merenjekalorija.com), do njega se može doći pretragom preko internet pretraživača ili direktnim kucanjem domena.

### **User friendly interfejs**

Sistem je pravljen tako da bude jasan i koncizan, namenjem korisnicima svakog uzrasta. Sačinjen je od formi, pored koje piše šta treba da se upiše, te je olakšano svim korisnicima.

### **Bezbednost i sigurnost sistema**

Upis u bazu podataka je siguran, podaci od šifre se konvertuju u md5 koji vraća hash od 32 karaktera, te se tako upisuje u bazu.

## Raspolozivost sistema

Nakon učitavanja stranice, sistem je spreman za rad. Korisnik može da pristupi potpunom sistemu nakon registracije. Sistem će biti dostupan 24 časa dnevno, tokom cele godine, sa bilo koje lokacije, pomoću uređaja koji imaju mogućnost konekcije na internet.

## Modelovanje i dizajn sistema

Pre izrade aplikacije, prvi korak je bilo kreiranje i modelovanje dijagram slučajeva korišćenja aplikacije

### Dijagram slučajeva korišćenja aplikacije

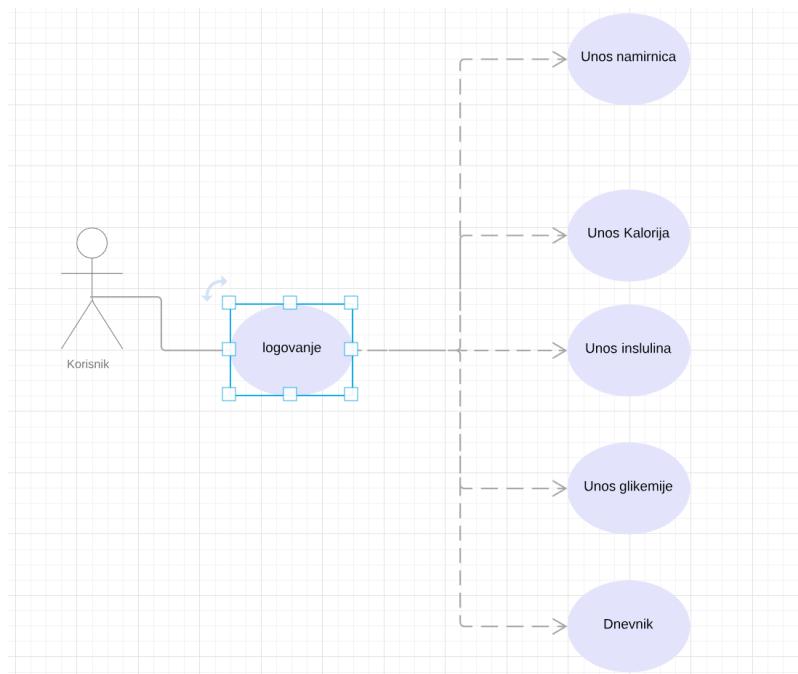
Sistem se sastoji od dva različita aktera koji ga koriste:

1. Administrator
2. Korisnik

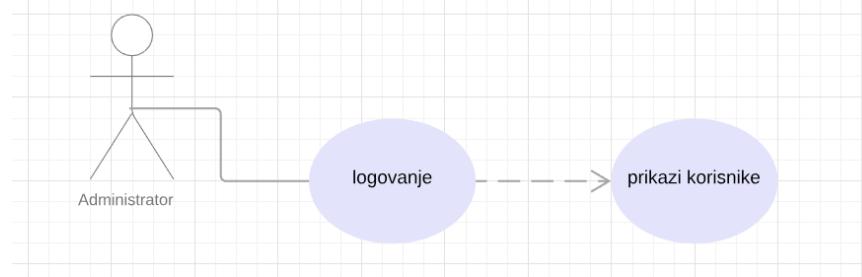
U ovom sistemu, postoji jedan actor a to je korisnik. Korisnik kreira nalog tako što ispunjava formu sa stranice registracija. Nakon napravljenog naloga, može da pristupi sistemu. Nakon registracije, korisnik ima uvid u kalorisku tablicu svih namirnica, nakon pronalaženja odredjene namirnice, upisuje u tablicu gde se računa unos kalorija a može i da unosi glikemiju i insulin

Dijagram slučajeva korišćenja je prikazan na sledećoj slici:

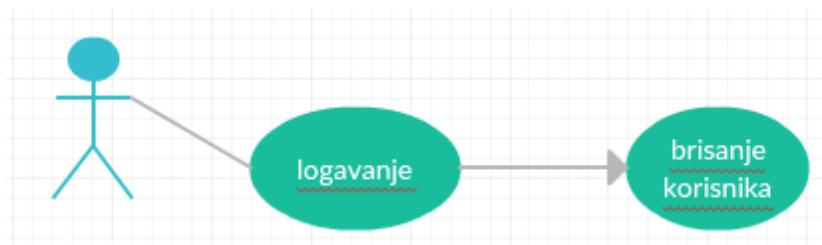
**Use case dijagram**



Slika 1 - Use case dijagram korisnik



Slika 2 - Use case dijagram administrator



Slika 3 - Use case dijagram administrator, brisanje korisnika

### Scenariji slučajeva korišćenja korisnik

#### **Tabela slučaja korišćenja registracije**

<b>Slučaj korišćenja: Registracija</b>	
<b>Opis:</b>	Registracija korisnika
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Sistem je aktivan
<b>Optimistični tok dogadjaja</b>	Korisnik unosi podatke za registraciju: ime, prezime, username i password
<b>Uslovni tok</b>	Bez pravljenja naloga, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik je registrovan i može da pristupi sistemu

#### **Tabela slučaja korišćenja prijavljivanja na sistem**

<b>Slučaj korišćenja: Prijava na sistem</b>	
<b>Opis:</b>	Prijava korisnika na sistem
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Sistem je aktivan

<b>Optimistični tok dogadjaja</b>	Korisnik unosi username i password i pristupa sistemu
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik se prijavio na sistem

### **Tabela slučaja korišćenja pregled kalorijske tablice**

#### **Slučaj korišćenja: Pregled kalorijske tablice**

<b>Opis:</b>	Korisnik pristupa kalorijskoj tablici
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Pregleda kalorije tablice
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima uvođenje u kalorijske vrednosti kako bi znao šta da unosi u sistem

### **Tabela slučaja korišćenja upisa kalorija**

#### **Slučaj korišćenja: Upis kalorija**

<b>Opis:</b>	Korisnik upisuje namirne i kalorije u svoju tablicu
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik upisuje broj kalorija i namirnicu u sistemu
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima upisan broj kalorija i namirnice u sistemu

### **Tabela slučaja korišćenja iščitavanje kalorija**

#### **Slučaj korišćenja: Iščitavanje kalorija**

<b>Opis:</b>	Korisnik isčitava dnevni unos kalorija i namirnica
<b>Aktor</b>	Korisnik

<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem I da ima upisane vrednosti kalorija I namirnica u bazi
<b>Optimistični tok dogadjaja</b>	Korisnik iscitava unos kalorija I namirnica
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik je informisan o svom unosu kalorija I namirnicama

### **Tabela slučaja korišćenja upisa namirnice**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik upisuje namirnece i kalorijsku vrednost
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik upisuje ime namirnice i kalorijskuvrednost
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima upisan broj kalorija I namirnicu u sistem

### **Tabela slučaja korišćenja upisa insulina**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik upisuje tip insulina, vrednost i datum
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik upisuje tip insulina, vrednost i datum
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima upisan tip insulina, vrednost i datum

### **Tabela slučaja korišćenja upisa glikemije**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik upisuje vrednost glikemije, vreme i datum
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik upisuje vrednost glikemije, vreme i datum

<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima upisanu vrednost glikemije, vreme i datum

### **Tabela slučaja korišćenja upisa dnevnika po danu**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik iscrtava vrednost podataka po danu
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik iscrtava sa stranice dnevnik, podatke za dan
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima informacije o podacima iz tog dana

### **Tabela slučaja korišćenja upisa dnevnika po nedelju**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik iscrtava vrednost podataka po nedelji
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik iscrtava sa stranice dnevnik, podatke za nedelju
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima informacije o podacima iz te nedelje

### **Tabela slučaja korišćenja upisa dnevnika po mesecu**

<b>Slučaj korišćenja: Upis namirnice</b>	
<b>Opis:</b>	Korisnik iscrtava vrednost podataka za mesec
<b>Aktor</b>	Korisnik
<b>Preduslovi</b>	Korisnik mora da bude prijavljen na sistem
<b>Optimistični tok dogadjaja</b>	Korisnik iscrtava sa stranice dnevnik, podatke za mesec
<b>Uslovni tok</b>	Bez prijave na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Korisnik ima informacije o podacima iz tog meseca

## Scenariji slučajeva korišćenja administrator

### **Tabela slučaja korišćenja prijavljivanja na sistem kao administrator**

Slučaj korišćenja: Prijava na sistem	
<b>Opis:</b>	Prijava korisnika na sistem sa administratorskim kredencijalima
<b>Aktor</b>	Administartor
<b>Preduslovi</b>	Sistem je aktivan
<b>Optimistični tok dogadjaja</b>	Administartor unosi username i password za administratora
<b>Uslovni tok</b>	Bez prijave kao administrator na sistem, korisnik ne može da pristupi sistemu
<b>Postuslovi</b>	Administartor se prijavio na sistem kao administrator

### **Tabela slučaja korišćenja prikaza korisnika**

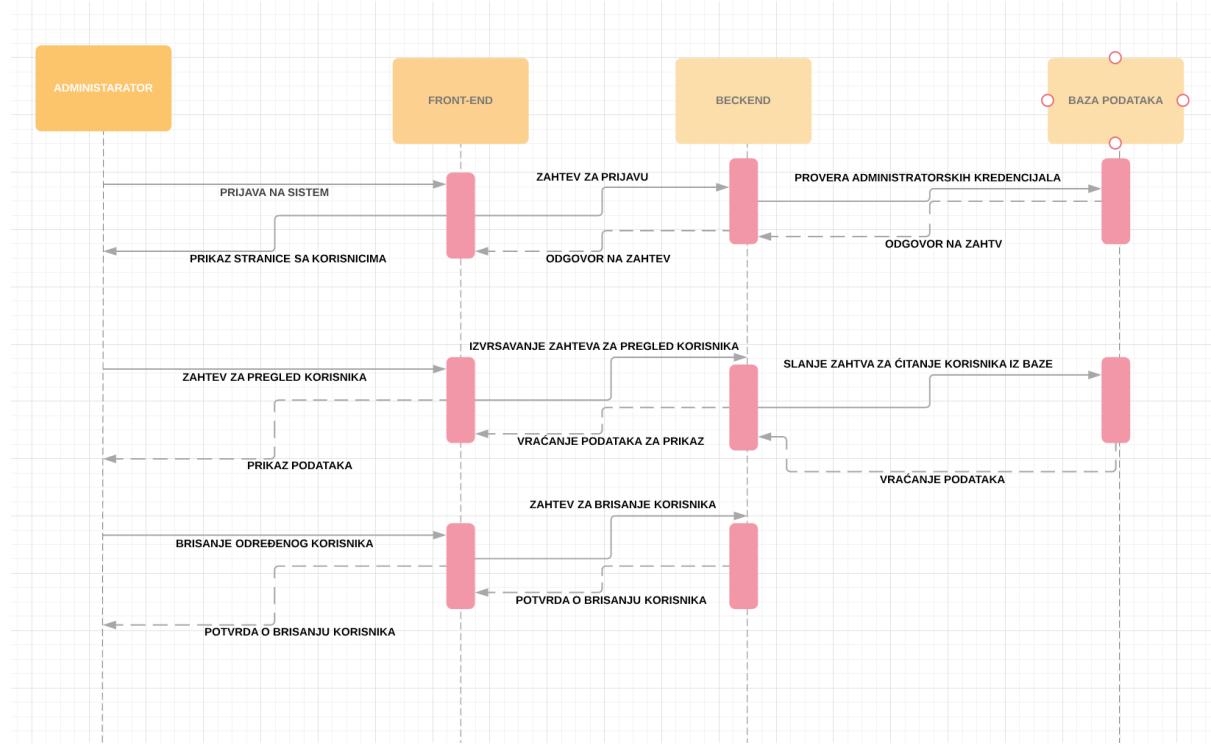
Slučaj korišćenja: Prijava na sistem	
<b>Opis:</b>	Prijava administratora na sistem i pristup bazi korisnika
<b>Aktor</b>	Administartor
<b>Preduslovi</b>	Sistem je aktivan
<b>Optimistični tok dogadjaja</b>	Administrator manipuliše korisnicima
<b>Uslovni tok</b>	Administrator briše korisnike
<b>Postuslovi</b>	Administrator je uspešno izbrisao korisnika

## Dijagram sekvenci

U daljem tekstu će biti prikazani dijagrami sekvene, koji će da objasne tok zahteva aplikacije.

### **Dijagram sekvenci administratorskih interakcija**

Na slici ispod, prikazan je dijagram sekvenci administratorskih interakcija sa sistemom.



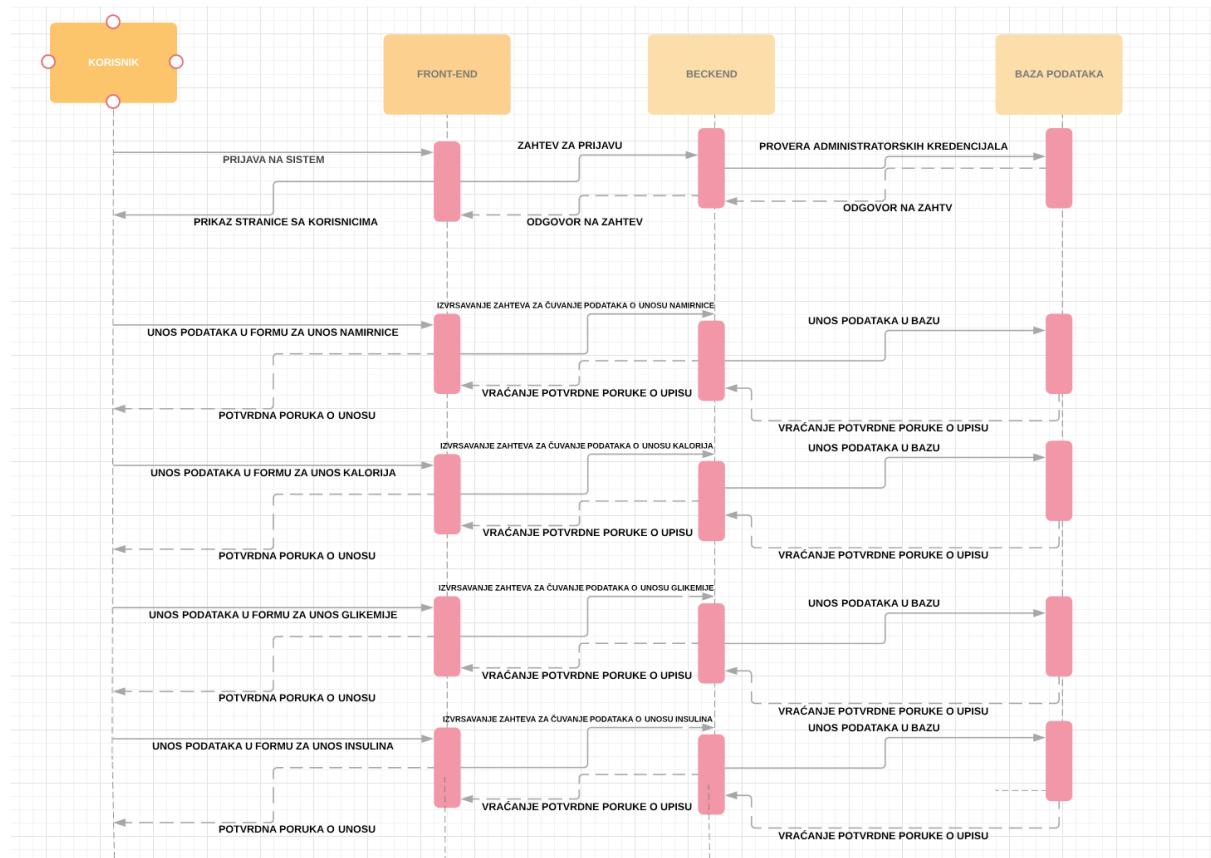
Slika 4 - Sekvencijalni dijagram administrator

Administrator pristupa stranici za prijavu na sistem, nakon uspešnog prijavljivanja sa administratorskim kredencijalima, administrator je prijavljen na sistem. Nakon toga, administrator ima mogućnost da izvrši pregled korisnika, front deo šalje zahtev beck-endu, preko kojeg se iščitavaju podaci iz baze, te se prosleđuje na front deo, kako bi korisniku bile prikazane informacije. Korisnih šalje zahtev za

brisanje korisnika, klikom na dugme naspram imena korisnika. Nakon klika, front deo šalje naredbu back delu, podatke za kojeg korisnika treba da se izvrši naredba brisanja. Back deo, proverava da li u bazi postoji takav korisnik, i ako postoji briše ga, te vraća potvrđnu poruku na front deo, kako bi korisnik dobio poruku o izvršenom zahtevu - brisanju korisnika.

## **Dijagram sekvenci klijentskih interakcija**

Na slici ispod, prikazan je dijagram sekvenci klijentskih interakcija sa sistemom

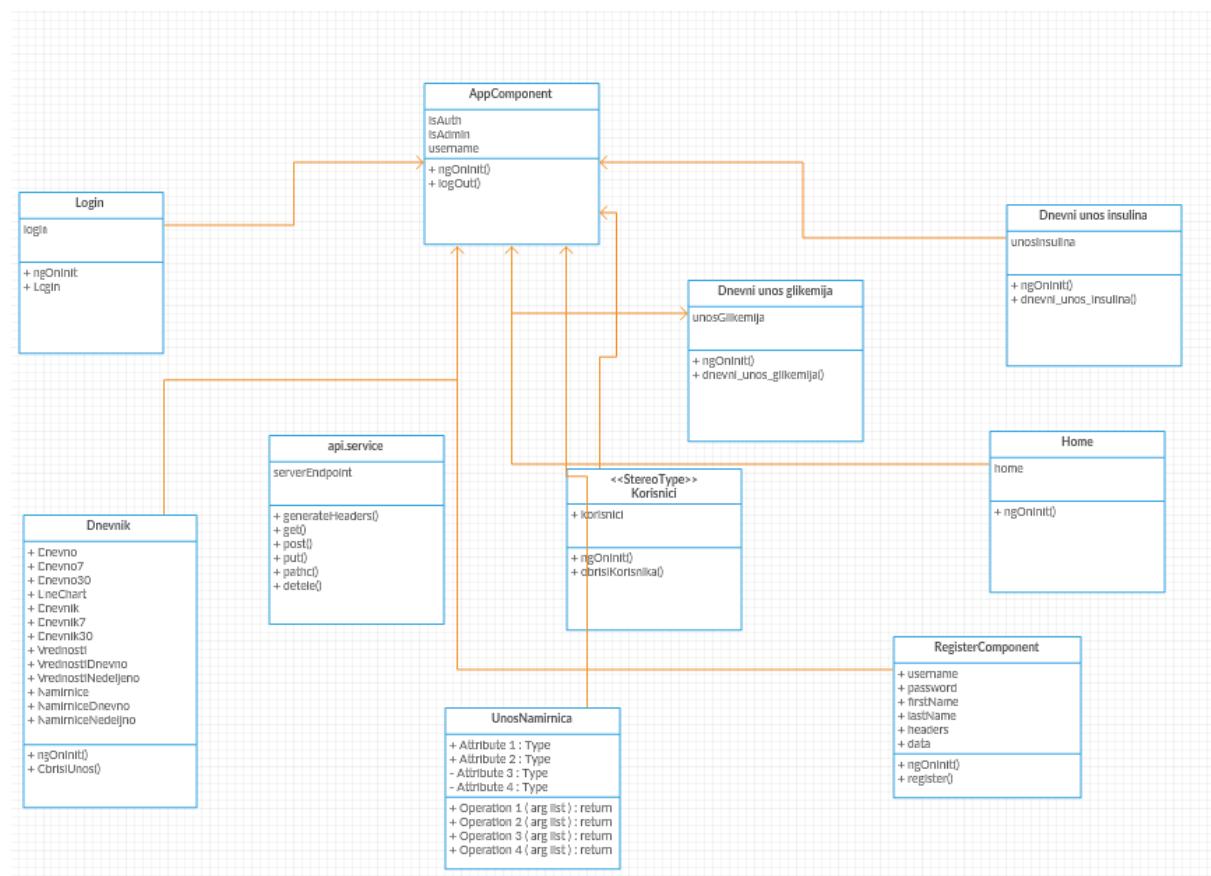


Slika 5 - Sekvencijalni dijagram korisnik

Korisnik pristupa stranici za prijavu zatim front-end deo šalje zahtev back-endu da proveri da li se u bazi podataka nalazi korisnik sa unetim kredencijalima. Nakon provere, back-end vraća odgovor o pozitivnom ili negativnom ishodu provere, te front deo prikazuje korisniku poruku. Svaki zahtev koji korisnik pošalje, prolazi kroz validaciju na front-u, stiže do beck-a i upisuje se u bazu, ili se iščitava iz iste. Nakon svakog zahteva, baza vraća podatke, u slučaju upisa, te se na front delu prikazuje korisniku.

## Klasni dijagram

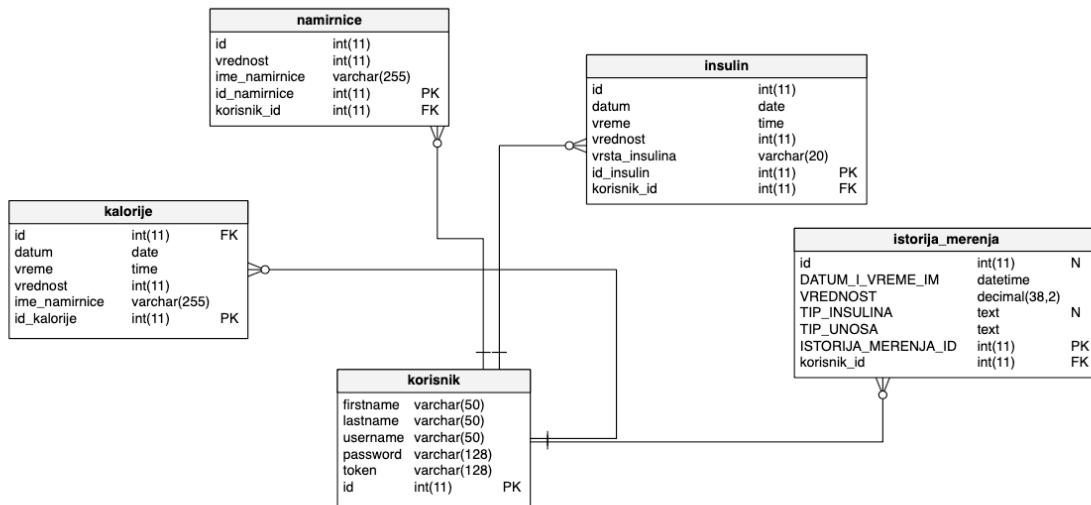
Na slici ispod se nalazi klasni dijagram, aplikacija se sastoji iz deset komponenti koji se sastoje iz klasa. AppComponent je glavna klasa, koja kontrolise celu aplikaciju, i povezana je sa svim klasama.



6 - Klasni dijagram

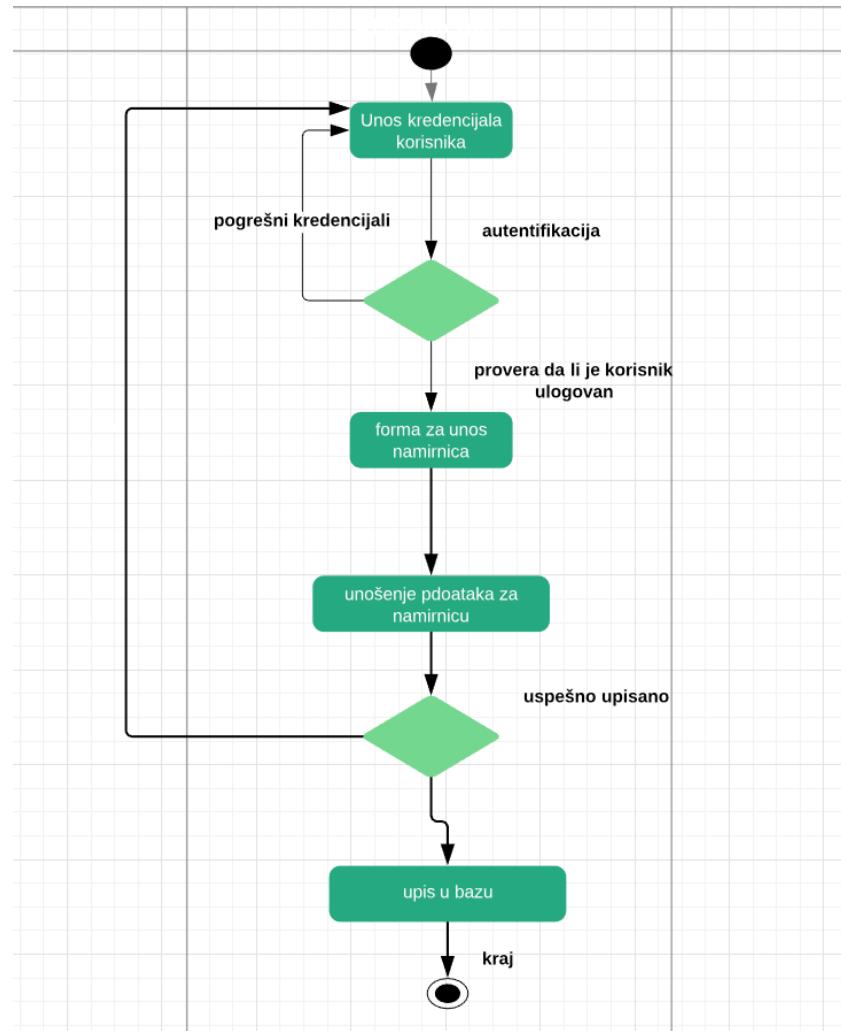
## Model baze podataka

Dok su se pravili planovi o izradi aplikacije, jedan od ključnih stvari je bilo da se osmisli i modeluje baza podataka jer je ovakav sistem u konstantnoj interakciji sa bazom, jer se za skoro svaki zahtev vrši čitanje ili upis. Bez baze podataka, ovaj sistem ne bi mogao pravilno da funkcioniše, pojavljivalo bi se puno grešaka jer predefinisano da sistem komunicira sa bazom. Za modelovanje baze, korišćenja je besplatna verzija online web aplikacije "Vertabelo", nakon modelovanja baze, ovaj program nam omogućava exportovanje modela, i ubacivanje u phpmyadmin. Na slici ispod je prikazan model baze podataka.



Slika 7 - Model baze podataka

### Dijagram aktivnosti



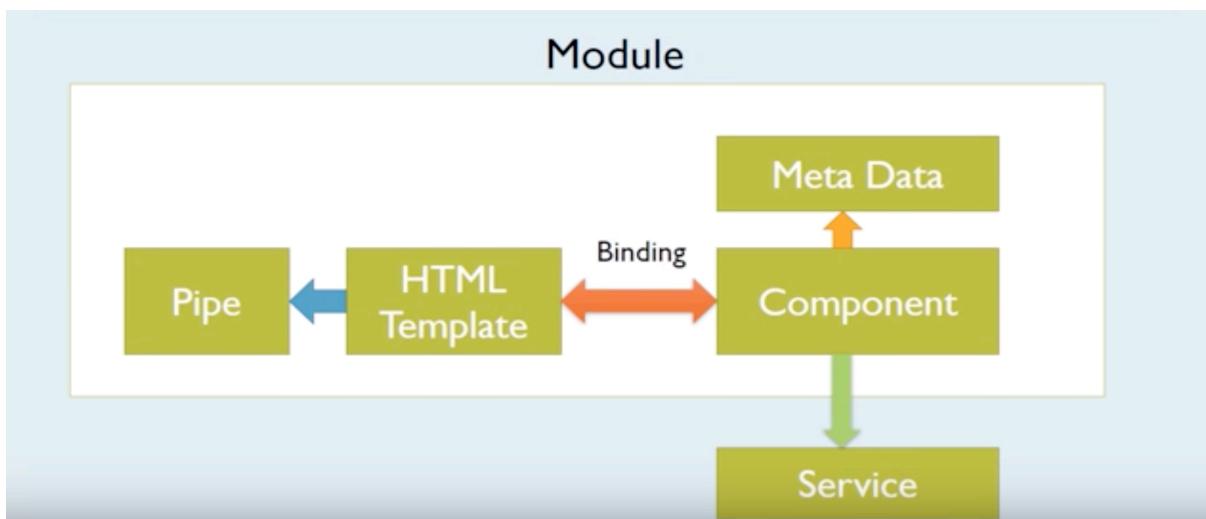
Slika 8 - dijagram aktivnosti za unošenje namirnice

Funkcionalnost ovog sistema je da korisnik u interakciji sa sistemom može da izvrši upise, te prati dalji razvoj događaja, jedan proces izvršavanja upisa namirnica je prikazan kroz dijagram aktivnosti.

## Arhitektura sistema

### Arhitektura angular

Arhitektura angular-a se sastoji od modula koji se sastoji iz komponenata kao što su html template, u kojem se pišu html dokumenti, koji su povezani sa component komponentom, a ako je potrebno, html dokument poziva i pipe. Component, ako je potrebno komunicira sa Meta data i service komponentom. Component, je grupa podataka i metoda(Event handlers)



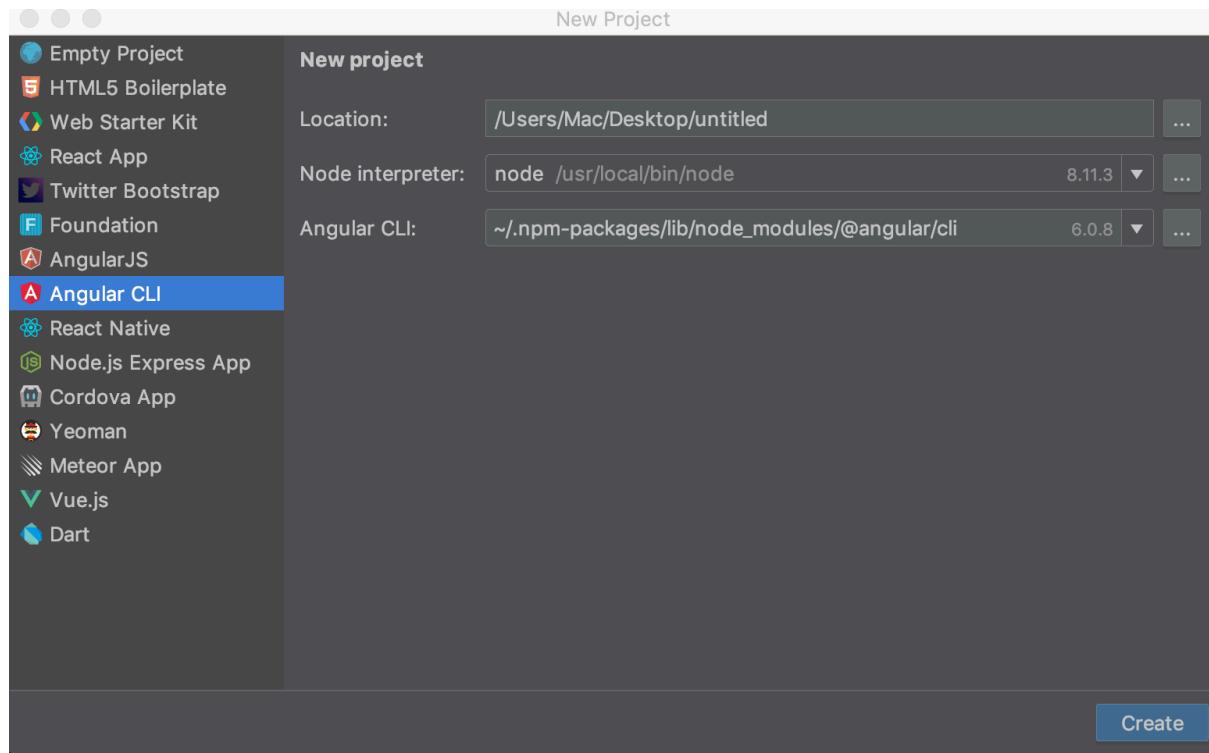
Slika 9 - Arhitektura angular-a

Prikaz veb aplikacije u veb čitaču je kombinacija angular 5 i php-a. U angularu se nalaze html dokumenti, koji prikazuju željene podatke koji se šalju preko php-a, a on ih čita iz mysql baze podataka u kojoj se oni skladiše. Cela arhitektura i sistem ne bi mogao radi bez Apache web server-a.

## Kreiranje veb aplikacije u Web storm i Php storm-u, i podešavanje okruženja

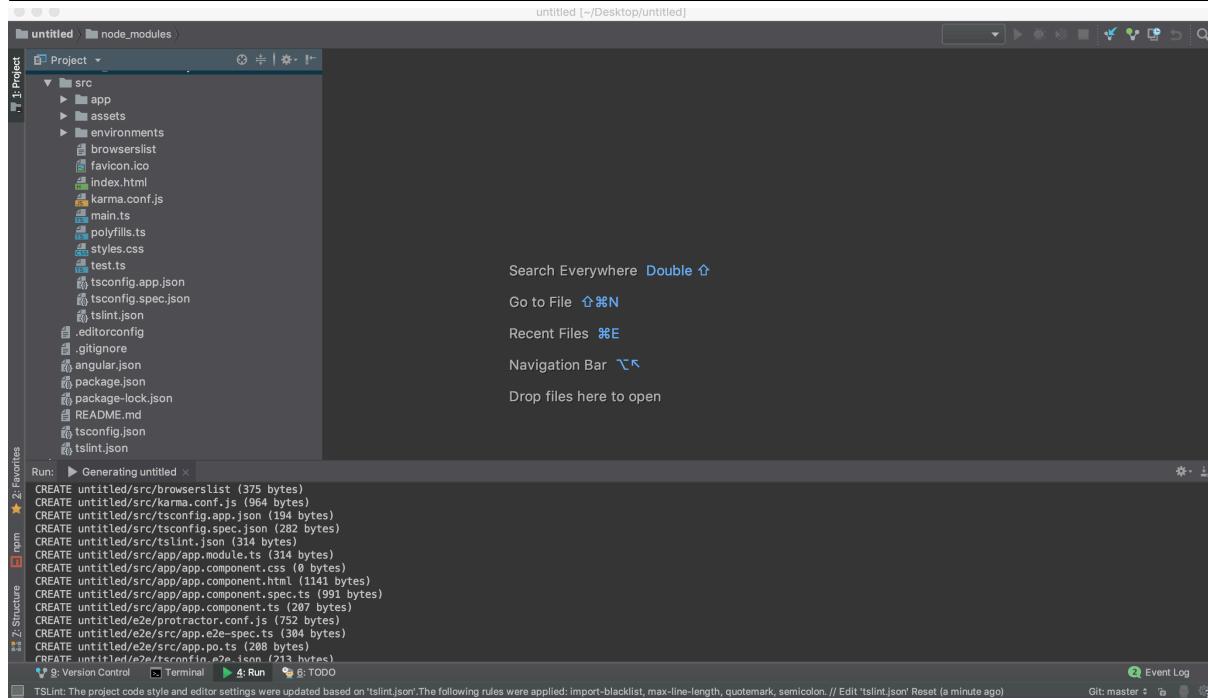
### Kreiranje aplikacije

Dalje u tekstu, biće prikazano kreiranje aplikacije kako za front, tako i za back-end, kao i sama adaptacija Web Storm-a na angular.



Slika 10 - Kreiranje novog projekta

Da bi razvojno okruženje znalo u kojem programskom jeziku pišemo aplikaciju, pri kreiranju aplikacije mora da se definiše sam programski jezik, u ovom slučaju Angular CLI. Nakon odabira, web storm kreira aplikaciju i ubacuje node module u root folder same aplikacije, kako bi imali pristup željenom modulu



Slika 11 - Uspešno kreiran novi projekat

## Angular CLI

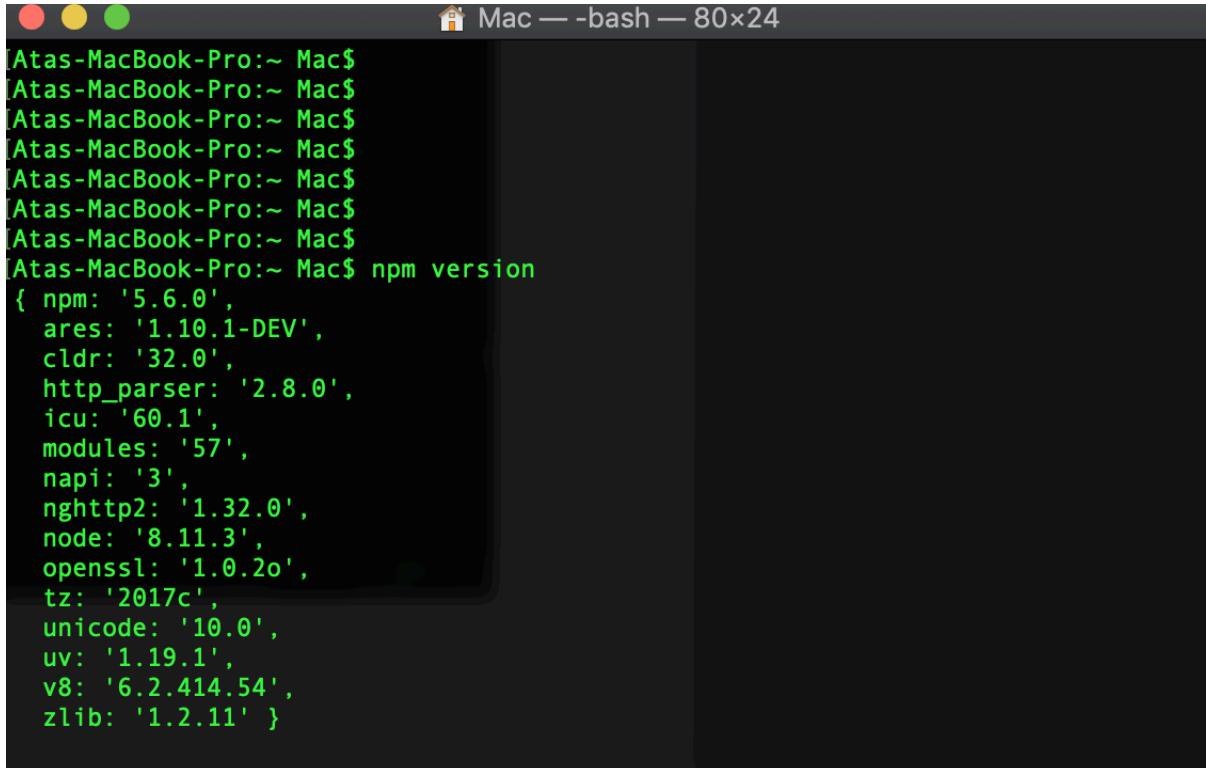
Ako se u node modulima ne nalazi Angular cli, najbitnije je da se instalira u terminalu sledećom komandom:

```
npm install -g @angular/cli
```

The screenshot shows a terminal window on a Mac OS X system. The title bar says 'Mac — bash — 80x24'. The command 'npm install -g @angular/cli' is typed into the terminal and is highlighted in green. The terminal is otherwise empty, showing only the prompt 'Atas-MacBook-Pro:~ Mac\$' repeated many times.

Slika 12 - Instalacija node angular cli

Nakon instalacije, možemo da izvršimo proveru, da li je Angular CLI instaliran komandom u terminalu:  
npm version

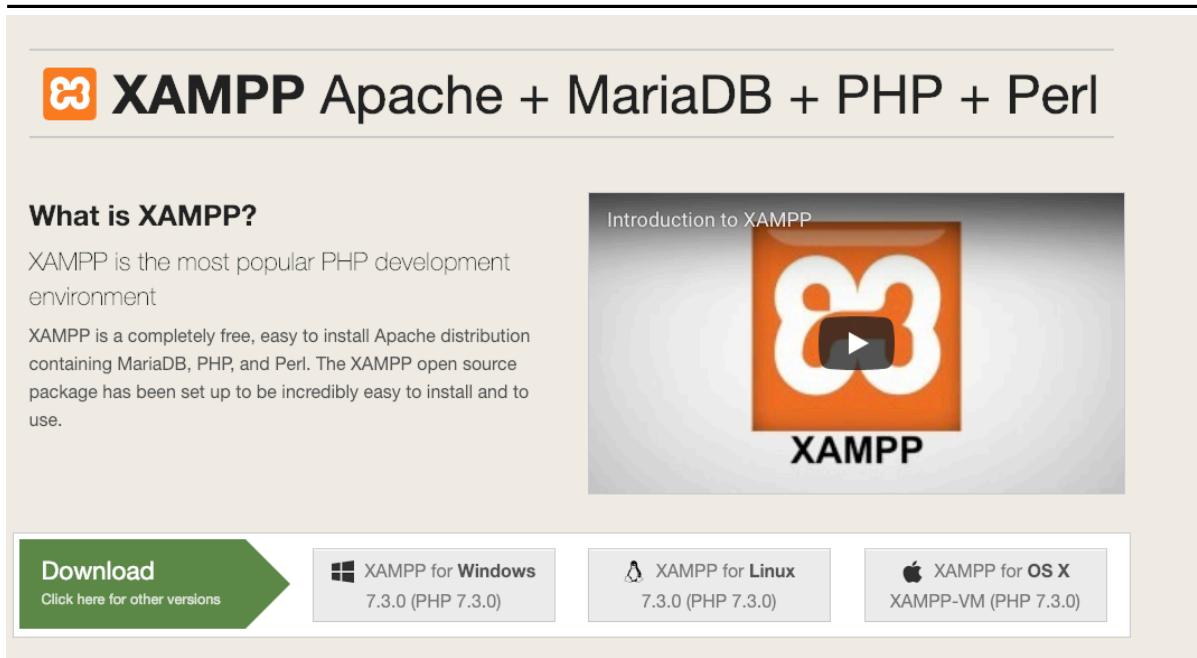


```
[Atas-MacBook-Pro:~ Mac$  
[Atas-MacBook-Pro:~ Mac$ npm version  
{ npm: '5.6.0',  
ares: '1.10.1-DEV',  
cldr: '32.0',  
http_parser: '2.8.0',  
icu: '60.1',  
modules: '57',  
napi: '3',  
nghttp2: '1.32.0',  
node: '8.11.3',  
openssl: '1.0.2o',  
tz: '2017c',  
unicode: '10.0',  
uv: '1.19.1',  
v8: '6.2.414.54',  
zlib: '1.2.11' }
```

Slika 13 - Provera verzije angular cli

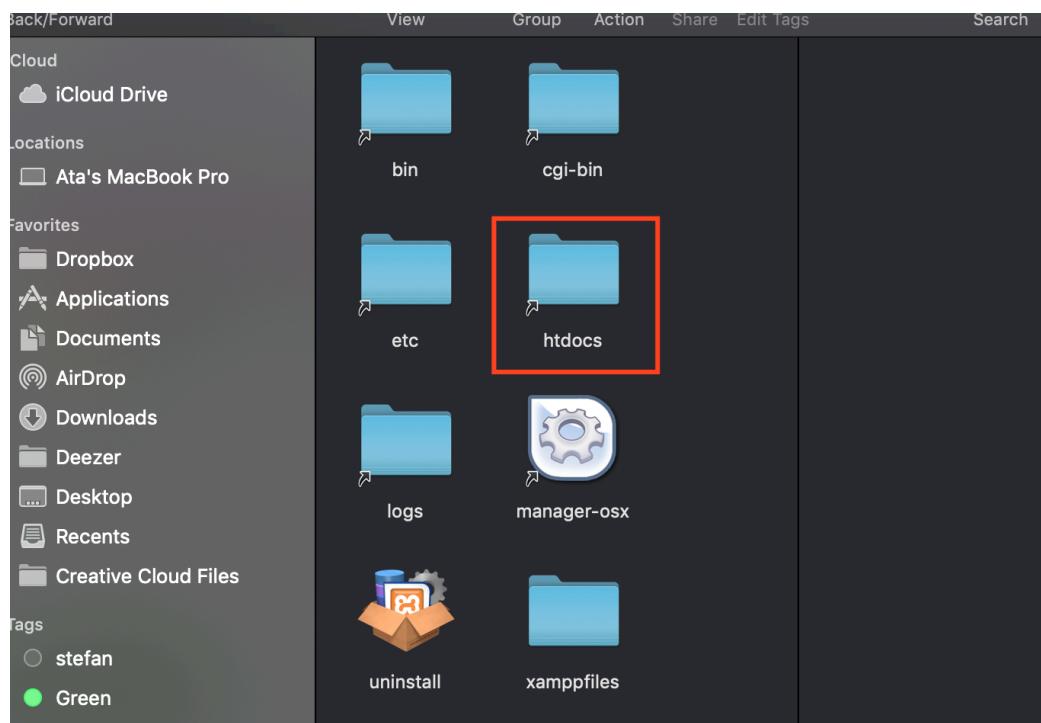
Ako imamo potvrdu verzije da je sve instalirano kako treba, možemo da generišemo aplikaciju komandom: ng new ime-aplikacije, te se komandom cd ime-aplikacije ulazi u folder u kojem je kreirana aplikacija. ng serve je komanda kojom se pokreću serveri i aplikacija se prikazuje u browser-u. Ovim putem može da se kreira aplikacija, kao i preko web-storm-a. Ako želimo da pišemo novi kod, izaberemo opciju otvaranje već postojeće aplikacije u web-storm-u.

Pri kreiranju php fajlova, potrebno je da se instalira MySQL i Apache Web Server, u ovom slučaju će biti instaliran preko aplikacije koja se naziva XAMPP, koja je besplatna i preuzima se sa njihovog oficijalnog sajta. Xampp je najpopularniji program u kojem se developuje php.



Slika 14 - XAMPP

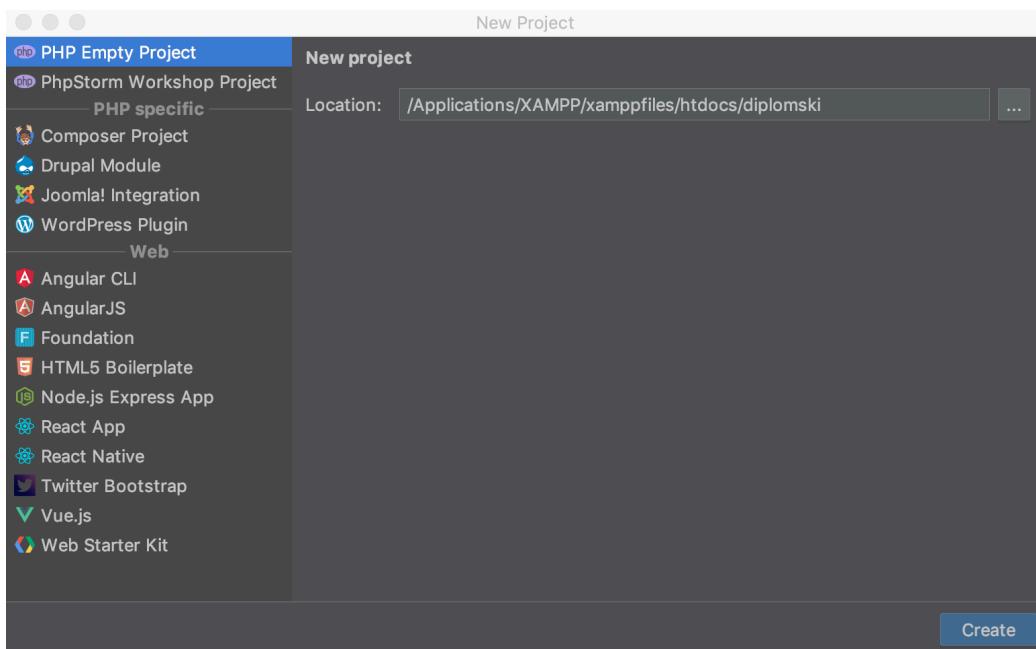
Nakon instalacije xampp-a, potrebno je locirati gde je instaliran, jer projekat koji hoćemo da kreiramo u php-u mora da se smesti u xampp root app/htdocs



Slika 15 - Htdocs folder u xampp-u

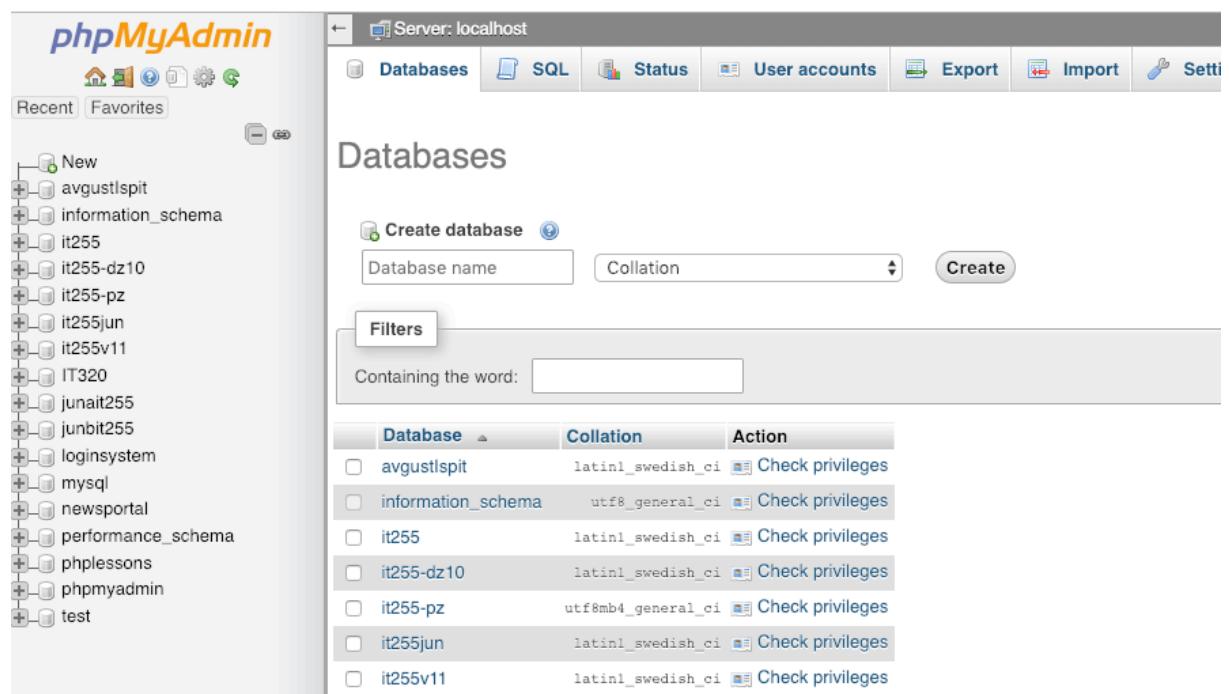
U folder-u htdocs, se pravi novi folder i u njemu se smestaju php fajlovi.

Nakon krairanja foldera, u razvojnom okruženju PHP Storm, biramo opciju za kreiranje novog projekta, ali kako je bitno da se izabere ruta do htdocs, kako bi kreirani fajlovi mogli da se pokrenu na serveru.



Slika 16 - Kreiranje php projekta

Nakon kreiranja željenih fajlova, potrebno je kreirati bazu podataka preko phpmyadmin-a, a njemu se pristupa iz lokalnog hosta linkom na : <http://localhost/phpmyadmin/>



The screenshot shows the phpMyAdmin interface. On the left, there's a sidebar with 'Recent' and 'Favorites' sections, and a tree view of databases: New, avgustlspit, information\_schema, it255, it255-dz10, it255-pz, it255jun, it255v11, IT320, junit255, junbit255, loginsystem, mysql, newsportal, performance\_schema, phlessons, phpmyadmin, and test. The main area is titled 'Server: localhost' and shows the 'Databases' section. It has a 'Create database' form with 'Database name' and 'Collation' fields, and a 'Create' button. Below it is a 'Filters' section with a search input. A table lists existing databases:

Database	Collation	Action
avgustlspit	latin1_swedish_ci	
information_schema	utf8_general_ci	
it255	latin1_swedish_ci	
it255-dz10	latin1_swedish_ci	
it255-pz	utf8mb4_general_ci	
it255jun	latin1_swedish_ci	
it255v11	latin1_swedish_ci	

Slika 17 - Kreiranje baze

Pošto je ranije napravljen model baze podataka, biramo opciju da importujemo sql file, preko kojeg ubacujemo urađenu bazu podataka.

**File to import:**

File may be compressed (gzip, bzip2, zip) or uncompressed.  
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer:  diplomski.sql (Max: 128MiB)

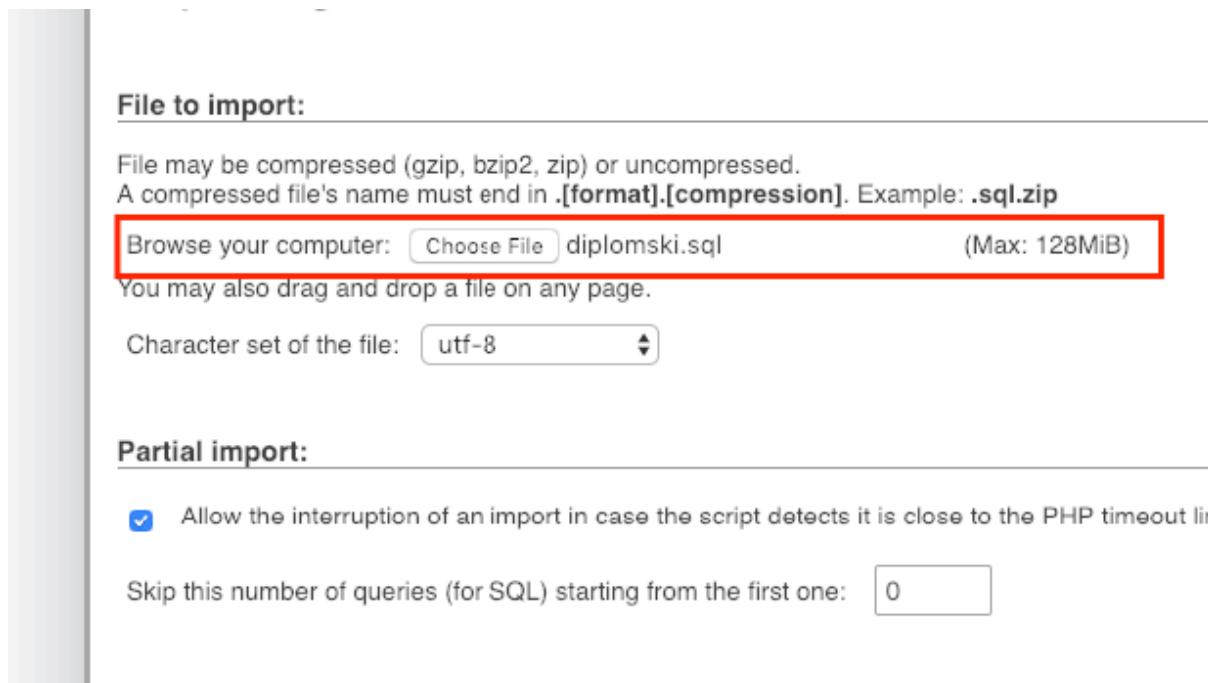
You may also drag and drop a file on any page.

Character set of the file:

**Partial import:**

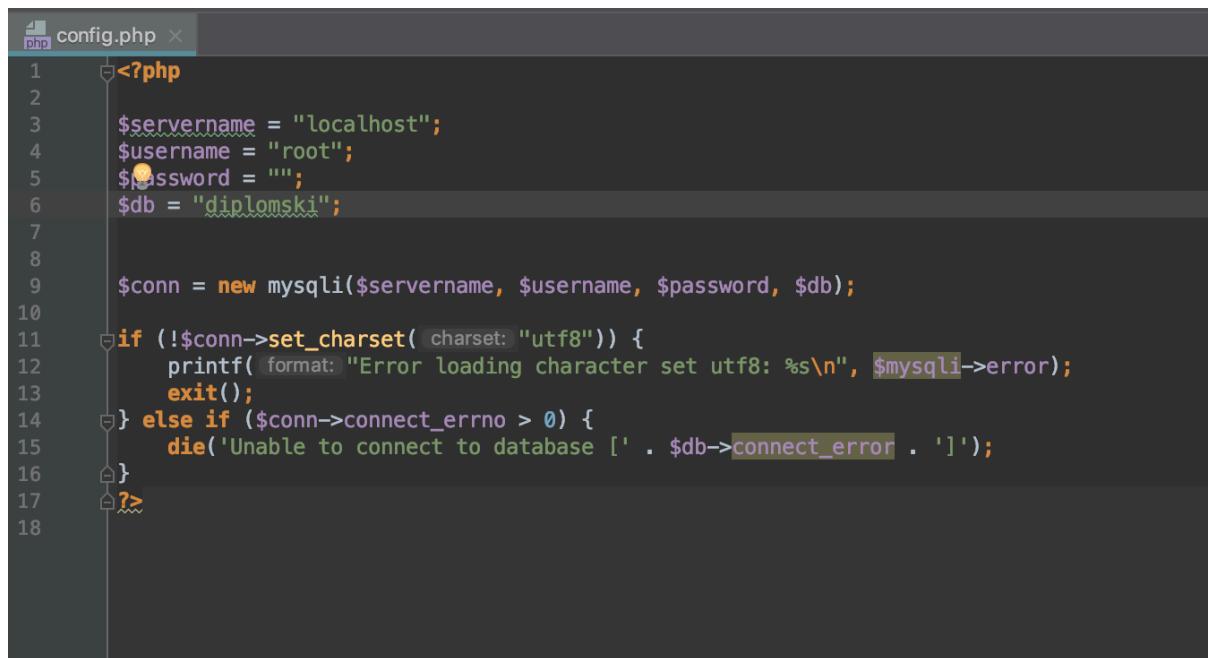
Allow the interruption of an import in case the script detects it is close to the PHP timeout limit

Skip this number of queries (for SQL) starting from the first one:



Slika 18 - Importovanje baze

Nakon ubacivanje baze podataka, potrebno je da povežemo bazu sa php config fajlom. Fajl mora ručno da se kreira. U config.php, moramo da definišemo ime servera, username, password i ime baze kao i otvorena konekcija sa istom.



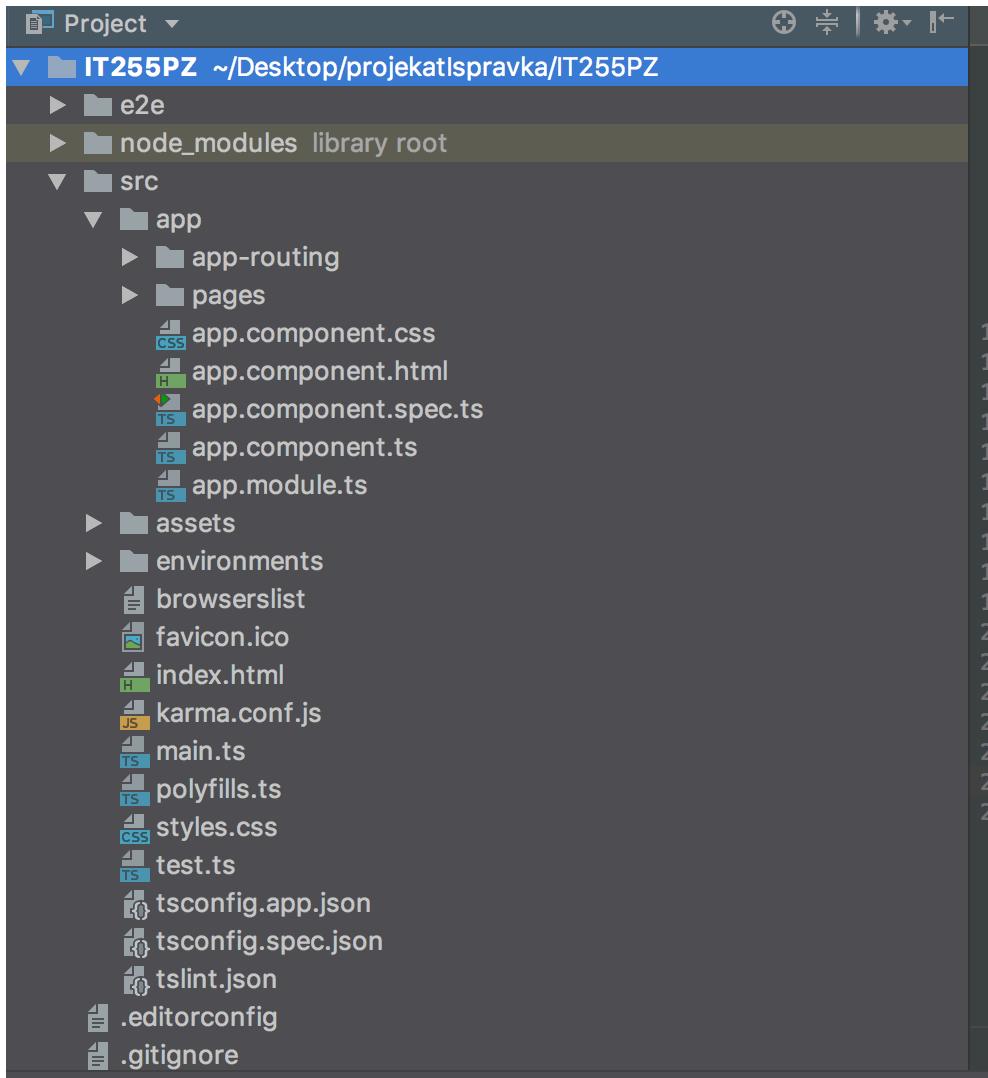
```
config.php
1 <?php
2
3     $servername = "localhost";
4     $username = "root";
5     $password = "";
6     $db = "diplomski";
7
8
9     $conn = new mysqli($servername, $username, $password, $db);
10
11    if (!$conn->set_charset("utf8")) {
12        printf("Error loading character set utf8: %s\n", $conn->error);
13        exit();
14    } else if ($conn->connect_errno > 0) {
15        die('Unable to connect to database [' . $conn->connect_error . ']');
16    }
17
18 ?>
```

Slika 19 - Config.php

Pored konekcije sa bazom, napravljen je "IF statement " ako konekcija na bazu nije dobra, da prikaze gresku.

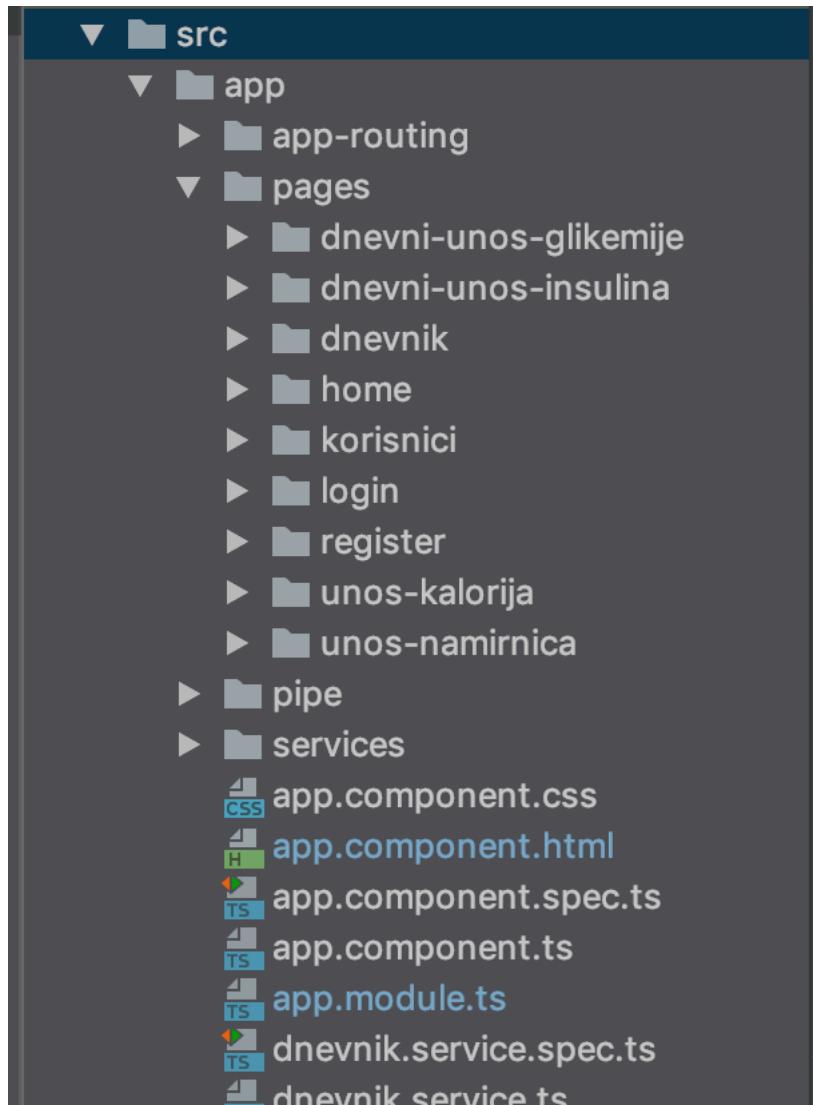
## Kodna implementacija

Pošto su izvršena sva dodavanja biblioteka i odrađene su sve konfiguracije za projekat, kodna implementacija je sledeći korak. Front-end deo se piše u WebStorm-u, dok se Back-end piše u PhpStorm-u. Front deo se sastoji od kontrolera, komponenata i servisa koji su pisani u Angular 5 jeziku, za izgled stranica je koriscen html i plugin Bootstrap. Backend je pisan u php-u, preko njega se vrši komunikacija fronta i baze, upis, iscitavanje.



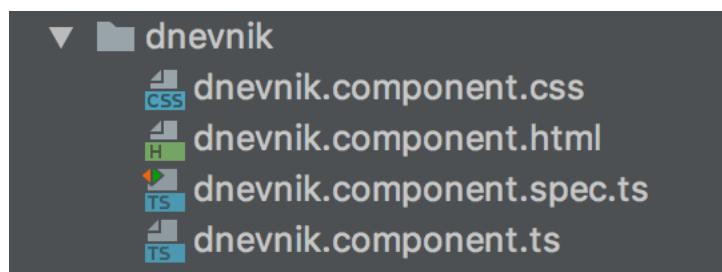
Slika 20 - Struktura projekta

Projekat se sastoji iz 5 foldera, od kojih ćemo se najviše bazirati na folder "Src". U src folderu se nalazi app folder, u kojem je smeštena cela naša aplikacija. U root folderu pored Src foldera, nalazi se node\_modules, to je folder koji se kreira preko node.js.



Slika 21 - app folder

U app folderu, su smesteni folderi sa stranicama projekta: Dnevnik, Home, orisnici, login, register, unos-kalorija, unos-namirnica, dnevni-unos-glikemije, dnevni-unos-insulina. Svaki folder od stranice se sastoji od četiri fajla, koji se generisu preko android cli, komandom u terminalu "ng generate component [name]"



Slika 22 - Primer stranice sa komponentama

[name].component.html – HTML fajl za odredjenu stranicu, sluzi za pisanje html-a stranice

[name].component.css – Css fajl za odredjenu html stranicu, u kojoj se pise izgled

[name].component.ts – Angular typescript stranica za pisanje angulara

[name].component.spec.ts – Automatski generisan angular fajl

```
import {Component, OnInit} from '@angular/core';
import {Headers, Http} from '@angular/http';

@Component({
  selector: 'app-dnevnik',
  templateUrl: './dnevnik.component.html',
  styleUrls: ['./dnevnik.component.css']
})
export class DnevnikComponent implements OnInit {

  dnevnik: any = [];

  constructor(private _http: Http) {}

  ngOnInit() {
    const headers = new Headers();
    headers.append('Content-Type', 'application/x-www-form-urlencoded');
    headers.append('token', localStorage.getItem('token'));

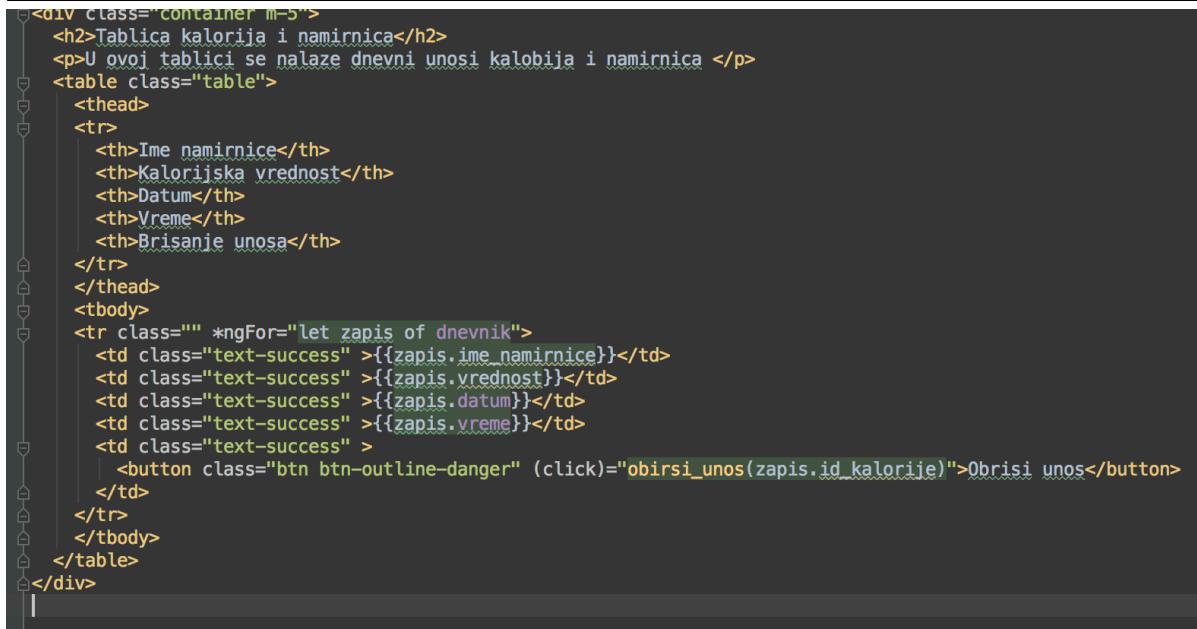
    this._http.get('http://localhost/projects/getDnevnik.php', {headers: headers}).subscribe(
      next: data => {
        this.dnevnik = JSON.parse(data['body']).zapis;
      },
      error: error => {
        console.log('Error dnevnik.component.ts: \n' + error.toString());
      }
    );
  }

  public obirsi_unos(id: number) {
    const headers = new Headers();
    headers.append('Content-Type', 'application/x-www-form-urlencoded');
    headers.append('token', localStorage.getItem('token'));

    const data = 'id=' + id;

    this._http.post('http://localhost/projects/obrisi_unos.php', data, {headers: headers}).subscribe(
      next: response => {
        console.log(response.toString());
        location.reload();
      },
      error: error => {
        console.log(error.toString());
      }
    );
  }
}
```

Slika 23 - Primer typescript stranice

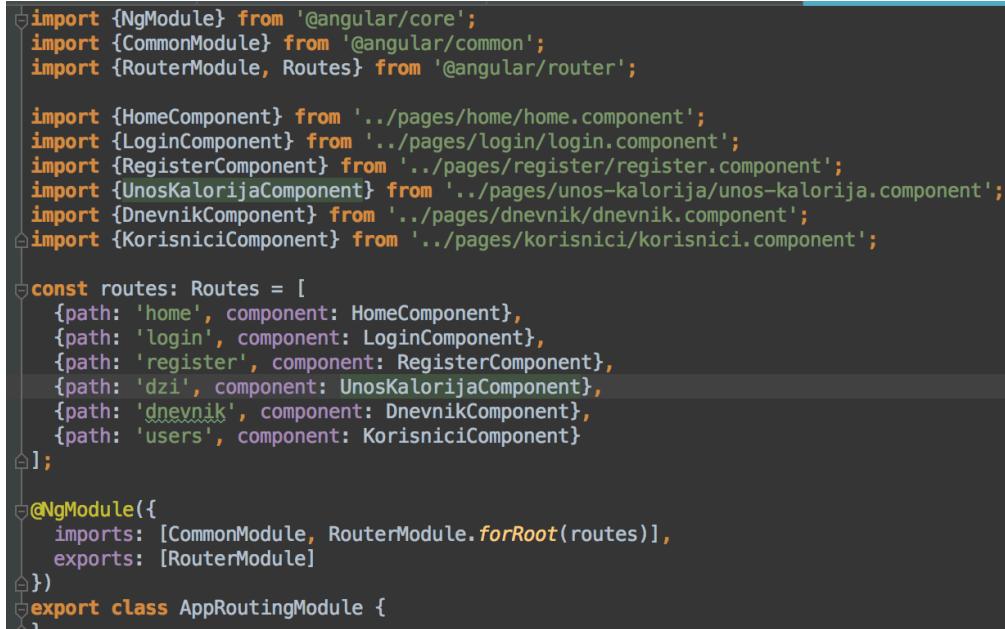


The screenshot shows a code editor with an Angular component template. The template consists of an HTML structure with a container div, an h2 header, a paragraph, and a table. The table has a header row with five columns: 'Ime namirnice', 'Kalorijska vrednost', 'Datum', 'Vreme', and 'Brisanje unosa'. The body of the table contains multiple rows, each representing a data item ('zapis'). Each row has five cells corresponding to the table headers. The last cell in each row contains a button with the text 'Obriši unos'.

```
<div class="container m-5">
  <h2>Tablica kalorija i namirnica</h2>
  <p>U ovoj tablici se nalaze dnevni unosi kalorija i namirnica </p>
  <table class="table">
    <thead>
      <tr>
        <th>Ime namirnice</th>
        <th>Kalorijska vrednost</th>
        <th>Datum</th>
        <th>Vreme</th>
        <th>Brisanje unosa</th>
      </tr>
    </thead>
    <tbody>
      <tr class="" *ngFor="let zapis of dnevnik">
        <td class="text-success" >{{zapis.ime_namirnice}}</td>
        <td class="text-success" >{{zapis.vrednost}}</td>
        <td class="text-success" >{{zapis.datum}}</td>
        <td class="text-success" >{{zapis.vreme}}</td>
        <td class="text-success" >
          <button class="btn btn-outline-danger" (click)="obirsi_unos(zapis.id_kalorije)">Obriši unos</button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Slika 24 - Primer html stranice

U angular jeziku, da bi se prikazao fajl kao stranica, mora da se definise gde se fajl nalazi, ruta.



The screenshot shows a code editor with an Angular routing module. It defines routes for various components: HomeComponent, LoginComponent, RegisterComponent, UnosKalorijaComponent, DnevnikComponent, and KorisniciComponent. Each route is defined with a path and a component. The routes array is part of the RouterModule configuration.

```
import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';
import {RouterModule, Routes} from '@angular/router';

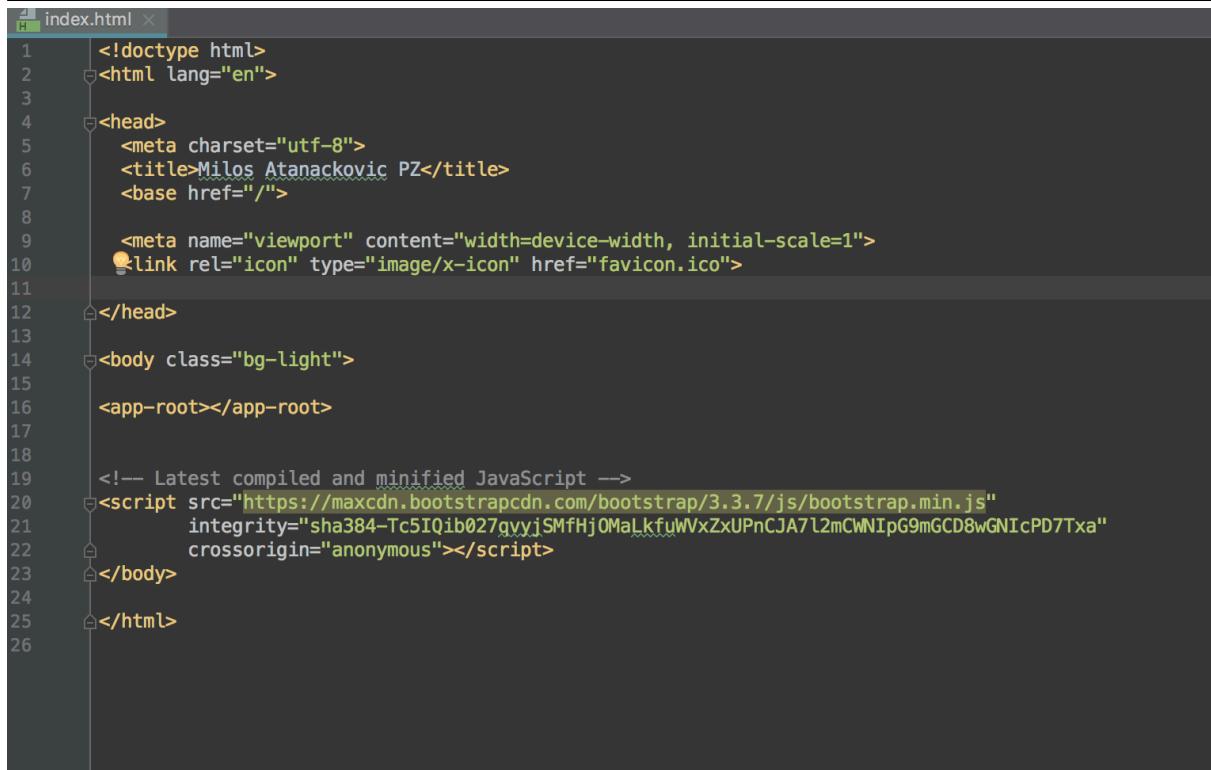
import {HomeComponent} from '../pages/home/home.component';
import {LoginComponent} from '../pages/login/login.component';
import {RegisterComponent} from '../pages/register/register.component';
import {UnosKalorijaComponent} from '../pages/unos-kalorija/unos-kalorija.component';
import {DnevnikComponent} from '../pages/dnevnik/dnevnik.component';
import {KorisniciComponent} from '../pages/korisnici/korisnici.component';

const routes: Routes = [
  {path: 'home', component: HomeComponent},
  {path: 'login', component: LoginComponent},
  {path: 'register', component: RegisterComponent},
  {path: 'dzi', component: UnosKalorijaComponent},
  {path: 'dnevnik', component: DnevnikComponent},
  {path: 'users', component: KorisniciComponent}
];

@NgModule({
  imports: [CommonModule, RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Slika 25 - App routing modul

U ovom modulu, importuju se stranice kao komponente i njegova ruta, te se definise nakon domena kako ce se pristupati odredjenij stranici.

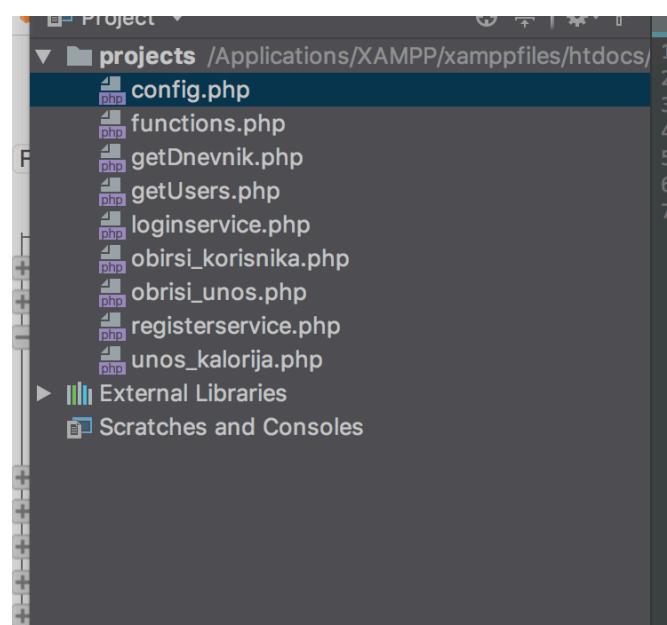


```
index.html
1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <title>Milos Atanackovic PZ</title>
7      <base href="/">
8
9      <meta name="viewport" content="width=device-width, initial-scale=1">
10     <link rel="icon" type="image/x-icon" href="favicon.ico">
11
12 </head>
13
14 <body class="bg-light">
15
16     <app-root></app-root>
17
18
19     <!-- Latest compiled and minified JavaScript -->
20     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
21             integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWxUPnCJA7l2mCWNIpG9mGCD8wGNiCpd7Tx" crossorigin="anonymous"></script>
22
23 </body>
24
25 </html>
```

Slika 26 - Index.html stranica

U ovom slučaju, index.html stranica se koristi za ubacivanje komponenata u nju. <app-root></app-root> znači da smo ucitali stranicu preko određenog selektora, u ovom slučaju app-root. Index.html se takođe koristi za importovanje određenih skripti, u ovom slučaju bootstrap.js skripta.

Sto se tice back-end dela, php fajlovi da bi radili, moraju da se ubace u htdocs, apache mysql server citaju odatle fajlove.



Slika 27 - htdocs folder

Baza podataka se sastoji iz pet kolona, insulin, istorija\_merenja, kalorije, korisnik i namirnice

Struktura tabele kalorije se sastoji iz id\_kalorije, id( id od korisnika), datum, vreme, vrednost i ime namirnice. Prilikom upisa namirnice, gorenavedeni podaci se ispisuju preko forme, a id sluzi kako bi znali koji korisnik sta upisuje.

Tabela insulin se sastoji iz id\_insulin, id, datum, vreme, vrednost i vrsta\_insulina.

**id\_insulin** - jedinstveni broj unosa insulina

**id** - po broju se zna koji je korisnik izvršio unos

**datum** - datum unosa

**vreme** - vreme unosa

**vrednost** - vrednost unosa insulina

**vrsta\_insulina** - odabir iz padajuće liste za vrstu inslunina

+ Options									
	← T →		<b>id_insulin</b>	<b>id</b>	<b>datum</b>	<b>vreme</b>	<b>vrednost</b>	<b>vrsta_insulina</b>	
<input type="checkbox"/>					1	1	2018-01-01 01:01:00	1	Brzodelujuci
<input type="checkbox"/>					2	1	2018-01-02 02:02:00	2	Sporo delujuci

Slika 28 - tabela insulin

Tabela istorija merenja, je tabela u koju se smeštaju podaci o merenju, sastoji se iz ISTORIJA\_MERENJA\_ID, id, DATUM\_I\_VREME\_IM, VREDNOST, TIP\_INSULINA, TIP\_UNOSA.

**ISTORIJA\_MERENJA\_ID** - jedinstveni broj unosa istorije merenja

**id** - po broju se zna koji je korisnik izvršio unos

**DATUM\_I\_VREME\_IM** - datum i vreme unosa

**VREDNOST** - vrednost unosa insulina

**TIP\_INSULINA** - odabir iz padajuće liste za vrstu inslunina

**TIP\_UNOSA** - postoje dva tipa unosa, insulin i glikemija

+ Options										
	← T →		<b>ISTORIJA_MERENJA_ID</b>	<b>id</b>	<b>DATUM_I_VREME_IM</b>	<b>VREDNOST</b>	<b>TIP_INSULINA</b>	<b>TIP_UNOSA</b>		
<input type="checkbox"/>					4	1	2018-01-01 01:00:00	1.00	Brzodelujuci	Insulin
<input type="checkbox"/>					5	1	2018-12-13 12:31:00	2.00	Brzodelujuci	Glikemija
<input type="checkbox"/>					6	1	2018-12-13 12:31:00	3.00	Brzodelujuci	Glikemija
<input type="checkbox"/>					7	1	0000-00-00 00:00:00	3.00	Brzodelujuci	Insulin
<input type="checkbox"/>					8	1	2018-12-13 14:13:00	3.00	Brzodelujuci	Insulin
<input type="checkbox"/>					9	1	2018-12-13 14:13:00	3.00	Brzodelujuci	Insulin
<input type="checkbox"/>					10	1	2018-12-13 14:13:00	11.00	Brzodelujuci	Insulin
<input type="checkbox"/>					11	1	2018-12-13 14:13:00	13.00	Sporo delujuci	Insulin

Slika 29 - tabela istorija merenja

Tabela kalorije se sastoji od podataka vezanih za unos kalorija, sastoji se iz: id\_kalorije, id, datum, vreme, vrednost i ime\_namirnice

**id\_kalorije** - jedinstveni broj unosa kalorija

**id** - po broju se zna koji je korisnik izvršio unos

**datum** - datum unosa

**vreme** - vreme unosa

**vrednost** - vrednost unosa insulina

**ime\_namirnice** - unešeno ime namirnice

+ Options								
	← T →		id_kalorije	id	datum	vreme	vrednost	ime_namirnice
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	10	2018-07-01 14:22:00	11	Jabuka
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	10	2018-07-01 14:22:00	22	Gibanica
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	10	2222-02-22 12:21:00	200	Riba
<input type="checkbox"/>	 Edit	 Copy	 Delete	12	1	2018-08-23 11:11:00	299	riba

Slika 30 - tabela kalorije

Tabela korisnik se sastoji od podataka vezanih za osnovne podatke o korisniku kao i njegove kredencijale. Sastoji se iz: id, firstname, lastname, username, password i token.

**id** - po broju se zna koji je korisnik

**firstname** - ime korisnika

**lastname** - prezime korisnika

**username** - korisničko ime korisnika

**password** - enkriptovana lozinka korisnika

**token** - jedinstven token dodeljen svakom korisniku

+ Options								
	← T →		id	firstname	lastname	username	password	token
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Milos	Atanackovic	milosata	efa739c244f18c1c3788c33918a3a070 a76ba0073732b2ac2e1e07b0e2f94107781bcc11
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Admin	admin	21232f297a57a5a743894a0e4a801fc3 34b07f8ccfb6a8ba5476c36574ada213cc49c7	
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	Milos	Atanackovic	milos	ffecba6889b7f73d28b55794d9727593 23c4ec061fa23b21125201ee324dd58e91cce869
<input type="checkbox"/>	 Edit	 Copy	 Delete	13	Igor12	Igor12	Igor12	9d0eca871b834b6edcedbb0403989102 34020f9972c5e1a0d035632f2ed576466f1c60ba
<input type="checkbox"/>	 Edit	 Copy	 Delete	16	Milos	Atanackovic	ataata	dabfd8ac2a97138a460dc7887412206a 4fd03b28a72403b2c945d9d5119465d8d871af1e
<input type="checkbox"/>	 Edit	 Copy	 Delete	17	dasjdsanjk	dsakjndanskj	atatali	236684963d4267a191a21979fdad3c1e e71d3d77125532d60cacbb243a96e6078893d237

Slika 31 - tabela korisnik

Tabela namirnice sadrži podatke unosa namirnica. Sastoji se iz id\_namirnice, id, vrednost i ime\_namirnice.

**id\_namirnice**- jedinstveni broj namirnice

**id** - po broju se zna koji je korisnik izvršio unos

**vrednost** - vrednost namirnice

**ime\_namirnice** - naziv namirnice

			<b>id_namirnice</b>	<b>id</b>	<b>vrednost</b>	<b>ime_namirnice</b>
<input type="checkbox"/>	Edit	Copy	Delete	32	16	0 null
<input type="checkbox"/>	Edit	Copy	Delete	33	16	0 null
<input type="checkbox"/>	Edit	Copy	Delete	34	16	0 null

Slika 32 - tabela namirnice

## Testiranje sistema

### Funkcionalno testiranje

Funkcionalno testiranje je vrsta testiranja softvera gde je sistem testiran po funkcionalnim zahtevima. Testira se tako što se unose podaci i ispituj se izlaz. Ovim tipom testiranja, osiguravamo zahteve u radu aplikacije. Funkcionalnim testiranjem, koristimo sistem i imamo mogućnost da da uočimo propuste u sistemu.

#### Testovi slučajeva

Testiranje funkcionalnosti izvršava se po slučajevima, te pisanjem slučajeva može se otikriti propust u ispunjavanju funkcionalnih zahteva.

Naziv	Registracija korisnika
Opis	Korisnik treba da bude u mogućnosti da se registruje  Preduslovi: Korisnik mora da bude na stranici za registraciju korisnika
Testni koraci	<ol style="list-style-type: none"><li>Pristup početnoj stranici</li><li>Pristup stranici Registruj se</li><li>Unos imena</li><li>Unos prezimena</li><li>Unos korisničkog imena</li><li>Unos lozinke</li><li>Klik na dugme Registracija</li></ol>
Očekivani rezultat	Uspešno registrovan novi korisnik

Naziv	Autentifikacija korisnika
Opis	Korisnik treba da bude u mogućnosti da se prijavi na sistem sa kredencijalima koje je uneo na stranici registracija  Preduslovi: Korisnik mora da bude registrovan
Testni koraci	<ol style="list-style-type: none"><li>Pristup stranici Prijavi se</li><li>Unos korisničkog imena</li><li>Unos lozinke</li><li>Klik na dugme Prijava</li></ol>
Očekivani rezultat	Uspešno je prijavljen korisnik

<b>Naziv</b>	Testiranje svake stranice iz header-a
<b>Opis</b>	<p>Korisnik treba da bude u mogućnosti da mu se nakon klik na određenu stranicu, ta stranica prikaže</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Unos namirnica</li><li>2. Pristup stranici Unos kalorija</li><li>3. Pristup stranici Unos insulina</li><li>4. Pristup stranici Unos glikemije</li><li>5. Pristup stranici Dnevnik</li></ol>
<b>Očekivani rezultat</b>	Uspešno je prikazana svaka stranica kojoj je korisnik pristupio
<b>Naziv</b>	Testiranja unosa namirnice
<b>Opis</b>	<p>Korisnik treba da bude u mogućnosti da nakon unosa podataka o namirnici, vidi podatke u bazi</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Unos namirnica</li><li>2. Unos imena namirnice</li><li>3. Unos kalorijske vrednosti namirnice</li><li>4. Klik na dugme unesi</li><li>5. Pregled da li uneti podaci stoje u bazi</li></ol>
<b>Očekivani rezultat</b>	Uspešan unos imena i kalorijske vrednosti namirnice

<b>Naziv</b>	Testiranja unosa kalorija
<b>Opis</b>	<p>Korisnik treba da bude u mogućnosti da nakon unosa kalorija vidi podatke u bazi</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Unos kalorija</li><li>2. Iz padajućeg menija biranja imena namirnice</li><li>3. Unos kalorijske vrednosti namirnice</li><li>4. Unos datuma</li><li>5. Unos vremena unosa</li><li>6. Klik na dugme unesi</li><li>7. Pregled da li uneti podaci stoje u bazi</li></ol>
<b>Očekivani rezultat</b>	Uspešan unos kalorija
<b>Naziv</b>	Testiranja unosa insulina
<b>Opis</b>	<p>Korisnik treba da bude u mogućnosti da nakon unosa insulina vidi podatke u bazi</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Unos insulinina</li><li>2. Iz padajućeg menija biranje tipa insulinina</li><li>3. Unos vrednosti insulinina</li><li>4. Unos datuma i vremena unosa</li><li>5. Klik na dugme unesi</li><li>6. Pregled da li uneti podaci stoje u bazi</li></ol>
<b>Očekivani rezultat</b>	Uspešan unos insulinina

<b>Naziv</b>	Testiranja unosa glikemije
<b>Opis</b>	<p>Korisnik treba da bude u mogućnosti da nakon unosa glikemije vidi podatke u bazi</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Unos glikemije</li><li>2. Unos vrednosti glikemije</li><li>3. Unos datuma i vremena unosa</li><li>4. Klik na dugme unesi</li><li>5. Pregled da li uneti podaci stoje u bazi</li></ol>
<b>Očekivani rezultat</b>	Uspešan unos glikemije
<b>Naziv</b>	Testiranja dnevnika, dnevno
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u dnevno</li><li>3. Pregled dijagrama</li></ol>
<b>Očekivani rezultat</b>	Uspešan pregled podataka

<b>Naziv</b>	Brisanje unosa dnevnih podataka
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun, te imamo podatke koje možemo da obrišemo</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u dnevno</li><li>3. Odabir podatka koji želimo da izbrišemo</li><li>4. Klik na taster obriši</li><li>5. Pregleda da li je podatak izbrisan</li></ol>
<b>Očekivani rezultat</b>	Uspešano obrisan željeni podatak
<b>Naziv</b>	Testiranja dnevnika, nedeljno
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u nedeljno</li><li>3. Pregled dijagrama</li></ol>
<b>Očekivani rezultat</b>	Uspešan pregled podataka

<b>Naziv</b>	Brisanje unosa nedeljnih podataka
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun, te imamo podatke koje možemo da obrišemo</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u nedeljno</li><li>3. Odabir podatka koji želimo da izbrišemo</li><li>4. Klik na taster obriši</li><li>5. Pregleda da li je podatak izbrisан</li></ol>
<b>Očekivani rezultat</b>	Uspešano obrisan željeni podatak
<b>Naziv</b>	Testiranja dnevnika, mesečno
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u mesečno</li><li>3. Pregled dijagrama</li></ol>
<b>Očekivani rezultat</b>	Uspešan pregled podataka

<b>Naziv</b>	Brisanje unosa mečnih podataka
<b>Opis</b>	<p>Na korisničkom profilu mora da postoji unete namirnice, kalorije, insulin i glikemija kako bi dnevnik bio potpun, te imamo podatke koje možemo da obrišemo</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem.</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici Dnevnik</li><li>2. Pristup tab-u mesečno</li><li>3. Odabir podatka koji želimo da izbrišemo</li><li>4. Klik na taster obriši</li><li>5. Pregleda da li je podatak izbrisani</li></ol>
<b>Očekivani rezultat</b>	Uspešano obrisan željeni podatak
<b>Naziv</b>	Pristup admin panel-u
<b>Opis</b>	<p>Korisnik pristupa administratorskom panelu</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem kao administrator</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici za unos kredencijala</li><li>2. Unos administratorskih kredencijala</li><li>3. Uspešan pristup admin panel-u</li></ol>
<b>Očekivani rezultat</b>	Uspešan pristup administratorskom panel-u

<b>Naziv</b>	Brisanje korisnika iz amin panel-a
<b>Opis</b>	<p>Korisnik pristupa administratorskom panelu, kako bi mogao da manipuliše korisnicima.</p> <p>Preduslov: Korisnik mora da bude prijavljen na sistem kao administrator</p>
<b>Testni koraci</b>	<ol style="list-style-type: none"><li>1. Pristup stranici za unos kredencijala</li><li>2. Unos administratorskih kredencijala</li><li>3. Uspešan pristup admin panel-u</li><li>4. Odabir korisnika za brisanje</li><li>5. Klik dugmeta obriši</li><li>6. Provera u bazi da li je korisnik obrisan</li></ol>
<b>Očekivani rezultat</b>	Uspešanno brisanje korisnika iz baze podataka

## Jedinično(Unit) testiranje

Jedinično testiranje osigurava da se moduli softverskog sistema testiraju, i da svaki radi ispravno. Jedinično testiranje ipak ne garantuje ispravan rad modula kada se integrisu u sistem. Kod jediničnog testiranja, svaka jedinica koda može posebno da se testira, pri čemu jedinica koda može da bude funkcija, metoda, klasa, operacija, struktura podataka. Ovim načinom testiranja ne utvrđujemo kvalitet koda, već samo njegove funkcionalnosti. Jedinični test ispituje određenu metodu sa određenim uslovima, dokazujemo da ono deo koda radi, ono za šta je namenjen. Ovakvim vidom testiranja smanjujemo vreme debugging-a. Po određenom uslovu koji smo definisali, testovi mogu da prođu ili padnu, developer mora da pronađe grešku u kodu ako je test pao. Pre testiranja, moramo znati šta metoda radi, i šta očekujemo da ona uradi. U angular cli, koristi se Karma + Jasmine za testiranje. Test se deli iz dva dela, "describe" i "it". Funkciju "describe" prima dva parametra: string i funkciju. String je ime ili titula za specifikaciju paketa .Funkcija je blok koda koji implementira paket. Specifikacije su definisane zvanjem globalne funkcije "it", koja, kao i "describe" prima dva parametra: string i funkciju. String je titula specifikacije, a funkcija je specifikacija ili test. Specifikacija može da sadrži jedan ili više očekivanja tj. expect-a (eng. Expectation) koji testiraju stanje koda. Expect u Jasmine je tvrdnja koja je true ili false. Specifikacija sa svim true expect-ima je prolazna specifikacija. Specifikacija sa jednim ili više false expect-ima je specifikacija sa greškom.

```
import {async, ComponentFixture, TestBed} from '@angular/core/testing';
import {DnevniUnosGlikemijeComponent} from './dnevni-unos-glikemije.component';
// ...
describe( description: 'DnevniUnosGlikemijeComponent', specDefinitions: () => {
  let component: DnevniUnosGlikemijeComponent;
  let fixture: ComponentFixture<DnevniUnosGlikemijeComponent>;
  // ...
  beforeEach(async( fn: () => {
    TestBed.configureTestingModule( moduleDef: {
      declarations: [ DnevniUnosGlikemijeComponent ]
    })
    .compileComponents();
  }));
  // ...
  beforeEach( action: () => {
    fixture = TestBed.createComponent(DnevniUnosGlikemijeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
  it( expectation: 'should create', assertion: () => {
    expect(component).toBeTruthy();
  });
});
```

Slika 33 - Test kreiranja komponenti na Stranici Unos glikemije

U ovom testu, pozvali smo metodu za kompajlovanja komponenata, pre kompajlovanja komponenata, karma i jasmin kreiraju komponentu za dnevni unos glikemije. Ako je komponenta kreirana, ovaj test je prošao. Da smo pozvali metodu `toBeFalsy()`, a da se komponenta kreirala, test bi pao jer su komponente kreirane.

```
describe( description: 'LoginComponent', specDefinitions: () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  // ...
  it( expectation: 'form invalid when empty', assertion: () => {
    expect(LoginComponent.form.valid).toBeFalsy();
  });
});
```

Slika 34 - Testiranje prazne forme

Na login stranici, testiravamo login formu, da li je prazna. Ovaj test prolazi, samo ako forma bude prazna zbog pozivanja metode `toBeFalsy()`.

```
describe( description: 'LoginComponent', specDefinitions: () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;

  it( expectation: 'email field validity', assertion: () => {
    let username = LoginComponent.form.controls['username']; (1)
    expect(username.valid).toBeFalsy(); (2)
  });
});
```

Slika 35 - Testiranje username u formi

U ovom slučaju, proveravamo validnost korisničkog imena. Test je prošao ako forma nije validna.

```
import { LoginComponent } from './login.component';

describe( description: 'LoginComponent', specDefinitions: () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;

  it( expectation: 'submitting a form emits a user', assertion: () => {
    expect(component.form.valid).toBeFalsy();
    component.form.controls['email'].setValue("test@test.com");
    component.form.controls['password'].setValue("123456789");
    expect(component.form.valid).toBeTruthy();

    let user: User;
    // Subscribe to the Observable and store the user in a local variable.
    component.loggedIn.subscribe((value) => user = value);

    // Trigger the login function
    component.login();

    // Now we can check to make sure the emitted value is correct
    expect(user.username).toBe( expected: "milos.atanackovic.1542@metropolitan.ac.rs");
    expect(user.password).toBe( expected: "123456789");
  });
});
```

Slika 36 - Testiranje više elemenata na login stranici

U ovom slučaju, pored email-a dodajemo i polje password koje se testirava. Za username i password postavljamo uslove, i imamo unete vrednosti šta očekujemo na kraju testa. U ovom slučaju, nakon pozivanja metode login(), ispunjavaju se vrednosti u polja, i test proverava da li su vrednosti dobre.

```
describe( description: 'LoginComponent', specDefinitions: () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  let input: HTMLInputElement;

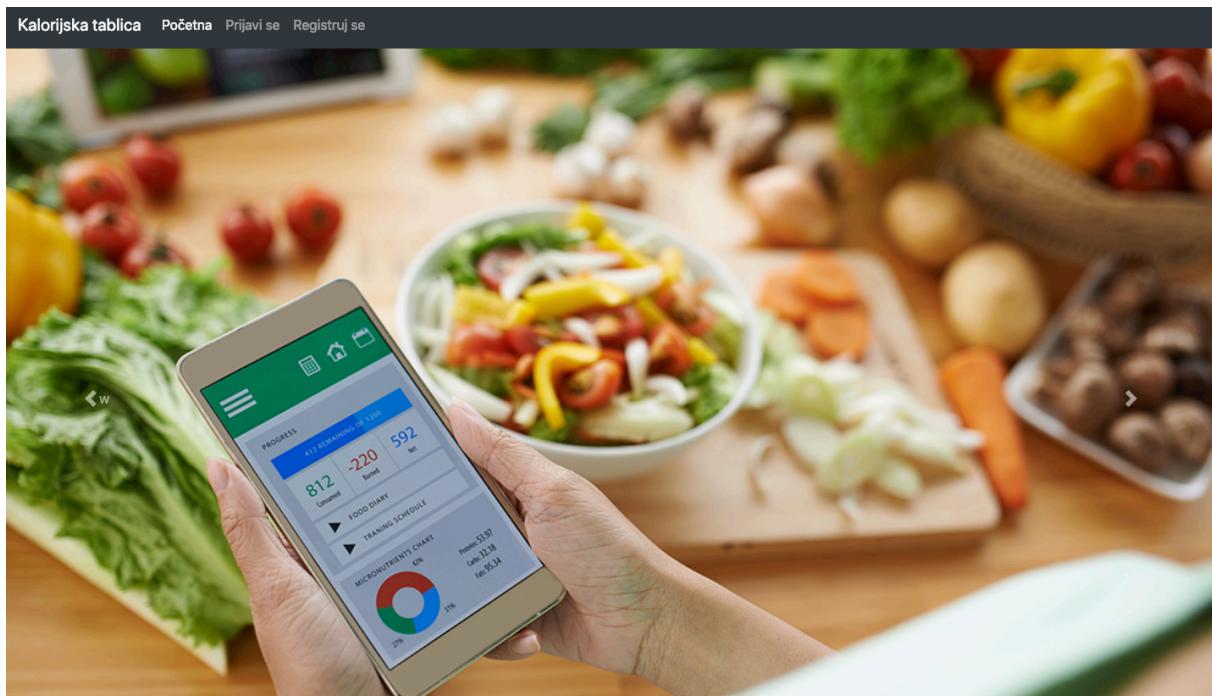
  it('should render "test input" into the div', () => {
    component.input = 'test input';
    component.processInput();
    fixture.detectChanges();
    expect(fixture.nativeElement.querySelector('div').innerText).toEqual('test input');
  });
});
```

Slika 37 - Testiranej pojedinačnog input polja

Nekad u aplikajama obično input polje može da napravi problem. Upisivanjem vrednosti, možemo da proverimo da li se iz dom elementa kupe određeni podaci.

# Korisničko uputstvo

Nakog pristupa aplikaciji na domenu, trenutno na lokalnom serveru, prikazuje nam se početna stranica. Iz nje, možemo da se registrujemo ili da se prijavimo na aplikaciju.



Slika 38 - Početna stranica

Klikom na stranicu prijavi se, prikazuju se dva polja za unos, unosom kredencijala pristupamo aplikaciji.

Kalorijska tablica Početna Prijavi se Registruj se

Username:

Password:

Slika 39 - Stranica za prijavljivanje na sistem

U slučaju da je posetilac potencijalni korisnik, i hoće da se registruje na sistem, pritiskom na stranicu registruj se, pristupa se stranici za registraciju. Unosom podataka, posetilac postaje korisnik sistema.

Kalorijska tablica Početna Prijavi se Registruj se

First name:  Last name:   
Username:  Password:

Slika 40 - Stranica za registraciju

Nakon unosa tačnih kredencijala, korisnik je uspešno prijavljen na sistem, i može da koristi sistem. Sljedeća slika je prikaz stranice za unos namirnica. Unosom u polja korisnik dodaje namirnicu.

Kalorijska tablica   Početna   Unos namirnica   Unos kalorija   Unos insulina   Unos glikemije   Dnevnik   Odjavi se

Ime namirnice	Kalorijska vrednost
<input type="text" value="ime namirnice"/>	<input type="text" value="broj kalorija"/>
<b>Unesi</b>	
<a href="#">Meso i mesni proizvodi</a> <a href="#">Riba</a> <a href="#">Mleko i mlečne prerađevine</a> <a href="#">Žitarice</a> <a href="#">Povrće</a> <a href="#">Voće</a>	

Slika 41 - Stranica za unos namirnica

Ispod dugmeta unos, nalaze se tabovi sa kalorijskim tablicama za odredjene namirnice po kategorijama. Odabirom kategorije, prikazuju se sve namirnice iz te kategorije kao i njihove kalorijske vrednosti.

Kalorijska tablica   Početna   **Unos namirnica**   Unos kalorija   Unos insulina   Unos glikemije   Dnevnik   Odjavi se

Ime namirnice	Kalorijska vrednost
<input type="text" value="ime namirnice"/>	<input type="text" value="broj kalorija"/>
<b>Unesi</b>	
<a href="#">Meso i mesni proizvodi</a> <a href="#">Riba</a> <a href="#">Mleko i mlečne prerađevine</a> <a href="#">Žitarice</a> <a href="#">Povrće</a> <a href="#">Voće</a>	

**MESO I MESNE PRERAĐEVINE**

Hrana	Kalorije	Proteini	Ugljeni hidrati	Masti
Bubrezi (teleći)	121	15	1	6
Dimljeni kare	272	24	2	16
Divlja svinja	104	21	0	2
Govedina mršava	119	21	4	1
Govedina srednje masna	155	20	1	8
Govedina masna	307	19	1	25
Govedina jako masna	410	14	0	39
Govedina u konzervi	223	26	0	14
Govedi file	138	16	0	8
Goveđe meso za gulaš	155	21	0	8
Goveđe meso za rolade	121	20	2	5
Goveđe meso za juhu	223	19	0	17
Goveda jetra	121	20	2	3
Hrenovke (govedina + svinjetina)	320	11	2	29
Hrenovke (pileće)	258	13	7	20
Hladetina	239	43	1	7
Janjeći kotlet	348	19	0	17
Janjetina mršava	207	19	0	13
Janjetina srednje masna	250	18	0	18

Slika 42 - Stranica za unos namirnica, kategorije

Na stranici unos kalorija, biramo prethodno unetu namirnicu iz padajućeg menija, i unosimo tačan unos, i količinu unetih kalorija.

Kalorijska tablica Početna Unos namirnica **Unos kalorija** Unos insulina Unos glikemije Dnevnik Odjavi se

Ime namirnice	Kalorijska vrednost
Jaja	200
Datum unosa	Vreme unosa
01/14/2019	02:31 PM

Slika 43 - Stranica za unos kalorija

Drugi deo aplikacije je najviše namenjen dijabetičarima, te imam unos insulina i unos glikemije.

Na stranici unos inulina, biramo tip insulina (brzodelujući ili sporodelujući), vrednost i vreme unosa.

Kalorijska tablica Početna Unos namirnica **Unos kalorija** Unos insulina Unos glikemije Dnevnik Odjavi se

### Unos insulina

Tip insulina

Brzodeljujuci

20

Datum i vreme unosa

01/14/2019, 03:21 PM

**Unesi**

Slika 44 - Stranica za unos insulina

Na stranici unos glikemije, unosimo vrednost i vreme i datum unosa.

Kalorijska tablica Početna Unos namirnica **Unos kalorija** Unos insulina Unos glikemije Dnevnik Odjavi se

### Unos insulina

2

Vreme i datum unosa

01/14/2019, 02:31 AM

**Unesi**

Slika 45 - Stranica za unos glikemije

Da bi videli sve podatke za unoše, pristupamo stranici dnevnik.

Kalorijska tablica Početna Unos namirnica Unos kalorija Unos insulina Unos glikemije Dnevnik Odjavi se

### Tablica kalorija i namirnica

U ovoj tablici se nalaze dnevni unosi kalorija i namirnica

Pretraga namirnica

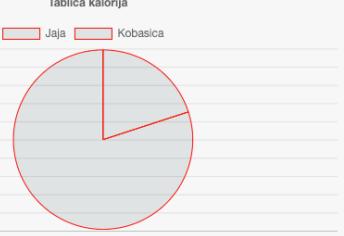
Dnevno Nedeljno Mesecno

#### DNEVNO

Ime namirnice	Kalorijska vrednost	Datum	Vreme	Brisanje unosa
Jaja	50	2019-01-14	12:33:00	Obrisu unos
Kobasica	200	2019-01-14	01:12:00	Obrisu unos

Trenutni unos kalorija: 250 kcal

Tablica kalorija



Jaja      Kobasica

Tablica kalorija



Broj kalorija

2019-01-14      2019-01-14

Slika 46 - Dnevnik

Na stranici dnevnik, nalaze se dnevni, nedeljni i mesečni izveštaj. U tabeli, vidimo namirnice koje su unete, vreme, ime namirnice i kalorijsku vrednost. U gornjem levom uglu, možemo da vršimo pretragu namirnica po imenu. Ispod u tabeli, na kraju imamo polje brisanje unosa, klikom na Obrisu unos, brišemo izabrani unos. Ispod tabele, imamo polje sa trenutnim unosom kalorija, gde se sabiraju sve unete kalorijske vrednosti. Ispod trenutnog unosa kalorija, nalaze se dva char-ta sa podacima. U prvom, prikazane su namirnice, unosi, istih, dok je u drugom prikazan datum, i namirnice po datumu.

Unosom administratorskih kredencijala, sistem prikazuje admin panel, stranicu samo za administratora. Na toj stranici je moguće brisanje korisnika.

Kalorijska tablica Prijavi se Registruj se

Username:

Password:

**Login**

Slika 47 - Unos administratorskih kredencijala

Nakon unosa kredencijala, sistem nas automacki vodi na stranicu Korisnici. Na stranici korisnici nam se prikazuju spisak svih korisnika, klikom na dugme izbriši korisnika, brišemo izabranog korisnika iz sistema.

## Korisnici

U ovoj tabeli se nalazi spisak korisnika

ID	Ime	Prezime	Korisnicko ime	
1	Milos	Atanackovic	milosata	Obrisni korisnika
2	Admin	Admin	admin	Obrisni korisnika
10	Milos	Atanackovic	milos	Obrisni korisnika
13	Igor12	Igor12	Igor12	Obrisni korisnika
16	Milos	Atanackovic	ataata	Obrisni korisnika
19	atineli	atineli	atineli	Obrisni korisnika

Slika 48 - Stranica korisnici

## Zaključak

Kroz ovaj diplomski rad koriscenje su tehnologije kao sto su Angular za front-end i php- za beckend. Posto je web aplikacija, bice postavljena na određeni domen i moće da se globalno koristi.

Sistem je napravljen, tako da mu je moguće pristupiti preko svakog uređaja jer je user friendly, potrebno je samo da uređaj bude priljučen na internet. Što se tiče veličine ekrana uređaja sa kojeg se pristupa, sistem se automatski adaptira svakom tipu ekrana, jer je korišćen bootstrap grid sistem.

U budućnosti je planirana nadgradnja celog sistema kako front tako i beckend dela radi boljeg security-ja. U diplomskom je opisana aplikacija, definisani funkcionalni i nefunkcionalni zahtevi, rađeni testovi, kao i opisan princip izrade projekta. Na kraju je prikazan sam rad aplikacije u vidu uputstva kao prikaz aplikacije preko slika.

Cilj aplikacije je pomoći dijabeticarima i zdravim korisnicima da vode evidenciju o unosima namirnica.

## Literatura

- [1] LAMS – <http://lams.metropolitan.ac.rs:8080/lams/> \_Predavanja I vežbe
- [2] W3C - <http://w3schools.com>
- [3] CLI ANGULAR - <https://cli.angular.io>
- [4] Stack Overflow - <https://stackoverflow.com>
- [5] Bootstrap - <http://getbootstrap.com/>
- [5] Angular: Up and Running: Learning Angular, Step by Step, Shyam Sashadri
- [6] Programming PHP, Rasmus Lerdorf
- [7] **XAMPP Documentation** - [www.apachefriends.org/docs/](http://www.apachefriends.org/docs/)
- [8] WebStorm, Stefan rosca
- [9] PhpStorm Cookbook, Ankur Kumar