

**INTEGRACIJA VEB I MOBILNIH APLIKACIJA UPOTREBOM SISTEMA ZA RAZMENU PORUKA****INTEGRATION OF WEB AND MOBILE APPLICATIONS USING MESSAGING SYSTEMS**

Milica Kovačević, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – RAČUNARSTVO I AUTOMATIKA**

**Kratak sadržaj** – Rad prikazuje razvoj veb i mobilne aplikacije uz podršku slanja notifikacija korisnicima na različitim platformama, koje je realizovano pomoću Firebase Cloud Messaging sistema. Kao primer razvijena je aplikacija HealthZone, fitnes aplikacija koja nudi korisnicima online saradnju. Ideja aplikacije je da trener kreira plan treninga za svoje klijente. Za implementaciju serverskog dela korišćen je Express framework, dok je za implementaciju klijentskog dela veb aplikacije korišćen Angular framewrok. Mobilna aplikacija implementirana je kao aplikacija za mobilne uređaje sa Android operativnim sistemom. Za skladištenje podataka korišćena je nerelaciona MongoDB baza podataka.

**Ključne reči:** JavaScript, Angular, Express, Android, MongoDB, Mongoose, FCM

**Abstract** – The aim of these paper is to develop a web and mobile application with the support of sending notifications to users on different platforms, which was realized using the Firebase Cloud Messaging system. As an example, the HealthZone application was developed, a fitness application that offers users online collaboration. The idea of the application is for the trainer to create a training plan for his clients. The Express framework was used to implement the server part, while the Angular framework was used to implement the client part of the web application. The mobile application is implemented as an application for mobile devices with the Android operating system. A non-relational MongoDB database was used for data storage.

**Keywords:** JavaScript, Angular, Express, Android, MongoDB, Mongoose, FCM

**1. UVOD**

Ovaj rad će prikazati razvoj veb i mobilne aplikacije uz podršku slanja notifikacija korisnicima na različitim platformama, koje je realizovano pomoću Firebase Cloud Messaging sistema (FCM). FCM je besplatna usluga koja šalje poruke u realnom vremenu. Kao primer razvijena je aplikacija HealthZone koja spada u grupu fitnes aplikacija i nudi korisnicima online saradnju. Ideja aplikacije je da trener kreira plan treninga za svoje klijente.

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

Svaki trening treba da bude prilagođen potrebama klijenta, njegovoj formi i rezultatima koje klijent želi da postigne. Trening sadrži detaljna uputstva sa spiskom vežbi, brojem serija, brojem ponavljanja i opciono definisanim opterećenjem koje je potrebno uključiti kako bi se postigli određeni ciljevi i vremenskim intervalom u kom se vežba izvodi ali i ono najvažnije, slike za pravilno izvođenje vežbi.

**2. TEHNOLOGIJE**

U ovom poglavlju opisane su tehnologije pomoću kojih su implementirane veb i mobilna aplikacija. Za svaku tehnologiju navedene su samo osnovne karakteristike potrebne za dalje razumevanje projektnog zadatka.

**2.1. Angular**

Angular je besplatan, open-source JavaScript framework [1]. Koristi se za razvoj dinamičkih veb i mobilnih aplikacija. Omogućuje kreiranje aplikacija sa jednom stranom SPA – Single page Application. Angular aplikacije pišu se u programskom jeziku TypeScript. Angular aplikacija sastoji se iz modula. Modul predstavlja način grupisanja komponenti, servisa, direktiva i drugih fajlova koji su povezani, na takav način da predstavljaju logičku celinu.

Komponenta predstavlja glavni gradivni element Angular aplikacije. Svaka Angular aplikacija sadrži najmanje jednu komponentu. Komponentu možemo smatrati posebnim pogledom aplikacije sa sopstvenom logikom i podacima. Komponenta se definiše u okviru modula u kom se nalazi, i zbog toga se pre ključne reči class koristi ključna reč export. Dekoratoru @Component() prosleđuje se objekat sa sledećim svojstvima (listing 2):

- **selector** predstavlja naziv, na osnovu koga će komponenta biti prikazana u okviru neke druge komponente.
- **templateUrl** predstavlja putanju do HTML šablon koji definiše šta će biti prikazano na stranici.
- **stylesUrl** predstavlja putanju do CSS fajla u kom se definišu stilovi koji će biti primenjeni u HTML-u.

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-component',
  templateUrl: './app.component.html',
```

```

styleUrls: ['./app.component.css'] }}

export class AppComponent {
  text = 'Hello World!';
}

```

**Listing 1** – Primer root komponente

## 2.2. Bootstrap

Bootstrap je najpopularniji framework otvorenog koda za izradu web sajtova i mobilnih aplikacija [2]. Baziran je na HTML i CSS šablonima za tipografiju, kreiranju formulara, dugmadi, navigacionim i ostalim komponentama, kao i opcionim JavaScript dodacima. Za razliku od Angulara, Bootstrap je fokusiran na vizuelni deo i kao takav koristi se kao dodatak, radi poboljšanja korisničkog interfejsa i bržeg razvoja aplikacije.

## 2.3. PrimeNG

PrimeNG sadrži bogatu kolekciju komponenta korisničkog interfejsa za Angular, koje zadovoljavaju većinu korisničkih zahteva kao što su tabele, padajući meniji, dugmad, kalendar, poruke, obaveštenja i slično [3]. PrimeNG komponente podržavaju šablone pomoću kojih je moguće prilagoditi sadržaj komponente.

## 2.4. Express

Express je besplatan, open-source NodeJS framework koji se koristi za brzo i jednostavno kreiranje serverskih aplikacija [5]. Pruža mehanizam za jednostavno upravljanje HTTP zahteva koje klijentska aplikacija šalje. Podržava MVC (Model-View-Controller), veoma čestu arhitekturu za dizajn veb aplikacija. Podržan je na svim platformama (eng. cross-platform) i nije ograničen na određeni operativni sistem. Express aplikacija ne dolazi sa strogom strukturom fajlova i foldera, već daje slobodu da sami organizujemo strukturu aplikacije. Strukturiranje aplikacije na domenski povezane delove je najbolji pristup, jer na taj način inkapsuliramo delove aplikacije što olakšava dalji razvoj u budućnosti.

## 2.5. MongoDB Atlas baza podataka

MongoDB je vodeća NoSQL baza podataka koja čuva podatke kao JSON dokumente sa dinamičkim šemama. JSON (JavaScript Object Notation) je otvoreni standard zasnovan na tekstu, osmišljen za razmenu podataka koji su pogodni za čitanje ljudima [6]. Dokument baza podataka omogućuje da podaci mogu biti smešteni u ugnježđenom obliku do proizvoljne dubine. Dokument u Mongo bazi podataka se može posmatrati kao jedan red u tabeli relacione baze podataka.

Mongoose je objektni dokument mapper (eng. ODM - Object Document Mapper) koji olakšava korišćenje MongoDB-a, prevodenjem dokumenata iz MongoDB baze podataka u objekte koji se koriste u kodu. Mongoose koristi šeme za modelovanje podataka. Mongoose šema definiše strukturu podataka, predefinisane vrednosti, validatore i sl. Mongoose model je wrapper oko šeme, koji pruža interfejs bazi podataka za kreiranje, brisanje, ažuriranje, postavljanje upita i slično.

## 2.6. Firebase

Firebase je kompanija iz San Franciska koja pruža backend usluge i usluge računarstva u oblaku (eng. cloud

computing) [8]. Kompanija pruža veliki broj usluga za programere mobilnih i veb aplikacija, kao što su usluge baze podataka za rad u realnom vremenu i backend usluge, hoisting, firebase autentifikacija i druge. Firebase Cloud Messaging sistem omogućuje besplatnu razmenu poruka na različitim platformama u realnom vremenu.

## 2.7. Android

Za programiranje Android aplikacija koristi se programski jezik Java i Android SDK. Android SDK (Software Development Kit) je biblioteka koja sadrži skup razvojnih alata za izradu aplikacije, kao što su emulator Android uređaja i alati za testiranje i debugovanje. Okruženje koje se koristi za razvoj je Android Studio. Za kreiranje korisničkog interfejsa i kreiranje konfiguracionih fajlova koristi se XML jezik.

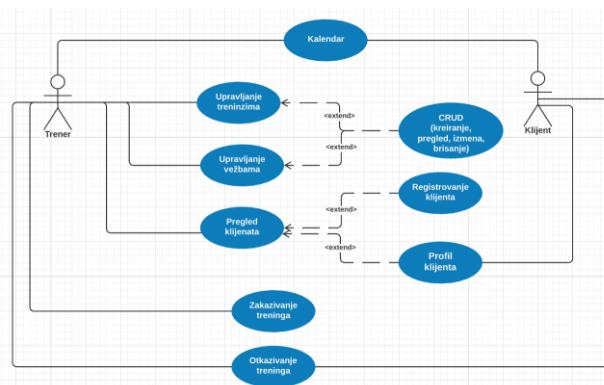
Android aplikacije se pišu u programskom jeziku Java. Prevedeni Java kod, zajedno sa svim datotekama, zapakovan je u datoteku sa sufiksom .apk. Ova datoteka služi za distribuiranje i instaliranje aplikacije na mobilnim uređajima. Aplikacijom se smatra sav kod koji se nalazi u .apk datoteci.

Prednosti Android aplikacije su te što pružaju kvalitetniji korisnički interfejs, poseduju bolje performanse i mogu upravljati resursima uređaja, za razliku od veb aplikacija koje se mogu izvršiti u veb pretraživačima uređaja.

## 3. SPECIFIKACIJA ZADATKA

HealthZone aplikacija spada u grupu fitnes aplikacija i nudi korisnicima online saradnju. Namenjena je personalnim trenerima i njihovim klijentima. Online trening podrazumeva da korisnik na svom uređaju ima sve podatke koji su potrebni za treniranje. Ideja aplikacije je da trener kreira plan treninga za svoje klijente. Svaki trening treba da bude prilagođen potrebama klijenta, njegovoj formi i rezultatima koje klijent želi da postigne. Trening sadrži detaljna uputstva sa spiskom vežbi, brojem serija, brojem ponavljanja i opciono definisanim opterećenjem koje je potrebno uključiti kako bi se postigli određeni ciljevi i vremenskim intervalom u kom se vežba izvodi ali i ono najvažnije, slike za pravilno izvođenje vežbi.

Na slici 1 prikazan je dijagram slučajeva korišćenja za prijavljenog trenera i klijenta.



**Slika 1** – Dijagram slučajeva korišćenja prijavljenog korisnika

## 4. IMPLEMENTACIJA

Realizovani sistem obuhvata veb i mobilnu aplikaciju. U narednim poglavljima biće ukratko opisane funkcionalnosti aplikacija iz ugla različitih tipova korisnika.

### 4.1. Veb aplikacija

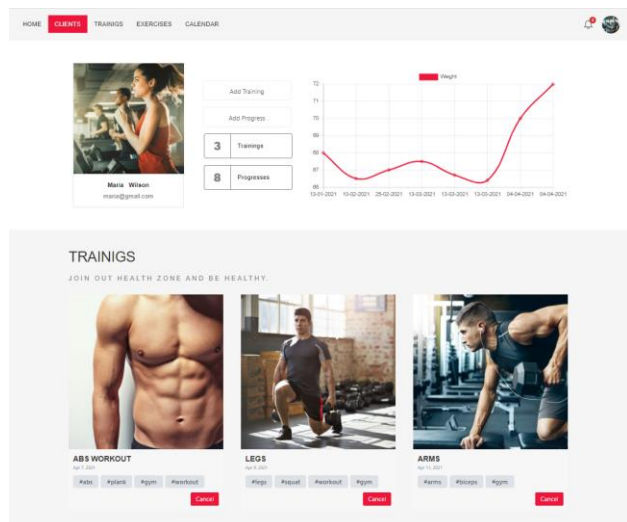
Veb aplikacija je aplikacija kojoj korisnici pristupaju preko internet mreže, koristeći veb pregledač (eng. browser). Funkcionišu tako što korisnik šalje zahteve veb serveru, putem interneta kroz veb pregledač. Veb server prosleđuje zahtev aplikaciji na serveru, koja izvršava taj zadatak, generiše rezultat i vraća odgovor nazad serveru i na kraju server prosleđuje rezultat korisniku u veb pregledaču.

HealthZone aplikacija namenjena je trenerima i klijentima. Korišćenje aplikacije omogućeno je samo registrovanim korisnicima. U zavisnosti od toga koji je tip korisnika prijavljen (trener ili klijent), navigacioni meni će sadržati različite linkove ka odgovarajućim stranicama. Trener ima opcije pregleda početne stranice ulogovanog klijenta, liste klijenata, liste treninga, liste vežbi i kalendara, dok klijent ima mogućnost da vidi početnu stranicu, svoj profil i kalendar.

Prilikom prijave na sistem serverskoj aplikaciji se šalju email adresa i lozinka. Ukoliko su uneti podaci ispravni klijentskoj aplikaciji se vraća odgovor koji sadrži token. Klijentska aplikacija token čuva u localstorage-u. Sve metode na serveru su zaštićene i nije im moguće pristupiti ako korisnik prethodno nije prijavljen na sistem, stoga je potrebno poslati token (dobijen prilikom prijave na sistem) prilikom svakog zahteva upućenog serverskoj aplikaciji.

Profil klijenta prikazan je na slici 2. Profil klijenta sadrži osnovne informacije o klijentu, opcije za zakazivanje treninga i dodavanje parametara napretka, prikaz broja zakazanih treninga, grafički prikaz parametra napretka gde su na x-osi prikazani datumi, a na y-osi numeričke vrednosti, i listu zakazanih treninga. Treninzi su prikazani u vidu kartica.

Trening sadrži osnovne informacije, naziv, sliku, datum kada se održava, tagove i dugme za opciju otkazivanja. Opcija otkazivanja treninga omogućena je klijentu i treneru. Klikom na dugme otkazi (eng. cancel), trening će biti otkazan i korisniku će se prikazati obaveštenje o uspešnosti.



Slika 2 – Profil klijenta

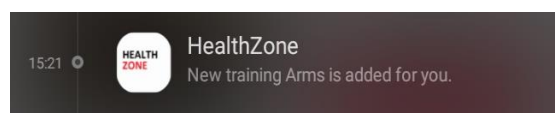
Dodavanje treninga za klijenta vrši se putem odgovarajućeg dijaloga, gde je potrebno izabrati datum i vreme, kao i trening iz padajućeg menija, koji sadrži listu prethodno kreiranih treninga za prijavljenog trenera. Ova opcija omogućena je samo treneru. Klikom na dugme potvrdi (eng. confirm), zahtev se prosleđuje serverskoj aplikaciji. Serverska aplikacija obrađuje zahtev, kreira i čuva trening u bazi podataka i nakon toga kreira notifikaciju za korisnika (slika 3).

Kreiraju se tri tipa notifikacija za klijenta za koga je trening zakazan.

Prva notifikacija obaveštava klijenta da je novi trening dodat, druga notifikacija šalje se klijentu sat vremena pred početak treninga kao podsetnik, dok se treća notifikacija šalje klijentu ukoliko je trening otkazan.

Slanje notifikacija implementirano je koristeći Firebase Cloud Messaging (FCM) koji pruža konekciju između serverske aplikacije i veb aplikacije ili mobilnog uređaja. FCM obaveštenja ili poruke može slati pojedinačnom uređaju ili grupi uređaja koji su pretplaćeni na primanje obaveštenja ili poruka.

Cloud Messaging funkcioniše tako što koristi Cloud Messaging Token, koji se čuva u bazi podataka a kreira se prilikom reistracije korisnika. Ovaj token se kasnije koristi za slanje poruka ka različitim uređajima. Brzina slanja poruka je veoma zadovoljavajuća kao i verovatnoća da će poruka biti isporučena.



Slika 3 – Notifikacija o zakazanom treningu

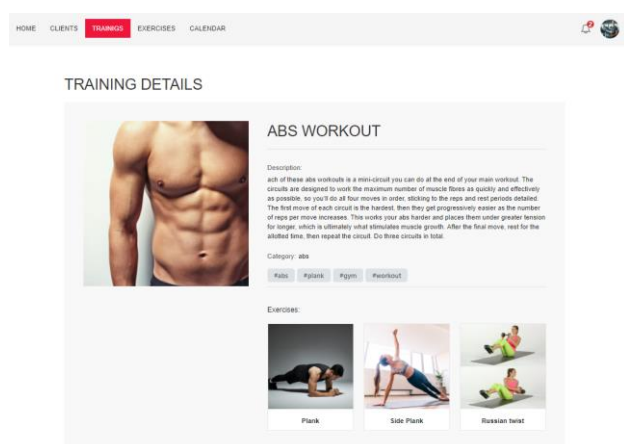
```
const notification = {
  token: registrationToken,
  title: "HealthZone",
  body: "New training "+training.name+" is added for
you."
};
firebaseAdmin.sendNotification(notification);
```

Listing 2 Deo koda koji prikazuje definisanje i slanje notifikacije nakon dodavanja treninga za klijenta

Kartica za prikaz treninga implementirana je kao generička komponenta. Koristi se jos za prikaz vežbi i rezervisanih treninga. Generička komponenta omogućuje da na osnovu ulaznih parametara koji joj se proslede prikaže različit sadržaj na ekranu. Na ovaj način prikazivanje podataka smešteno je u komponenti koja ne vodi računa o logici. Pored ulaznih parametara komponenta emituje događaje (eng. event) na klik određenog dugmeta i prosleđuje ga nazad komponenti koja je pozvala, koja dalje obrađuje taj događaj. Na taj način cela logika smeštena je u roditeljskoj komponenti.

Stranica za prikaz detalja odabranog treninga prikaza je na slici 4. Na ovoj stranici su prikazani svi detalji treninga kao i lista vežbi, koje su dodate u okviru treninga. Ova stranica dostupna je klijentima i trenerima. Klikom na karticu sa vežbom, koje su prikazane ispod osnovnih

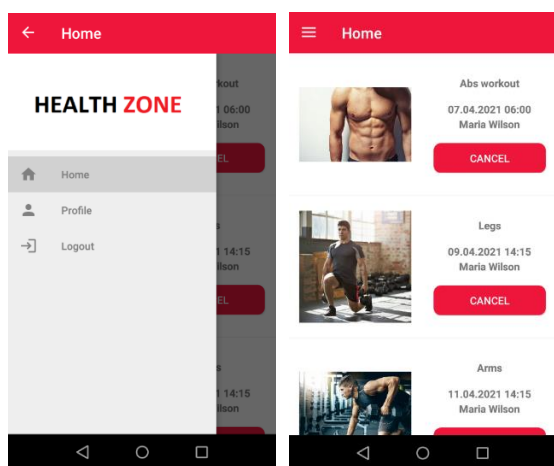
detalja o treningu, klijent će biti preusmeren na stranicu sa detaljima izabrane vežbe.



Slika 4 Prikaz detalja treninga

#### 4.1. Android aplikacija

Android aplikacija namenjena je mobilnim uređajima sa ekranom osetljivim na dodir kao što su mobilni telefoni i tablet uređaji. Izvršava se na Android operativnom sistemu. HealthZone spada u grupu android aplikacija koje se opciono mogu instalirati na mobilnom uređaju. Pristup aplikaciji omogućen je samo registrovanim korisnicima. Registrovanje korisnika može se izvršiti samo putem veb aplikacije. Android aplikacija podržava samo prijavu korisnika na sistem. Za razliku od veb aplikacije, android aplikacija podržava samo neke od funkcionalnosti sistema. Na slici 5 prikazan je navigacioni meni i početni ekran aplikacije. Navigacioni meni olakšava kretanje kroz aplikaciju. Sastoji se iz tri opcije: početni ekran, profil korisnika, i opcija za odjavu korisnika. Na početnom ekranu prikazana je lista zakazanih treninga za ulogovanog korisnika. Svaka kartica sarži samo osnovne informacije o treningu, naziv, datum i vreme održavanja i ime klijenta. Pored osnovnih informacija o treningu, postoji dugme za otkazivanje treninga.



Slika 5 Meni i početna stran

## 5. ZAKLJUČAK

Cilj ovog rada bio je kreiranje veb i mobilne aplikacije u okviru kojih je implementirano slanje različitih notifikacija korisnicima. Slanje notifikacija u realnom vremenu implementirano je korišćenjem Firebase Cloud Messaging sistema. Aplikacija nudi različite funkcionalnosti registrovanim korisnicima sistema za podršku online saradnje trenera i klijenta. Proširivanje i nadogradnja ove aplikacije podrazumevala bi dodavanje novih funkcionalnosti koje bi mogle unaprediti, olakšati i poboljšati saradnju između korisnika.

Aplikacije su razvijene u trenutno aktuelnim tehnologijama koje omogućuju laku i jednostavnu nadogradnju kao i održavanje. FCM sistem koji je korišćen za razmenu poruka na različitim platformama predstavlja trenutno najefikasnije rešenje za tu namenu. Stalni razvoj i održavanje pružaju mogućnost daljeg razvoja i nadogradnje kako veb tako i mobilne aplikacije.

## 6. LITERATURA

- [1] Angular <https://angular.io/docs>
- [2] Bootstrap <https://sr.wikipedia.org/sr-el/Bootstrap>
- [3] PrimeNG <https://www.primefaces.org/primeng>
- [4] Express <https://expressjs.com>
- [5] Express/Node uvod [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction)
- [6] MongoDB <https://sr.wikipedia.org/wiki/MongoDB>
- [7] Firebase <https://firebase.google.com/docs>
- [8] Firebase-wikipedija <https://sr.wikipedia.org/sr-el/Fajerbejs>
- [9] Android – wikipedija [https://sh.wikipedia.org/wiki/Android\\_\(operativni\\_sistem\)](https://sh.wikipedia.org/wiki/Android_(operativni_sistem))
- [10] Android <https://developer.android.com/docs>

### Kratka biografija:

**Milica Kovačević** rođena je 29. juna 1995. godine u Novom Sadu. Završila je prirodno-matematički smer gimnazije „Jovan Jovanović Zmaj“ u Novom Sadu. Po završetku srednje škole upisuje 2014.godine Fakultet tehničkih nauka u Novom Sadu, smer Računarstvo i automatika i na trećoj godini se opredeljuje za smer Primenjene računarske nauke i informatika. Diplomirala je 2018. godine i nakon toga upisuje master akademske studije na istom fakultetu na smeru za Računarstvo i automatiku, modul Elektronsko poslovanje. Položila je sve ispite predviđene planom i programom.

**Milan Vidaković** rođen je u Novom Sadu 1971. godine. Na Fakultetu tehničkih nauka u Novom Sadu završio je doktorske studije 2003. godine. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.