

**INTEGRACIJA FIREBASE CLOUD MESSAGING SISTEMA U VEB I MOBILNE APLIKACIJE****INTEGRATION OF FIREBASE CLOUD MESSAGING SYSTEM IN WEB AND MOBILE APPLICATIONS**Ilinka Kovačević, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

**Kratka sadržaj** – Zadatak rada predstavlja razvoj veb i mobilne aplikacije uz korišćenje Firebase Cloud Messaging (FCM), koji omogućava slanje poruka bez ikakvih troškova. Kao primer kreirana je aplikacija za teretane, koja nudi svojim klijentima mogućnost zakazivanje grupnih treninga uz stizanje notifikacija sat vremena pre zakazanog treninga. Klijentski deo veb aplikacije implementiran je pomoću Angular Framework-a, dok je serverski deo razvijen pomoću Express Framework-a. Mobilna aplikacija napravljena je isključivo za Android operativni sistem. Svi podaci se čuvaju u MongoDB.

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

**Abstract** – The aim of this paper is to develop a web and mobile application using the Firebase Cloud Messaging (FCM, which allows sending messages at no cost. As an example, an gym application is created, which offers its clients the possibility of scheduling group trainings with receiving notifications an hour before the scheduled training. The client part of the web application is implemented using the Angular Framework, while the server part is developed using the Express Framework. The mobile application is made exclusively for the Android operating system. All data is stored in MongoDB.

**Key words:** Angular, Express, Android, MongoDB, Mongoose, FCM

**1. UVOD**

Kada je reč o tehnologijama koje su korišćenje prilikom razvijanja aplikacije, serverski deo, koji je zajednički za veb i mobilnu aplikaciju, razvijen je upotrebom *Express Framework-a*. *Angular Framework*, kao jedan od najpopularnijih okruženja za razvoj veb aplikacija, korišćen je za izradu klijentskog dela veb aplikacije.

Mobilna aplikacija napravljena je isključivo za *Android* mobilni operativni sistem, za uređaje sa *Android 4.1* ili novijom verzijom zbog potrebe *FCM* klijenata kako bi uspostavili slanje poruka.

Svi podaci veb i mobilne aplikacije čuvaju se u *MongoDB*. Mapiranje objekata između *MongoDB-a* i serverskog dela ostvaruje se putem *Mongoose-a*.

**NAPOMENA:**

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

**2. TEHNOLOGIJE**

U ovom poglavlju osvrt je dat na tehnologije pomoću kojih su realizovane veb i mobilna aplikacija. Za svaku tehnologiju stavljen je fokus na bitne karakteristike koje su potrebne za dalje razumevanje implementirane aplikacije.

**2.1. Angular**

Angular 2+ predstavlja okvir i razvojnu platformu za stvaranje efikasnih klijentskih aplikacija na jednoj stranici (*SPA - Single Page Apps*) koristeći *HTML* i *TypeScript* [1]. Komponente predstavljaju osnovni gradivni element angular aplikacije. Svaka angular aplikacije ima bar jednu komponentu, takozvanu *root* komponentu (listing 1).

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
}
```

Listing 1 – Primer root komponente

Svaka komponenta sastoji se od *HTML* šablona koji definiše šta se prikazuje na stranici, *TypeScript* klase koja definiše ponašanje, *CSS* selektora koji definiše kako se komponenta koristi u šablonu i *CSS* fajla u kom su definisani stilovi primenjeni na *HTML* šablonu.

**2.2. Express**

*Express* je najpopularniji *Node.js* okvir, koji je posebno dizajniran za izgradnju veb i mobilnih aplikacija, predstavlja osnovnu biblioteku za brojne druge *Node.js* veb okvire. Izgrađen je na vrhu *Node.js* *http* modula i dodaje podršku za rutiranje. Omogućava programerima da kreiraju i održavaju server. Dolazi sa mnogim ugrađenim funkcijama, takođe ima i ogroman paket nezavisnih dodataka koje programeri mogu koristiti za pružanje bolje funkcionalnosti, poboljšanje brzine i povećavanje sigurnosti [2]. Programeri korsiite *Express* okvir za pisanje logike na strani servera u *JavaScript-u* za veb i mobilne aplikacije.

**2.3. Firebase Cloud Messaging**

*Firebase Cloud Messaging (FCM)* predstavlja rešenje za razmenu poruka na više platformi koje omogućava pouzdano slanje poruka bez ikakvih troškova. Koristeći *FCM* možete da šaljete poruke obaveštenja da biste podstakli ponovno angažovanje i zadržavanje korisnika

[3]. Pored toga FCM pruža mogućnost obaveštavanja klijentske aplikacije da su drugi podaci dostupni za sinhronizaciju, kao i razmenu trenutnih poruka pri čemu poruka može da sadrži teret veličine 4KB.

FCM pruža pouzdanu i efikasnu vezu između servera i uređaja koja omogućava da poruke i obaveštenja budu primljena i isporučena na *IOS* ili *Android* uređajima kao i na veb aplikaciji bez troškova.

## 2.4. Android

Android je mobilni operativni sistem kompanije Google, dizajniran za mobilne uređaje sa ekranima osljljivim na dodir kao što su pametni telefoni i tablet uređaji [4]. Android predstavlja softver otvorenog koda zasnovan na Linux jezgru, razvijen za širok spektar uređaja.

Android aplikacije mogu biti napisane pomoću jezika Java, C++ i Kotlin. Osnovni gradivni elementi Android aplikacije su komponente. Komponenta predstavlja ulaznu tačku kroz koju korisnik ili sistem mogu da pristupe aplikaciji. Postoje različite komponente kao što su aktivnosti, usluge i dobavljači sadržaja.

## 2.5. MongoDB

MongoDB je vodeća nerelaciona tj. *NoSQL* baza podataka. U skladu sa interesovanjem za alternativne sisteme za upravljanje bazama podataka, koji se razlikuju od relacionih baza podataka, pojavio se koncept takozvanih *NoSQL* baza podataka, koje ne koriste SQL (*Structured Query Language*) za povezivanje. *NoSQL* baze podataka su nerelacione, distribuirane, otvorenog koda i horizontalno skalabilne [5].

MongoDB 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. Dokumenti su organizovani unutar kolekcija pri čemu jedna kolekcija obično sadrži dokumente sa sličnom namenom. Dokument pored primitivnih tipova podržava nizove i ugneždene objekte kao vrednosti što poboljšava performanse prilikom čitanja i pisanja dokumenata.

MongoDB i Node.js se često koriste zajedno zbog zajedničke upotrebe JavaScripta-a. Mapiranje objekata između MongoDB-a i Node.js-a ostvaruje se putem Mongoose-a. Na listingu 2 dat je prikaz Mongoose šeme.

```
const mongoose = require('mongoose');
const classSchema = mongoose.Schema({
  name: { type: String, required: true},
  instructor: { type: String, required: true},
  location: { type: String, required: true},
  timeStart: { type: String, required: true},
  timeEnd: { type: String, required: true},
  description: { type: String, required: true}
});
module.exports = mongoose.model('class', classSchema);
```

Listing 2 – Primer Mongoose šeme

## 3. SPECIFIKACIJA ZADATAKA

Zadatak rada obuhvata kreiranje veb i mobilne aplikacije za zakazivanje grupnih treninga u datoj teretani. Aplikacija je razvijena tako da podržava komunikaciju sa

FCM sistemom koji služi za slanje obaveštenja korisnicima koji su zakazali trening.

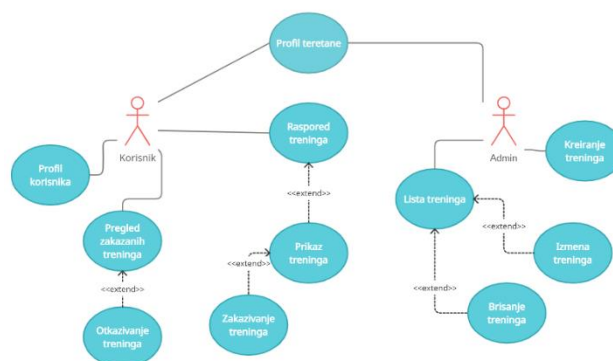
Aplikacija podržava dva tipa korisnika, a to su korisnici usluga koje teretana pruža i administrator sistema, koji održava aplikaciju. Mobilna aplikacija razvijena je samo za potrebe korisnika teretane, dok veb aplikaciji mogu da pristupe i administratori.

Prilikom dolaska na veb stranicu aplikacije korisnicima je omogućena prijava na sistem. Prijava se vrši putem email adrese i lozinke. Ukoliko neko od korisnika nema nalog nudi mu se opcija registracije na sistem.

Nakon uspešne autentifikacije, običnom korisniku se prikazuje raspored treninga za narednih sedam dana, pregled svih zakazanih treninga kao i istorija svih prethodnih zakazanih treninga koji su prošli, profil korisnika i profil teretane koji sadrži informacije o datoj teretani.

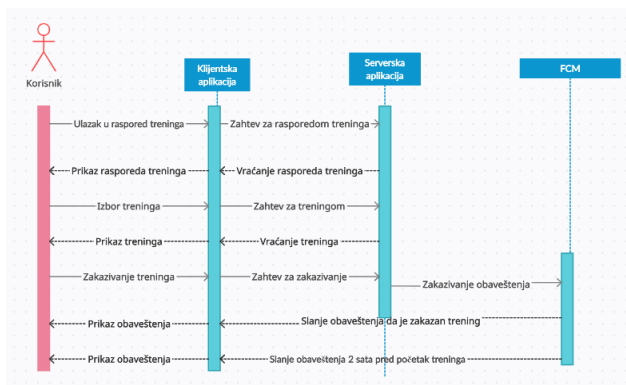
Korisnik ima mogućnost detaljnog prikaza svakog treninga koji se nalazi na rasporedu za narednih sedam dana, kao i mogućnost rezervisanja datog treninga. Korisnik pored rezervacije ima i opciju otkazivanja treninga.

Administratoru se nakon uspešne prijave na sistem, prikazuje lista svih treninga koji su kreirani, profil teretane koji može da uređuje i opcija za kreiranje novih treninga. Pored toga, administrator može da menja i uklanja postojeće treninge. Na slici 1 dat je prikaz dijagrama slučajeva korišćenja prijavljenog korisnika.



Slika 1. Dijagram slučajeva korišćenja prijavljenog korisnika

Na slici 2 dat je prikaz dijagrama sekvence za zakazivanje treninga. Sekvenca događaja počinje korisnikovim ulaskom u raspored treninga za narednih sedam dana. Nakon što serverska aplikacija vrati klijentskoj aplikaciji raspored treninga, korisnik bira trening koji želi da zakaže. Prikazuje mu se detaljan opis treninga i opcija da zakaže. Nakon što korisnik odabere da zakaže trening, klijentska aplikacija šalje zahtev za zakazivanje treninga serverskoj aplikaciji, koja zakazuje obaveštenja kod FCM. FCM šalje prvo obaveštenje klijentskoj aplikaciji da je trening uspešno zakazan i prikazuje to korisniku u vidu obaveštenja u Toolbar-u. FCM šalje drugo obaveštenje klijentskoj aplikaciji sat vremena pre zakazanog treninga kako bi podsetila korisnika na trening, obaveštenje je takođe prikazano u Toolbar-u



Slika 2. Dijagram sekvence

## 4. IMPLEMENTACIJA

Za potrebe rada razvijene su veb i mobilna aplikacija. Obe aplikacije sastoje se iz dva dela, klijentskog i serverskog. Serverski deo je zajednički za obe aplikacije. U daljem nastavku teksta biće detaljnije objašnjene implementacije obe aplikacije.

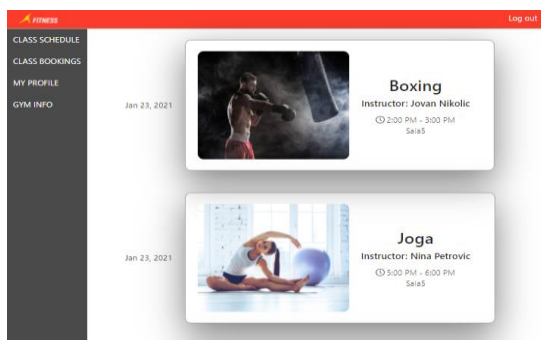
### 4.1. Klijentski deo veb aplikacije

Klijentski deo veb aplikacije, razvijen korišćenjem Angular Framework-a, podeljen je u tri modula. Prvi modul koji je takođe i podrazumevani modul predstavlja autentifikacioni modul. Rute iz ovog modula su dostupne prilikom dolaska na veb sajt, dostupne su samo neprijavljenom korisniku. Rute koje se nalaze u ovom modulu su rute za prijavu i registraciju na sistem. Prijavljen korisnik u zavisnosti od uloge može da pristupi ostalim modulima. Ukoliko je prijavljen običan korisnik, rutira se na modul app, a ukoliko je prijavljen administrator, rutira se na admin modul.

Nakon uspešne autentifikacije podaci o prijavljenom korisniku se čuvaju u *Local Storage*-u. Pored podataka o korisniku, čuva se i token koji se kreira od strane servera za svakog korisnika kada se uloguje, koji u sebi sadrži i rok važenja.

Kako ne bi moglo da se pristupi modulima koji dozvoljavaju pristup samo prijavljenim korisnicima, koriste se Angular-ovi zaštitnici ruta. U aplikaciji je za tu svrhu kreirana *AuthGuard* klasa koja implementira interfejs *CanActivate*. Svrha ove klase je da obavesti rutera na osnovu povratne vrednosti da li može da se pristupi traženoj ruti.

Ukoliko je prijavljen običan korisnik, nakon prijave prikazuje mu se početna strana, na kojoj je prikazan raspored treninga za narednih sedam dana (slika 3).



Slika 3. Prikaz početne strane korisnika

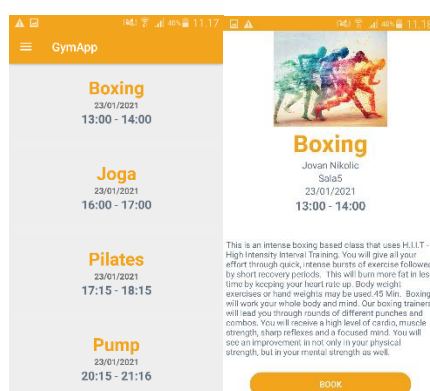
Korisniku se nudi mogućnost detaljnog prikaza treninga, klikom na karticu treninga u rasporedu treninga.. Detaljan prikaz sadrži informaciju o instruktoru, mestu održavanja treninga (broj sale), vreme održavanja i kratak opis treninga.

Ukoliko je korisnik zainteresovan za dati trening nudi mu se opcija zakazivanja treninga (*Book*).

### 4.2. Klijentski deo mobilne aplikacije

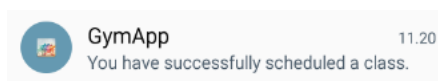
Klijentski deo mobilne aplikacije realizovan je kao Android aplikacija, za uređaje sa Android 4.1 ili novijom verzijom zbog potreba FCM klijenta. Android aplikacija namenjena je isključivo korisnicima teretane. Pokretanjem aplikacije korisniku se prikazuje forma za prijavu na sistem.

Unošenjem email adrese i lozinke korisniku se dozvoljava pristup njegovom nalogu i prelazak na početnu stranu (slika 4 - levo).



Slika 4 Prikaz početne strane (levo) i detaljan prikaz treninga (desno)

Selektovanjem konkretnog treninga, poziva se aktivnost za detaljan prikaz treninga (slika 4 - desno). Korisniku se nudi mogućnost zakazivanja treninga klikom na dugme *Book*. Nakon što korisnik zakaže trening prikazuje mu se obaveštenje u *Toolbar*-u, u vidu potvrde da je uspešno zakazao trening (slika 5). Slanje obaveštenja realizovano je uz pomoć FCM klijenta.



Slika 5. Prikaz obaveštenja u *Toolbar*-u

Da bi se kreirala klijentska aplikacija sa FCM za Android, potrebno je koristiti *FirebaseMessaging API* i *Android Studio 1.4* ili većom verzijom sa *Gradle*-om.

FCM pruža pouzdanu i efikasnu vezu između servera i uređaja koja omogućava besplatnu isporuku i primanje poruka i obaveštenja. Pre nego što se doda *Firebase* u projekat, potrebno je registrovati projekat na *Firebasu*-u.

### 4.3. Serverski deo

Serverski deo aplikacije kao što je već navedeno, implementiran je korišćenjem Express Framework-a. Serverska strana zadužena je za uspostavljanje komunikacije sa MongoDB preko Mongoose-a, koji predstavlja biblioteku za objektno modeliranje podataka između MnogoDB i Node.js. Nakon kreiranja Express aplikacije, uspostavlja se konekcija sa bazom podataka preko Mongoose-a, postavljaju se rute koje su vidljive klijentskim aplikacijama na podrazumevanom portu 3000. Rute su podeljene u tri grupe: autentifikacione rute, rute vezane za trening i rute vezane za teretanu. Takođe podešen je pristup rutama samo ukoliko se u zaglavlju zahteva koji stiže sa klijentske strane nalazi token, osim za autentifikacione rute kojima je dozvoljen pristup bez tokena.

Mongoose-ova šema definiše strukturu dokumenta, podrazumevane vrednosti i validatore. Sve u Mongoose-u počinje sa šemom, svaka šema se preslikava na MongoDB kolekciju i definiše oblik dokumenta u kolekciji.

Sve funkcionalnosti vezane za trening nalaze se u zasebnom kontroleru. Samo prijavljeni korisnici mogu da pristupe rutama kontrolera. Kontroler iz svakog HTTP zahteva preuzima parametre zahteva i prosleđuje ih servisu, u kome je izmeštena sva logika. Servis zatim vraća rezultat kontroleru, u vidu statusa i vrednosti samog rezultata nakon čega kontroler šalje tu povratnu vrednost klijentskoj aplikaciji. Sve funkcije u servisu su asihrone i kao povratnu vrednost vraćaju *Promise*. Pošto su funkcije asihrone, dozvoljeno je korišćenje *await* izraza. *Await* izrazi se stavljaju na pozive upućene ka bazi, kako bi se zaustavio napredak kroz funkciju.

Prilikom zakazivanja treninga, pored ažuriranja korisnikove liste zakazanih treninga u bazi podataka, šalje se prvo obaveštenje da je uspešno zakazan trening i vrši se zakazivanje obaveštenja koje treba da se pošalje sat vremena pre početka zakazanog treninga. Implementacija metode za slanje obaveštenja prikazana je na listingu 3. Kako bi se obaveštenje poslalo na određeni uređaj, potrebno je proslediti registracioni token.

```
firebaseAdmin.sendNotification = function (payload) {
  const message = {
    notification: {
      title: payload.title,
      body: payload.body,
    },
    token: payload.token,
  };
  return admin.messaging().send(message);
};
```

Listing 3. Implementacija metode za slanje obaveštenja

Obaveštenje koje treba da stigne sat vremena pre početka treninga zakazuje se pomoću *node-schedule*. Node Schedule omogućava da se zakazuju obaveštenja za određenje datume i vreme. Svako zakazano obaveštenje u rasporedu predstavljeno je objektom.

### 5. ZAKLJUČAK

Osnovni zadatak ovog rada bio je kreiranje veb i mobilne aplikacije uz korišćenje FCM za slanje obaveštenja nakon zakazivanja treninga kao i obaveštenja sat vremena pre početka zakazanog treninga. Rad obuhvata osnovne funkcionalnosti koje teretana nudi svojim korisnicima poput pregleda rasporeda treninga za narednih sedam dana, zakazivanje i otkazivanje treninga, kao i pregled liste zakazanih treninga. Dalja proširivost ovog zadatka, podrazumevala bi proširivost osnovnih funkcionalnosti koje se nude korisniku kako za veb tako i za mobilnu aplikaciju.

Prilikom implementacije zadatka korišćene su najnovije tehnologije koje se koriste za kreiranje veb i mobilnih aplikacija, sa ciljem lakog održavanje, i jednostavne i efikasne nadogradnje. FCM koji je korišćen u izradi ovog zadatka, predstavlja trenutno najefikasnije rešenje za razmenu poruka na više platformi bez ikakvih troškova. FCM je konstantno u procesu razvoja što pruža mogućnost proširenja u budućnosti. Takođe korišćenjem savremenih i trenutno aktuelnih tehnologija, stvaraju se mogućnosti za nadogradnju aplikacije u budućnosti kako bi se pružila još lakša i brža interakcija između korisnika i teretane.

### 6. LITERATURA

- [1] Angular, <https://angular.io/>
- [2] Express, <https://expressjs.com/>
- [3] Firebase Cloud Messaging, <https://firebase.google.com/docs/cloud-messaging>
- [4] Android, <https://developer.android.com/>
- [5] MongoDB, <https://www.mongodb.com/>

#### Kratka biografija:

**Ilinka Kovačević** rođena je 14.11.1996. godine u Novom Sadu. Završila gimnaziju "Svetozar Marković" u Novom Sadu. Godine 2015 upisuje Fakultet tehničkih nauka, smer Računarstvo i automatika. Na trećoj godini studija opredeljuje se za smer Primenjene računarske nauke i informatiku. Diplomirala je 2019. godine te je upisala master akademske studije na istom fakultetu, smer Računarstvo i automatika, modul Softversko inženjerstvo. 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*.