

PROJEKTNI RAD IZ PREDMETA

**MOBILNO RAČUNARSTVO**

Studenti:

Katarina Milić 2016/0242

Igor Mirković 2016/0274

Marina Nikolić 2016/0204

Beograd, 2020.

Sadržaj

[Uvod 3](#_Toc43544756)

[Korišćene tehnologije 3](#_Toc43544757)

[React Native 3](#_Toc43544758)

[Node.js 4](#_Toc43544759)

[MongoDB 4](#_Toc43544760)

[Docker 5](#_Toc43544761)

[Expo 7](#_Toc43544762)

[Aplikacija 7](#_Toc43544763)

[Projektovanje 7](#_Toc43544764)

[Implementacija 8](#_Toc43544765)

[Korisnički interfejs 8](#_Toc43544766)

[Struktura projekta 15](#_Toc43544767)

[Zaključak 17](#_Toc43544768)

[Literatura 18](#_Toc43544769)

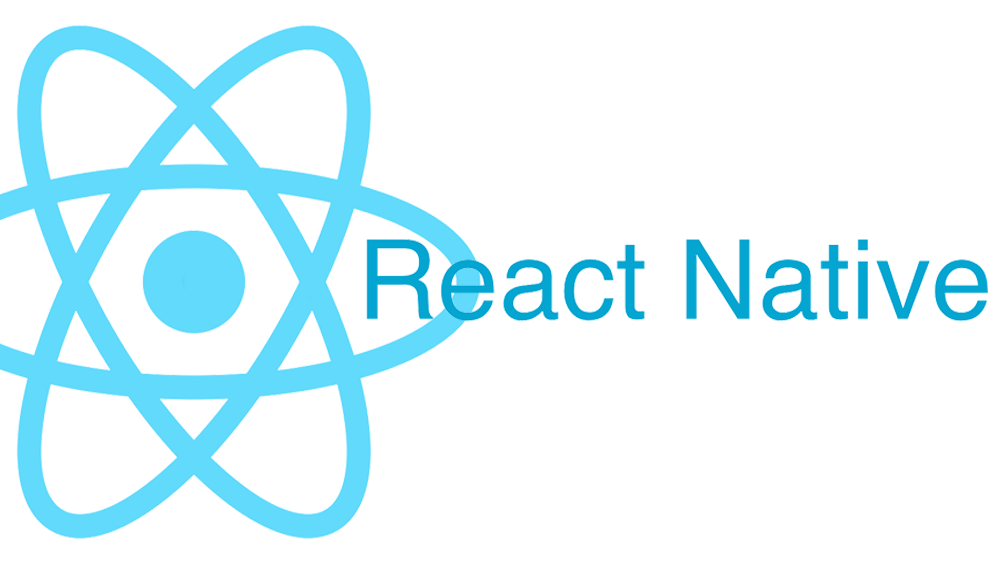
# Uvod

Napravili smo aplikaciju koja svojim korisnicima omogućava kreiranje korisničkog naloga, medjusobnu razmenu poruka, objavljivanje postova kao i pregled postova drugih korisnika. Za izradu aplikacije koristili smo tehnologije kao što su React Native, Node.js i Docker, dok smo za bazu podataka koristilil tehnologiju MongoDB. Takodje smo i obezbedili onlajn server, koji nam omogućava konzumiranje API-ja. Aplikacija je dostupna na svim postojećim operativnim sistemima.

# Korišćene tehnologije

## React Native

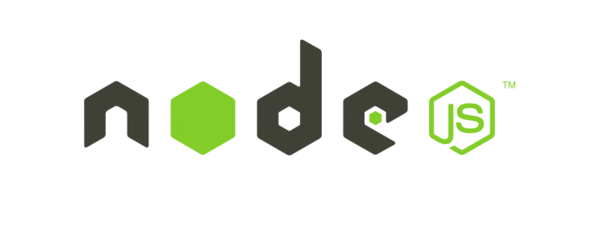
React Native je napravljen od strane Facebook-a. To je framework za razvoj mobilnih aplikacija, koristi se za razvoj aplikacija za android, iOS. Developerima je omogućeno da uz pomoć JavaScript-a razvijaju nativne aplikacije.



Slika 1 - React Native

## Node.js

Node.js je radno okruženje JavaScript programskog jezika koje se koristi na strani web servera. Node je omogućio da se JS koristi i za back-end razvoj odnosno serversko izvršavanje koda.



Slika 2 - Node.js

**Express**

Express ili Express.js je Node.js web framework. Koristi se za programiranje web aplikacija i [API](https://sr.wikipedia.org/wiki/%D0%90%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D1%98%D1%81) (Aplikacioni programski interfejs). Express olakšava i ubrzava mnoge radnje i sprečava pisanje suvišnih linija koda prilikom kreiranja Node.js aplikacija.

## MongoDB

**MongoDB**je [NoSQL](https://sr.wikipedia.org/w/index.php?title=NoSQL&action=edit&redlink=1) baza podataka.[[1]](https://sr.wikipedia.org/sr-el/MongoDB#cite_note-1) Prvi put je javno objavljen 2009. godine. Naziv Mongo potiče od reči humongous, koja označava nešto izuzetno veliko.  Napisan je u [C++](https://sr.wikipedia.org/wiki/C%2B%2B) jeziku.U skladu sa rastućim interesovanjem za alternativne sisteme za upravljanje bazama podataka, koji se razlikuju od tradicionalnih relacionih baza podataka, pojavio se koncept takozvanih NoSQL baza podataka, koje ne koriste SQL za povezivanje, nerelacione su, distribuirane, otvorenog koda i horizontalno skalabilne. Napisana je u [C++](https://sr.wikipedia.org/wiki/C%2B%2B) jeziku.

**MongoDB** čuva podatke kao [JSON](https://sr.wikipedia.org/wiki/JSON) dokumente sa dinamičkim šemama. Mongo je JSON dokument baza, mada su tehnički gledano podaci smešteni u binarnu formu JSON-a, poznatiju kao BSON.

****

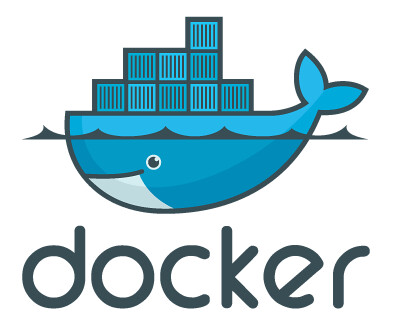
Slika 3 - MongoDB

## Docker

Docker je aplikacija otvorenog koda koja automatizuje razvijanje aplikacija u kontejneru. Sa Dockerom, programeri brinu samo o aplikacijama koje su pokrenute unutar kontejnera, ne i o upravljanju kontejnerima.

Kontejneri se pokreću iz slika koje su fleksibilne, mogu da se dele, menjaju i one predstavljaju izvorni kod kontejnera. Docker pakuje slike u registre koji mogu da budu privatni i javni. Javne slike se nalaze na Docker Hub-u koji trenutno sadrži više od 10000 slika kreiranih i podeljenih od strane drugih ljudi.

Jedna od najvećih prednosti Docker-a je to što je veoma brz, većina kontejnera može da se startuje za manje od sekunde i jednostavne aplikacije mogu da se dokerizuju vrlo brzo. Takođe, Docker koristi Copy on Write (COW) model koji omogućava brzu izmenu aplikacija pokrenutih unutar kontejnera.



Slika 4 - Docker

Docker se sastoji od:

* **Klijenta i servera** – Docker je klijent-server aplikacija. Docker klijent komunicira sa serverom ili demonom koji radi sav posao u pozadini. Docker-om može da se upravlja preko komandne linije ili RESTful API-ja.
* **Slika** – gradivni elementi Docker-a. Kontejneri se pokreću na osnovu slika. Slike se smatraju izvornim kodom kontejnera, portabilne su i mogu da se čuvaju, menjaju i dele.
* **Registara** – slike se čuvaju u registima. Postoje dva tipa registara: **privatni** i **javni**. Javni registar za slike odžava Docker i on se zove **Docker Hub**. Docker Hub podržava i privatne registre, svaki korisnik može privatno da sačuva svoju sliku i kasnije je preuzme uz upotrebu korisničkog imena i šifre.
* **Kontejnera** – se pokreću iz slika i mogu da sadže jedan ili više procesa u sebi. Docker kontejneri imaju isti koncept kao i pravi kontejneri, samo što umesto materijalnih stvari sadrže softver. Docker ne pravi razliku između kontejnera, nije bitno odakle je kontejner došao i šta se u njemu nalazi, svi kontejneri će biti pokrenuti na isti način.

## Expo

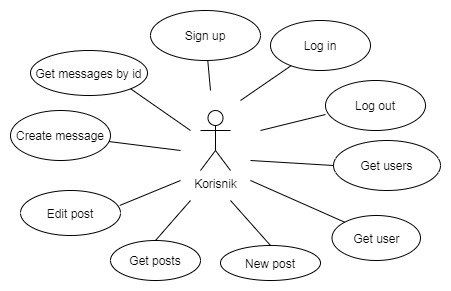
Expo je skup alata koji se koristi za lakši razvoj mobilnih aplikacija. Uz pomoć expo-a aplikacija se može pokrenuti i sa mobilnog telefona, uz njegovu instalaciju na isti.

# Aplikacija

U narednom tekstu smo objašnjavali projektovanje i implementaciju same aplikacije.

## Projektovanje

U našoj aplikaciji postoje sledeći slučajevi korišćenja:



Slika 5 - Slučajevi korišćenja

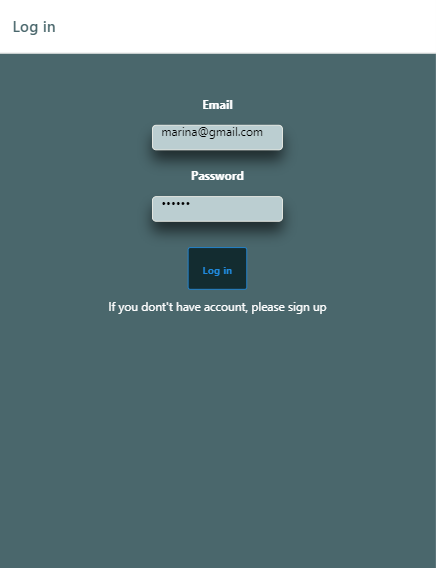
Projektovali smo tri ključna entita: User, Post, Message.

1. Sign up – Kada korisnik uđe u aplikaciju prikazuje mu se LogInScreen, na kome postoji mogućnost prijavljivanja korisnika. Ukoliko korisnik nema kreiran korisnički nalog, postoji mogućnost redirekcije na SignUpScreen, na kome mu se otvara forma za unos podataka i klikom na dugme kreira se korisnički nalog. Nakon uspešne prijave, aplikacija ga redirektuje na LogInScreen. U slučaju da prijava nije validna ispisuje se greška.
2. Log in – Korisnik ulazi na aplikaciju, popunjava svoje korisničko ime i lozinku. Ukoliko se korisnik uspešno prijavi na aplikaciju, server mu vraća token i aplikacija ga redirektuje na NewsFeedScreen.
3. Get posts -Korisnik je prоšao gateway. I inicijalno dobija postove svih korisnika na aplikaciji.

## Implementacija

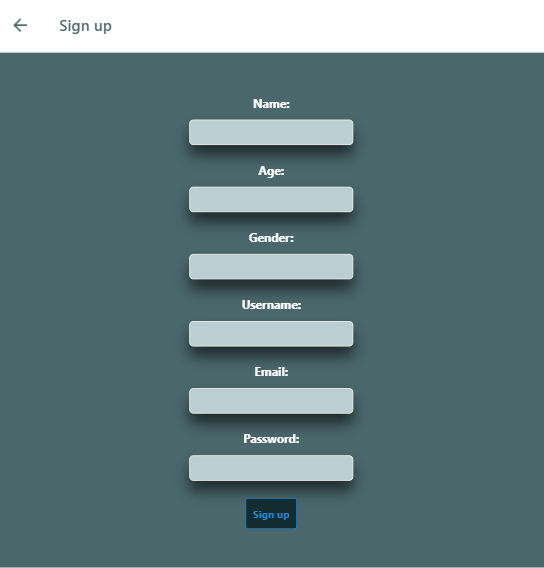
## Korisnički interfejs

1. Log in stranica



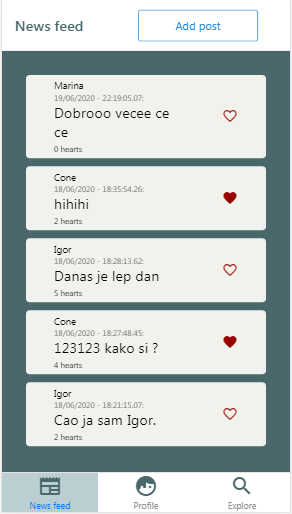
Slika 6 - LogInScreen

1. Sign up stranica



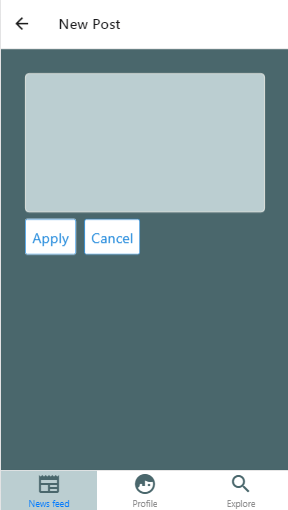
Slika 7 – SignUpScreen

1. News Feed stranica



Slika 8 – NewsFeedStranica

1. Stranica za dodavanje post-ova



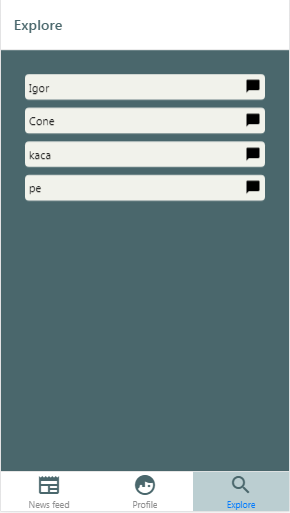
Slika 9 – NewPostScreen

1. Profile stranica



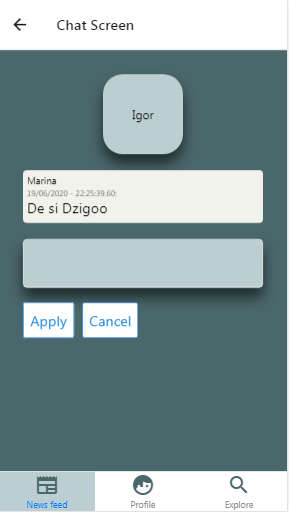
Slika 10 – ProfileScreen

1. ExploreScreen



Slika 11 – ExploreScreen

1. ChatScreen

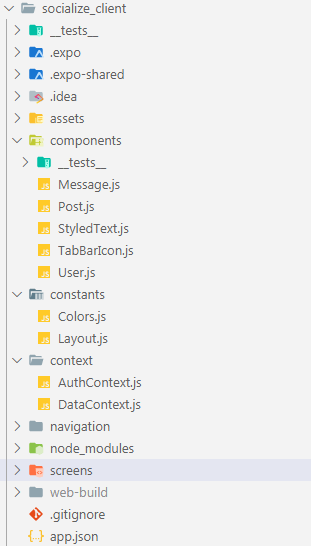


Slika 12 – ChatScreen

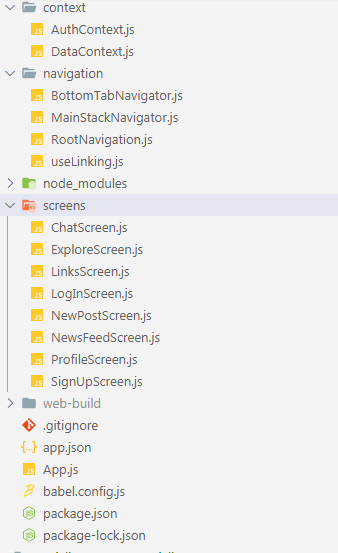
## Struktura projekta

Klijentska strana je razvijena uz pomoć React Native-a, dok je serverska uz pomoć Node.js.

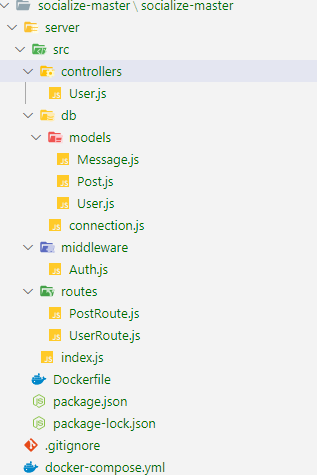
MongoDB smo koristili kao NoSQL bazu podataka.



Slika 13 - Struktura projekta klijentske strane I deo



Slika 14 - Struktura projekta klijentske strane II deo



Slika 15 - Struktura projekta serverske strane

# Zaključak

U radu je prikazan proces razvoja mobilne aplikacije koja predstavlja jednostavnu društvenu mrežu. Obuhvaćen je razvoj aplikacije od procesa specifikacije zahteva, implementacije, projektovanja, kao i samog dizajna aplikacije, sa raznim mogućnostima daljeg usavršavanja. U okviru rada takodje su i prikazane teoretske osnove koje se odnose na korišćene tehnologije prilikom razvoja aplikacije.

Tokom izrade same aplikacije, jedan od većih izazova bio je da se omogući jednostavno korišćenje same aplikacije. Ovo je urađeno kroz jednostavan, konzistentan i intuitivan dizajn uz pomoć React Native-a, dok su ključne funkcionalnosti realizovane u Node.js-u. Aplikacija je realizovana kao prototip i predstavlja dobru bazu za dalje usavršavanje i dodavanje novih funkcionalnosti.

# Literatura

1. <https://reactnative.dev/>
2. <https://www.docker.com/>
3. <https://nodejs.org/en/>
4. <https://www.mongodb.com/>