Razvoj klijentskih komponenti web aplikacije za stvaranje novih prijateljskih i ljubavnih veza

Sučić, Laura

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:000594

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurje Dobrile u Puli

Fakultet informatike u Puli

LAURA SUČIĆ

RAZVOJ KLIJENTSKIH KOMPONENTI WEB APLIKACIJE ZA STVARANJE NOVIH PRIJATELJSKIH I LJUBAVNIH VEZA

Diplomski rad

Pula, rujan, 2021. godine

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

LAURA SUČIĆ

RAZVOJ KLIJENTSKIH KOMPONENTI WEB APLIKACIJE ZA STVARANJE NOVIH PRIJATELJSKIH I LJUBAVNIH VEZA

Diplomski rad

JMBAG: 0246070890, redoviti student

Studijski smjer: Sveučilišni diplomski studiji Informatike

Predmet: Napredni algoritmi i strukture podataka

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Tihomir Orehovački

Pula, rujan, 2021. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Laura Sučić kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

Laura Sučić

U Puli, 19. rujna 2021. godine

IZJAVA

o korištenju autorskog djela

Ja, Laura Sučić dajem odobrenje Sveučilištu Jurja Dobrile

u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Razvoj klijentskih komponenti web aplikacije za stvaranje novih prijateljskih i ljubavnih veza“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 19. rujna 2021.

Potpis

Laura Sučić

Sadržaj

1. Uvod 1
2. Pregled trenutnih rješenja na temu 2
3. Istraživanje tržišta i motivacija za izradu aplikacije 4
3.1. Tinder 4
Funkcionalosti Tindera 4
3.2. Badoo 5
Funkcionalnosti Badoo-a 6
3.3. Facebook 7
Funkcionalnosti Facebook-a 8
3.4. Instagram 9
Funkcionalnosti Instagrama 9
3.5. Motivacija za izradu aplikacije 11
4. Korištene tehnologije i dijagram slučajeva aplikacije 13
4.1. React 13
JSX 13
Metode životnog ciklusa React-a 14
4.2. React Hooks 15
4.3. Bootstrap 15
4.4. Mapbox 17
4.5. Axios 18
4.6. React-router-dom 19
4.7. Dijagram slučajeva korištenja aplikacije 20
5. Arihtektura aplikacije, njeno korisničko sučelje i korisnički zahtjevi aplikacije 22
5.1. Uvodna stranica 22
5.2. Glavna stranica 22

5.3. Profilna stranica 23
5.4. Korisnikova stranica 24
5.5. Stranica za razgovor 25
5.6. Korisnički zahtjevi 25
6. Razrada funkcionalnosti klijentskih komponenti 27
6.1. Prijava 27
6.2. Zaboravljena lozinka 30
6.3. Registracija 31
6.4. Dovršavanje registracije nakon prijave 34
6.5. Access token 35
6.6. Mapa sa korisnicima u blizini 37
6.7. Dohvat trenutne lokacije 39
6.8. Filtriranje korisnika na mapi 40
6.9. Pronalaženje korisnika sa sličnim atributima 42
6.10. Sustav života 43
6.11. Označavanje korisnika sa sviđa mi se 46
6.12. Označavanje korisnika sa ne sviđa mi se 48
6.13. Notifikacijski sustav 49
6.14. Korisnički profili 52
6.14.1. Privatni profil 52
6.14.2. Javni profil 53
6.15. Promjena osobnih podataka 53
6.16. Chat 54
6.16.1. Blokiranje razgovora sa korisnikom 55
7. Zaključak 56
8. Literatura 57
9. Popis slika 59

Sažetak 61

Abstract 61

1. Uvod

Sve više i više ljudi traži ljubav i poznanstva preko interneta. Traženje ljubavi ili poznanstva online daje korisnicima puno veći izbor nego što bi to bilo samo u njihovom okruženju. Preko aplikacija za traženje ljubavnih veza ljudi se mogu pronaći u nekoj drugoj kulturi koja nije uvijek prihvaćena u njihovom okruženju. Te aplikacije olakšavaju proces prvotnog komuniciranja kako bi se pronašla „kemija“ i kompatibilnosti između korisnika. Možemo komunicirati s osobom ugodno u svojoj domaćoj atmosferi bez javnih društvenih pritisaka. Danas postoje mnoge aplikacije i socijalne mreže koje rješavaju problem ljubavi i prijateljstva. Neke od njih donijele su milijarderske zarade što objašnjava da je ljudima potreban alat za postizanje novih odnosa. Danas troje od deset odraslih osoba u Sjedinjenim Američkim Državama tvrdi da su koristili online stranicu za stvaranje ljubavnih veza, te njih 11% da su ju koristili unazad godinu dana (prema New Pew Research Center pregledu provedenom od 16. listopada do 28. listopada 2019. godine (Center, 2020)). Za neke od korisnika takvih aplikacija te platforme su bile zaslužne za kreiranje kvalitetnih i značajnih veza. Čak 12% ispitanika tvrdi da su se oženili ili su bili u dugotrajnoj vezi s nekim koga su prvotno upoznali preko stranica ili aplikacija za stvaranje ljubavnih odnosa (Center, 2020). Istraživanje također pokazuje da su navedene stranice veoma popularne u određenim grupama kao što su homoseksualne ili biseksualne osobe, te među adolescentima. (Center, 2020). Jasno je i da dolazi do veoma drugačijeg rodnog iskustva na dating stranicama. Muškarci tvrde kako ne dobivaju dovoljno poruka s obzirom na druge korisnike ženskog spola, dok mlade žene tvrde da dobivaju veoma seksualno eksplicitne poruke koje nisu tražile te da doživljavaju seksualno uznemiravanje od muških korisnika (Center, 2020). Zbog navedenih tvrdnji jasno je da broj korisnika koji koriste aplikacije za pronalaženje ljubavnih i prijateljskih odnosa raste iz godine u godinu, te da im je potrebna platforma gdje se osjećaju sigurno, vrijedno i zadovoljno. Cilj ovog rada je s kolegom Karlom Dini Kožićem (Kožić, 2021.) razviti novu aplikaciju na području Hrvatske koja analizira postojeće rezultate i dodaje nove funkcionalnosti kako bi se obogatili ljudski odnosi i povećala kvaliteta usluge aplikacije za sklapanje ljubavnih i prijateljskih odnosa.

Aplikacija je podijeljena na dva dijela. Ovaj rad se bavi razvojem klijentskih komponenti aplikacije dok se kolega Kožić u svom radu bavi razvojem poslužiteljskih komponenti aplikacije.

2. Pregled trenutnih rješenja na temu

U radu Ive Paska „Fast Choices and Emancipatory Spaces: Complex Reality of Online dating Apps“ govori se o kompleksnoj realnosti aplikacija za stvaranje ljubavnih i prijateljskih odnosa. Jedan od aspekata početka trećeg desetljeća 21. stoljeća je upoznavanje potencijalnih partera kroz mobilne aplikacije što je postalo učestalo u postizanju romantičnih veza. Najpopularnija aplikacija za postizanje ljubavnih veza je Tinder, te nakon nje slijede OKCupid, Tantan, Badoo i ostale (Paska, 2020). Tinder se smatra začetnikom modernih dating aplikacija. Sa svojim ogromnim brojem korisnika on postaje glavna platforma za upoznavanje novih potencijalnih partnera i prijateljstva: „The application Tinder is considered an innovator and leader within the industry of online dating, being present in more than 190 countries around the world, with more than 50 million users.“ (Paska, 2020). Naglasak se bazira na vizualnom aspektu gdje je korisniku dozvoljeno maksimalno šest fotografija i limitirani opis teksta s kojim će privući potencijalne partnere. Takozvano pomicanje prstom udesno ili ulijevo (engl. swipe) što predstavlja odbijanje ili prihvaćanje korisnika, postaje presudno za stvaranje novog ljubavnog interesa. Naglašava se kako biranje u moru potencijalnih korisnika postaje kao biranje namjernica u dućanima što se naziva „emocionalni kapitalizam“ gdje su korisnici preplavljeni izborom. Navedeni sustav omogućuje bezbolno odbijanje ljudi i smanjenje frustracije prilikom odbijanja, no to zapravo dovodi do još veće frustracije, jer se niti jedna veza ne doživljava ozbiljno, nego olakotno i bezbolno, što bitan bliski odnos ne bi trebao biti. Stoga osoba kreira površne odnose koji su lako zamjenjivi naspram dubokih odnosa u koje se može pouzdati.

Preko pola populacije druge seksualne orijentacije u Sjedinjenim Američkim Državama svoje je partnere upoznalo online što dokazuje da su im online dating aplikacije omogućile sigurno okruženje za upoznavanje novih partnera bez socijalnih pritisaka i diskriminacija od društva uživo. Istraživanja su pokazala da je čak 60% LGBT zajednice uspješno upoznalo svoje partnere online. 41% homoseksualnih parova se upoznalo online, od kojh je 17% heteroseksualnih (Paska, 2020).

U Ortega i Hergovich radu opisuje se kako su online dating aplikacije korisnicima olakšale upoznavanje novih ljudi s kojima nikako nebi mogli stupiti u kontakt preko svojih poznanika. Tehnološki napredak omogućuje lakše pronalaženje preko lokacije na mobitelu te olakšava organiziranje grupa i njihovog mjesta upoznavanja i druženja.

Također, lakše nam je naći i filtrirati korisnike sa sličnim interesima: „Thus it is not surprising that this way of connecting will be of greatest importance for those social groups whose members are hard to locate in the physical social reality, for example marginalized social groups that lack venues for meeting in a certain social context.“ (Ortega i Hergovich, 2017)

Glavni problemi koji su navedeni u istraživanjima kao što su: površni odnosi, more izbora i dr. nastoje se riješiti razvojem web aplikacije opisane u ovom radu.

3. Istraživanje tržišta i motivacija za izradu aplikacije

Na tržištu aplikacija za stvaranje ljubavnih i prijateljskih odnosa postoji mnogo rješenja. Neka su popularnija od drugih i većina njih traži naplatu za poboljšanje usluga aplikacije.

Aplikacije Tinder, Badoo, Facebook i Instagram bit će detaljnije objašnjene i opisane u narednom poglavlju.

3.1. Tinder

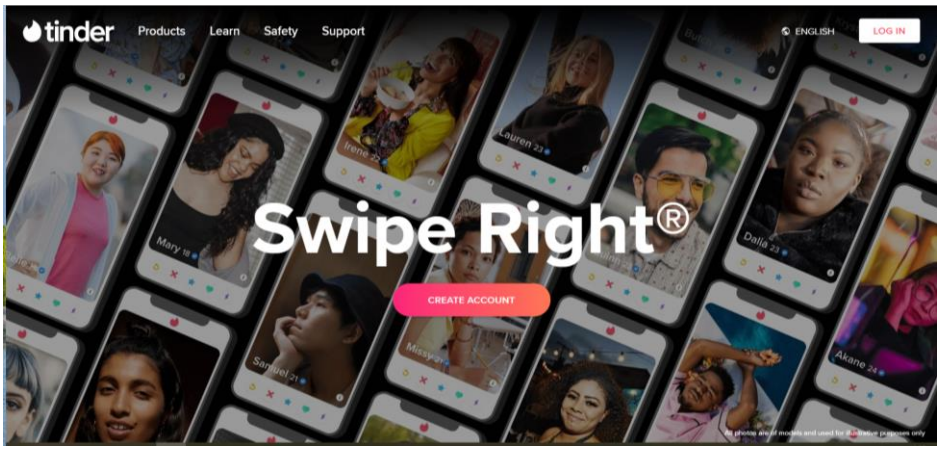
Tinder je aplikacija za traženje prijateljskih i ljubavnih odnosa. Bazira se na funkcionalnosti povlačenja prstom ulijevo ili udesno (engl. swipe) gdje korisnik pomakom prsta udesno odobrava osobu, a pomakom ulijevo odbija osobu. Tinder se najčešće koristi za postizanje kratkih seksualnih veza s korisnicima gdje se ljude na neki način prikazuje kao na traci i iskorištava za kratkotrajne usluge. (Short, 2021).

Funkcionalosti Tindera

Tinder omogućuje prijavu preko Facebook-a/Google e-pošte ili preko vlastitog broja mobitela. Korisniku je dodijeljen vlastiti profil preko kojeg može dodati slike, datum rođenja, mjesto rada i dr. Nakon ispunjenja navedenih informacija, korisniku je dana opcija nadogradnje profila uz nadoplate ili dodavanje tagova koji predstavljaju kratke opise. Korisniku su vidljive informacije kao što su: godine, udaljenost, zajednički prijatelji, zajednički interesi i tagovi drugih korisnika. Dostupnost na Tinderu se obavlja preko geolokacije, te je ona najbitnija za dohvaćanje potencijalnih kontakata. Funkcionalnost pretrage se bazira na filterima kao što su godine, spol i udaljenost. Međutim, mnogi od njih su dostupni nakon naplate. Glavna funkcionalnost Tindera je pomicanje prsta ulijevo ill udesno (engl. swipe) kako bi se odobrili ili odbili korisnici. Kako bi korisnici započeli razgovor moraju se uspješno spojiti (engl. match), a to se postiže kada korisnici oboje povuku prst udesno za određenog korisnika. Postoji mogućnost „Super Like“ gdje korisnik mora povući prstom prema gore što se s besplatnom verzijom može koristiti jednom na dan. Plaćena verzija omogućuje do pet super odobrenja. Kada se tako odobri korisnika, svi podaci postanu vidljivi i korisnik

vidi tko ga je odobrio, te se označava plavom vrpcom. Mnogi podaci poput udaljenosti, godina i dr. mogu se sakriti, kao i mnoge druge kontrole koje su moguće preko naplate aplikacije (Makhija, 2021).

Iako ima rijetkih primjera gdje su se ljudi povezali preko Tindera, Tinder svojim funkcionalnostima, od kojih se većina naplaćuje, ne uspijeva sretno povezati većinu osoba preko aplikacije.



Slika 1. Prikaz početne stranice Tinder aplikacije [9]

3.2. Badoo

Badoo je također aplikacija za pronalaženje prijateljskih i ljubavnih odnosa koja je većinu funkcionalnosti preuzela od Tindera.

Badoo je besplatna aplikacija za stvaranje ljubavnih i prijateljskih odnosa dostupna na Windows, macOS, iOS i Android sustavima. Aplikacija je nastala 2006. godine i tvrdi da ima preko 250 milijuna korisnika u 190 zemalja (Ally, 2021).

Badoo većinu svojih korisnika vuče iz Europe i Južne Amerike, dok se Tinder uglavnom koristi u Sjedinjenim Američkim Državama. (Short, 2021)

Zbog toga, ako je korisnik iz ovih područja, lakše će naći korisnike na toj aplikaciji.

Funkcionalnosti Badoo-a

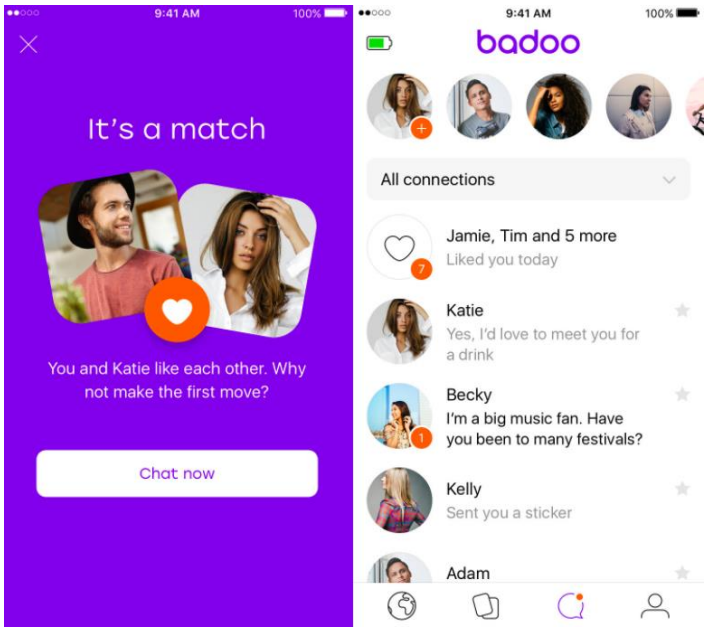
Postoje dva načina za upoznati ljude na Badoo aplikaciji, a to su slučajni susreti i ljudi u blizini. U susretima se ljudi spajaju preko samaca na aplikaciji gdje aplikacija nastoji povezati korisnike po pripadajućim sličnim interesima. Ne postoji ograničenje s koliko se osoba korisnik može spojiti na dan. Sekcija „Ljudi u blizini“ prikazuje se kao mreža ljudi koji su trenutno u području korisnika i dijele slične interese. Korisnik izabire iz mreže određene korisnike i klikom na ikonu „srce“ označava sa sviđa mi se, a klikom na „x“, sa ne sviđa mi se. Osnovni Badoo račun je besplatan. Plaćanjem mjesečne naknade otključavaju se dodatne funkcionalnosti s takozvanim imenom „Super Powers“. Neki od pristupa povlastica s premium pretplatom su:

- Mogućnost pregleda tko je u susretima označio profil sa „sviđa mi se“
- Mogućnost pregleda tko je postavio korisnika u listu favorita
- Anonimno pretraživanje profila
- Brisanje zadnjeg glasanja
- Razgovor s najpopularnijim Badoo korisnicima
- Korisnikove poruke dobivaju veći prioritet u primateljevoj pošti
- Bez reklama

Badoo predstavlja pravi „melting pot“ među korisnicima. Prijava je pojednostavljena slično kao i preko Tinder-a. Moguće ju je obaviti preko Facebook-a ili preko e-pošte. Potrebno je objaviti barem jednu sliku koja prati geste dane slike kako bi se moglo komunicirati s korisnicima. Ta slika se koristi samo za autentificiranje korisnika. Ako je prijava napravljena preko Facebook-a, Badoo preko njega automatski povlači sve informacije kao što su: visina, posao, godine, status veze, izgled i dr. Uz navedeno, dana je mogućnost kratkog opisa od 500 znakova gdje se nastoji impresionirati korisnike atraktivnim i zabavnim opisom o sebi (Ally, 2021).

Usprkos velike sličnosti Tinderu, Badoo ima još nekoliko drugih funkcionalnosti kao što su: mogućnost privatnih postavki, postavljanje nevidljivosti, prikaz posjeta profilu korisnika i ocjenjivanje dnevnih „matcheva“. Također ima i web aplikaciju za razliku od Tindera koji ima samo mobilnu aplikaciju.

Iako ima više funkcionalnosti od Tindera, Badoo i dalje povlači lošu reputaciju kao i Tinder. (Short, 2021)



Slika 2. Prikaz sučelja Badoo aplikacije [1]

3.3. Facebook

Facebook je socijalna mreža koja olakšava povezanost između prijatelja i obitelji online. Kreirao ga je Mark Zuckerberg 2004. godine s početnom namjenom za studente, no do 2006. godine svatko stariji od 13 godina mogao se pridružiti ovoj mreži. Danas Facebook broji preko više od bilijun korisnika iz cijelog svijeta (GCFGlobal , 2021).

Preko pola svijeta posjeduje Facebook profil. Glavni problem Facebooka je što funkcionira poput imenika. Kada na mrežu dodamo ljude koji su nam više poznanici nego prijatelji, automatski se pohranjuju pod listu „prijatelji“, što stvara iluziju velikog broja prijateljstva, dok ljudi najčešće imaju oko četiri do pet bliskih prijatelja (Good Therapy, 2019).

Takva iluzija ostavlja korisnika nesretnim, jer se osjeća kao da nema dovoljno prijatelja.

Funkcionalnosti Facebook-a

Facebook nudi mnogo funkcionalnosti koje se razvijaju iz dana u dan. Neke od glavnih općepoznatih funkcionalnosti su:

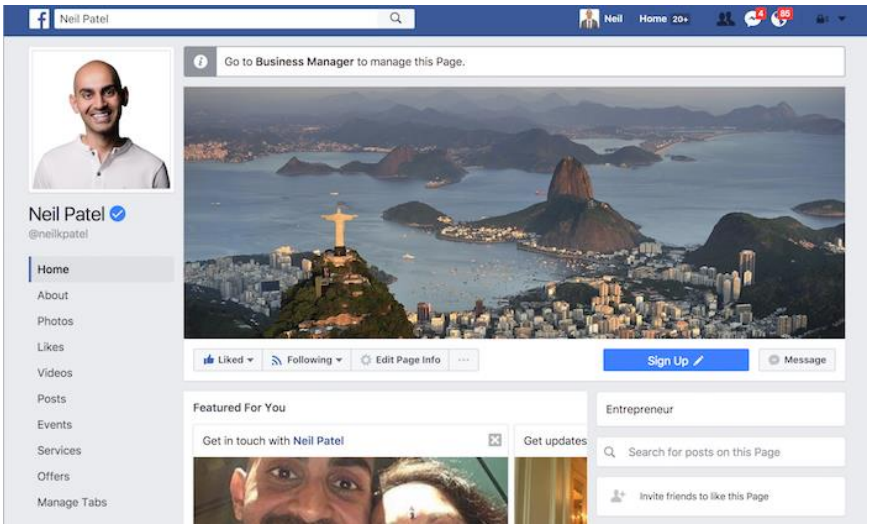
- Zid obavijesti (engl. news feed)
- Prijatelji
- Korisnikov zid
- Vremenska crta
- Sviđa mi se opcija i reakcije
- Komentari
- Poruke i pošta (engl. inbox)
- Notifikacije / obavijesti

Korisnici komuniciraju preko razgovora (engl. chat) koji im se nalazi na desnom dnu stranice. Korisnici mogu kreirati grupe unutar razgovora tako da istovremeno komuniciraju s više korisnika. Također mogu kreirati stranice istomišljenika gdje se razgovara o sličnim temama i objavljuju slike i statusi na određenu temu. Takva funkcionalnost je idealna za kreiranje i poboljšavanje zajednice gdje korisnici obitavaju. Neke od novih mogućnosti su „streaming“ ili uživo online komuniciranje s korisnicima ili snimanje vlastitog sadržaja i dijeljenje s ostatkom svijeta. Dodana je i mogućnost filtriranja sadržaja koje dobivamo tako da mičemo reklame koje nam se ne sviđaju i sl., te arhiva besplatne glazbe koju korisnici mogu preuzeti bez brige o autorskim pravima i dr (Peard, 2021).

Mana Facebooka je da su korisniku predloženi poznanici ili prijatelji od prijatelja i s time se ne mogu lagano pomaknuti iz svoje okoline poznanstva pri tome da je teško dobiti prihvaćen zahtjev za prijateljstvo od nepoznate osobe jer ljudi nisu skloni na Facebooku prihvaćati osobe koje ne poznaju.

U moru novih funkcionalnosti i već postojećih, korisnik se može izgubiti u beskrajnoj zabavi i sadržaju, pa se ova aplikacija ne smatra kvalitetnim mjestom za stvaranje dubokih veza. Fokus nije bačen na upoznavanje i komunikaciju, nego je namijenjen

za veće zajednice. Stoga Facebook ne predstavlja kvalitetnu aplikaciju za stvaranje ljubavnih veza među korisnicima (Miller, Tales from Facebook, 2011).



Slika 3. Prikaz profila korisnika na Facebook stranici [5]

3.4. Instagram

Instagram je pojednostavljenje Facebooka s manje funkcionalnosti i fokusom na slike korisnika. Problem Instagrama je što se on koristi u marketinške svrhe za oglašavanje proizvoda pomoću poznatih influencera, a ne za postizanje kvalitetnih odnosa. Instagram je socijalna mreža koji omogućuje korisnicima izmjenjivanje slika sa svojim prijateljima (pratiteljima). Može se preuzeti besplatno sa trgovine aplikacija. Broji preko 200 milijuna korisnika, te je kupljena od Facebook-a 2021.godine. Glavna funkcionalnost koja je donijela popularnost aplikaciji je uređivanje slika (webwise, 2021).

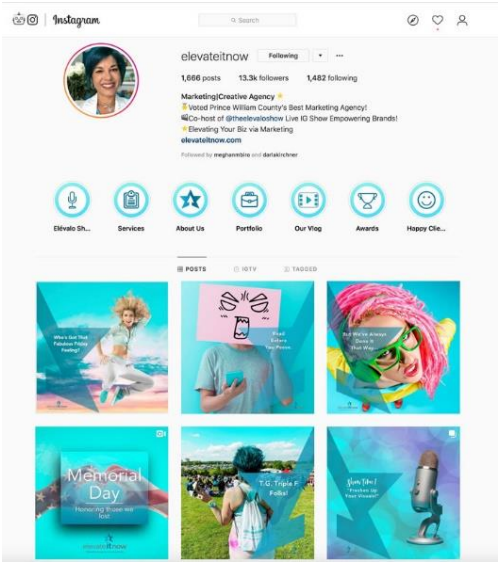
Funkcionalnosti Instagrama

Glavne funkcinonalnosti Instagrama mogu se opisati u 3 cijeline:

- Instagram pitanja
- Instagram direktno
- Instagram priče

„Instagram pitanja“ funkcionalnost je koja je dodana 2018. godine. Ona omogućuje korisnicima da odgovore na pitanja postavljena od njihovih pratitelja. Odgovori se mogu objaviti u priči, te tako svi pratitelji mogu pratiti sadržaj kreatora. „Instagram direktno“ (engl. Instagram Direct) omogućuje dijeljenje fotografija, tagova, profila i lokacija sa jednom ili više osoba (do 15) izravno preko korisnikovog zida. „Instagram priče“ omogućuje korisnicima da objave selekciju odabranih slika i video zapisa u jednu koherentnu priču. Priče su dostupne 24 sata i onda se spremaju u arhivu korisnika. Vidljive su samo prijateljima od korisnika osim ako oni ne postave privatnost drukčije. Kada korisnik kreira račun, njegov je profil automatski javan što znači da je cijeli sadržaj vidljiv svakome. Treba naglasiti da je to onda njegova odgovornost, te treba biti pažljiv sa zlostavljanjem na internetu (engl. cyberbullying) gdje se informacije mogu zloupotrebljavati. Zbog navedenog, Instagram povlači mnoge kontroverze na temu droga i dr. (BBC News, 2013). Kao i mnoge druge aplikacije, minimalna dob za posjedovanje profila je 13 godina, a kod nekih država i više (primjer Irske sa minimalnom dobi od 16 godina da bi se posjedovao račun na Instagramu (Instagram, 2021)). Još neke od funkcionalnosti su: blokiranje korisnika, prijava nepoželjnog sadržaja, komentari i „sviđa mi se“ oznake, razne mogućnosti privatnosti (webwise, 2021).

Instagram je odlična platforma za mlade korisnike koji žele objavljivati svoje slike i video zapise sa svojim prijateljima i pratiteljima, ali aplikacija povlači kompleks popularnosti među korisnicima. Ako osoba nema zanimljiv profil neće privući puno pratitelja, gdje vidimo da je aplikacija privlačna mnogim marketinškim tvrtkama namijenim za promoviranje i zaradu među korisnicima. Stoga Instagram ne predstavlja dobru platformu za postizanje kvalitetnih prijateljskih i ljubavnih odnosa (Dietz, 2019).



Slika 4. Prikaz Instagam profila korisnika [4]

3.5. Motivacija za izradu aplikacije

Kroz analizu postojećih rješenja uočeno je da ne postoji kvalitetna aplikacija za upoznavanje novih ljubavnih i prijateljskih odnosa, te se kroz ovaj rad pokušavaju rješiti mane poput:

• automatski odbijanje ljudi, rangiranje ljudi po izgledu prije upoznavanja (psihološke studije su pokazale da što se više družimo s određenim ljudima oni su nam ljepši i privlačniji (Kelly, 2010)) (Tinder, Badoo)

• istodobna komunikacija s prevelikom brojem ljudi smanjuje važnost komunikacije (ljudi izgledaju površno i nezanimljivo i mijenjaju se kao na „traci“) (Tinder, Badoo)

• micanje i pokušavanje izbjegavanja Facebook-ove mane imenika (ljudi su samo evidencija, a ne postoji interakcija među njima)

• pokušavanje smanjivanja popularnosti tipa (influenceri, celebrity, reklame, sponzori, kompanije) kao najvažnijih aktera u platformi (Instagram,TikTok itd.). Zbog influencera aplikacije postaju profitabilne, ali gube na važnosti kao glavne platforme za interakciju ljudi.

Zbog navedenih primjera i razloga su messenger aplikacije i dalje najvažnije platforme za interakciju sa prijateljima.(Discord-ovo super rješenje vlastitih servera (soba) kao mjesto zajedničkih interesa velikih ili malih grupa ljudi).

4. Korištene tehnologije i dijagram slučajeva aplikacije

4.1. React

React (također poznat kao i React.js ili ReactJS) je besplatna i otvorenog koda JavaScript biblioteka za izradu klijentskih web aplikacija. Koristi se za izradu korisničkog sučelja ili takozvanih korisničkih komponenti. Održava ga Facebook i zajednica individualnih programera i kompanija.

React kod se sastoji od entiteta zvanih komponente. Komponente se dodjeljuju određenom elementu DOM-a koji koristi React DOM (engl. Document Object Model) biblioteku. Pomoću virtualnog React DOM-a napravi kopiju DOM-a i potom zamijeni vrijednost u virtualnom DOM-u i pošalje promjene na DOM.

JSX

Sintaksa kojom se piše u React-u je JSX sintaksa. JSX ili JavaScript XML je ekstenzija JavaScript jezikovne sintakse. Svojom jakom naličju na HTML, JSX omogućuje strukturiranje renderiranja komponenti koristeći sintaksu poznatu mnogim developerima. Nakon kompilacije, JSX postaje regularan JavaScript i sve se čita kao JavaScript kod. Zbog toga JSX možemo koristiti unutar if i for petlji, dodijeliti ga varijabli, vratiti ga iz funkcije i dr (React, 2021).

```
function getGreeting(user) {
  if (user) {
    return <h1>Hello, {formatName(user)}!</h1>;
  }
  return <h1>Hello, Stranger.</h1>;
}
```

Slika 5. Prikaz JSX sintakse u React-u [7]

Zbog toga se tijekom ponovnog renderiranja izmjenjuju samo komponente koje su promijenile stanje za razliku od cijele stranice.

React je kreirao inženjer softvera Jordan Walke u Facebook-u, te je obavio prvi rani prototip React-a pod nazivom „FaxJS“. Bio je inspiriran bibliotekom PHP-a za HTML komponente pod nazivom XHP. React je prvotno objavljen na Facebook News Feed 2011. i kasnije na Instagramu 2012. godine. Postao je otvoreni kod 2013. godine, a 2017. godine pušten je javno.

Metode životnog ciklusa React-a

Metode životnog ciklusa (engl. lifecycle methods) pozivamo kada želimo nešto primijeniti nakon određenog mijenjanja komponente u React-u. One dozvoljavaju izvršavanje koda na određenim točkama trajanja komponente.

Najpoznatije funkcije su:

- shouldComponentUpdate
- componentDidMount
- componentWillUnmount

shouldComponentUpdate dozvoljava razvojnom inženjeru da zaustavi nepotrebno ponovno renderiranje komponente tako da vraća false ako render nije zadovoljen.

componentDidMount se pozove jednom čim se komponenta kreira i prvotno pozove pri kreiranju. Ova se funkcija najčešće koristi za okidač pri učitavanju podataka kroz neki API ili preko HTTP zahtjeva.

componentWillUnmount se odmah zove kada se komponenta izbriše ili nestane iz DOM-a. Npr. Za korištenje setInterval() funkcije ili tijekom poziva nekog eventa pomoću eventListener funkcije za sigurno prekidanje navedenih funkcija (React, 2021) (Wieruch, 2021).

4.2. React Hooks

React Hooks koristi takozvane „Hooks“ koje su zapravo komponente, ali u obliku funkcija i tako formiramo funkcijske komponente. One omogućuju korištenje React-a bez klasa.

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Slika 6. Primjer klasne komponente [6]

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Slika 7. Primjer funkcijske komponente [6]

React se može koristiti kao baza za izradu „single-page“ ili mobilnih aplikacija. Bazira se na upravljaju stanja i renderiranja stanja DOM-a koristeći se virtualnim „ReactDom-om“.

Dodatne klijentske funkcionalnosti i biblioteke za rutiranje su potrebne kako bi aplikacija uspješno funkcionirala.

4.3. Bootstrap

Bootstrap je besplatan, otvorenog koda, CSS okvir koji se koristi za izradu responzivnih, klijentskih web usluga. Sadrži sve potrebne elemente bazirane na JavaScript dizajn predlošcima za tipografiju, forme, gumbe, navigaciju i ostalo.

Bootstrap jedan je od 10 najpopularnijih okvira za izradu projekata na GitHub-u (GitHub, 2018).

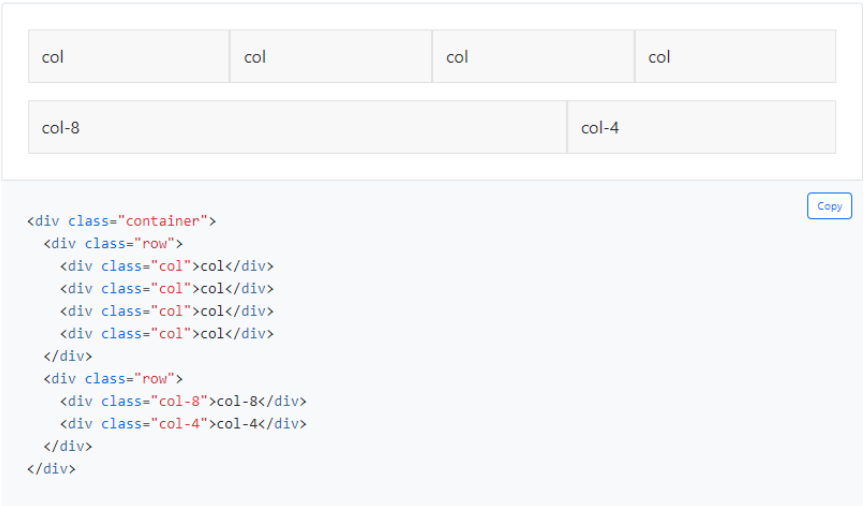
Bootstrap su razvili Marko Otto i Jacob Thornton u Twitteru kao standardni okvir za izradu internih alata na webu kako bi smanjio velika održavanja i objedinio biblioteke. Bootstrap je bio poznat kao Twitter Blueprint. Objavljen 19. 8. 2011., do sada je imao preko dvadeset puštanja uključujući dva najveća ažuriranja s verzijama v2 i v3.

S Bootstrap 2 dodana je responzivna funkcionalnost na cijeli razvojni okvir kao opcionalna stilska tablica. Sa nastavkom u Bootstrap 3 dorađena je responzivnost i dodana je opća zadana responzivnost s „mobilna responzivnost prvo“ pristupom. Sa Bootstrap 4 dodane su glavne promjene za migraciju na Sass i Css flexbox.

Kako bi se počeo koristiti Bootstrap u projektu, prvo se treba uključiti preko css linka koji se postavlja u javnoj index.html datoteci u projektu. Neke od komponenti zahtijevaju korištenje JavaScript funkcija pa se određeni linkovi postavljaju u script komponentu za uključivanje u istoj datoteci.

Bootstrap nudi jedno od mnogih početnih predložaka za počinjanje razvoja. Koristi se HTML5 sintaksa.

Pregled stranice se kreira koristeći Container komponente. Unutar Container komponente doda se atribut fluid tako da se kontejner proširuje preko cijele stranice. Nakon toga se sadržaj stavlja u Row i Col. Row predstavlja redove stranice. Unutar svakog reda se stavljaju Col ili kolone. U svaki red stane 12 kolona.



Slika 7. Primjer slaganja kolona u redovima [3]

Bootrap nudi mnogo gotovih komponenti za inpute, forme, gumbe i dr.

Bootrsap održava mali tim developera na GitHubu.

4.4. Mapbox

Mapbox je poslužitelj prilagođenih online mapa za web stranice i aplikacije. Mapbox je kreator i veliki pogovornik biblioteka otvorenog koda za mape i aplikacije, od kojih je značajnija Mapbox GL-JS za JavaScript biblioteke koja se koristi u ovom projektu.

Startup je nastao radi potrebe ponude kostimiranih mapi za neprofitne korisnike 2010. godine.

Mapbox omogućuje samostalno uređivanje svih komponenti mape kao što su jezera, ceste, gradovi itd. tako da daje korisniku veliku slobodu pri kreiranju individualnih mapi za vlastite potrebe.

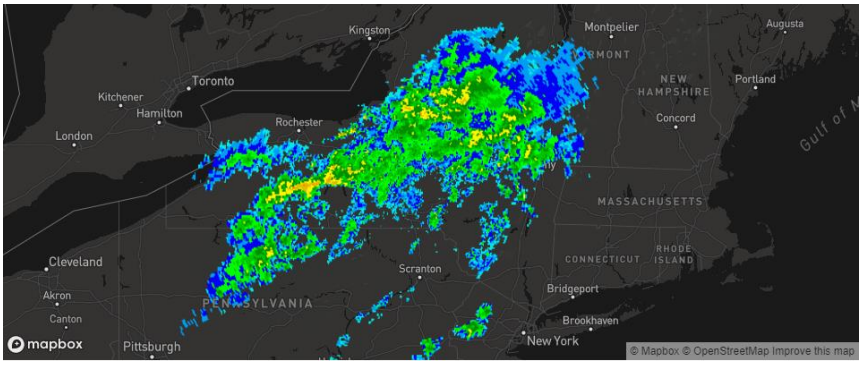
Kako bi se mapa uspješno koristila na aplikaciji, potreban je odgovarajući token za mapu. Tokeni se dodjeljuju ovisno o namjeni korištenja mape i njene verzije. Kako bi se mapa uspješno renderirala, potreban joj je token. Token se unosi u atribut za dohvaćanje mape unutar ReactMapGL komponente (react-map-gl, 2021).

```
<ReactMapGL
  {...viewport}
  mapboxApiAccessToken={process.env.REACT_APP_MAPBOX_ACCESS_TOKEN}
  onViewportChange={(nextViewport) => setViewport(nextViewport)}
  mapStyle="mapbox://styles/fakkkkkk123/ckq6pcalq3jpd17o6c7kiwz2y"
  style={{ position: "absolute" }}
>
```

Slika 8. Unos tokena preko environment varijable

Slika broj 9 prikazuje unos mapbox access tokena preko environment varijable.

Prikaz stanja mape ažurira se preko viewport varijable. Svakom promjenom radnje na mapi ažurira se novi viewport. Ako želimo naglo skočiti na neku lokaciju na mapi možemo preko funkcije za dolazak u novi grad aktivno promijeniti viewport i mapa će prikazivati novi grad.



Slika 9. Prikaz kostimirane mape [2]

4.5. Axios

Axios je npm (Node Package Manager) biblioteka namijenjena za JavaScript. Bazira se na „promise“ HTTP zahtjevima za preglednik i za node.js (npm, 2021).

Funkcionalnosti su joj :

- kreira XMLHTTP zahtjeve s preglednika i node.js-a
- podržava Promise API (prezentacija odgovora koji je eventualno završio ili došao do greške na asinkronoj operaciji i njenoj rezultirajućoj vrijednosti)
- Pretvara odgovor (engl. response) i odgovor podataka
- Automatski pretvara podatke za JSON oblik podataka
- Omogućuje klijentsku podršku za zaštitu od XSRF (Cross-site request forgery)

```
$ yarn add axios
```

Slika 10 Instalacija paketa Axios preko yarn

```
axios.post('/user', {
    firstName: 'Fred',
    lastName: 'Flintstone'
  })
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
```

Slika 11. Prikaz POST zahtjeva pomoću Axiosa [10]

4.6. React-router-dom

React-router-dom je biblioteka namijenjena za kreiranje ruta u React.js okviru.

Biblioteka omogućuje relativne linkove, navigacije i rute. Omogućuje kreiranje ugniježđenih ruta. Sagrađena je od manjih komponiranih dijelova tako da se koriste samo dijelovi koji su potrebni aplikaciji što smanjuje kompleksnost korištenja ove biblioteke.

Omogućuje pristup „location“ i „navigate“ podataka pomoću „useLocation()“ funkcije bilo gdje u aplikaciji. Omogućuje validaciju u parametrima rute, te podudaranje ruta na stanju lokacije (REACT TRAINING / REACT ROUTER, 2021).

```jsx
export default function App() {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li>
              <Link to="/">Home</Link>
            </li>
            <li>
              <Link to="/about">About</Link>
            </li>
            <li>
              <Link to="/users">Users</Link>
            </li>
          </ul>
        </nav>

        {/* A <Switch> looks through its children <Route>s and
            renders the first one that matches the current URL. */}
        <Switch>
          <Route path="/about">
            <About />
          </Route>
          <Route path="/users">
            <Users />
          </Route>
          <Route path="/">
            <Home />
          </Route>
        </Switch>
      </div>
    </Router>
  );
}
```

Slika 12. Primjer Router komponente u App komponenti [8]

Slika 13 prikazuje korištenje Router komponente. Unutar nje se definiraju rute tako da se omotaju u Switch komponentu. Iz Switch komponente se navedu sve postojeće rute i njihova pripadajuća putanja. Unutar određene rute stavlja se ime komponente koja se učitava pri pristupu određenoj putanju.

4.7. Dijagram slučajeva korištenja aplikacije

Glavni akter je korisnik koji može biti novi korisnik ili već postojeći korisnik. Ako je korisnik novi, onda prvo prolazi kroz proces registracije. Nakon toga sljedeći nužan korak je validacija e-pošte korisnika. Nakon što je validacija uspješno provedena, sljedeći korak je prijava korisnika. Nakon što se korisnik prijavio, aplikacija ga postavlja na glavnu stranicu. Sa glavne stranice nalazi se nekoliko opcija koje korisnik može

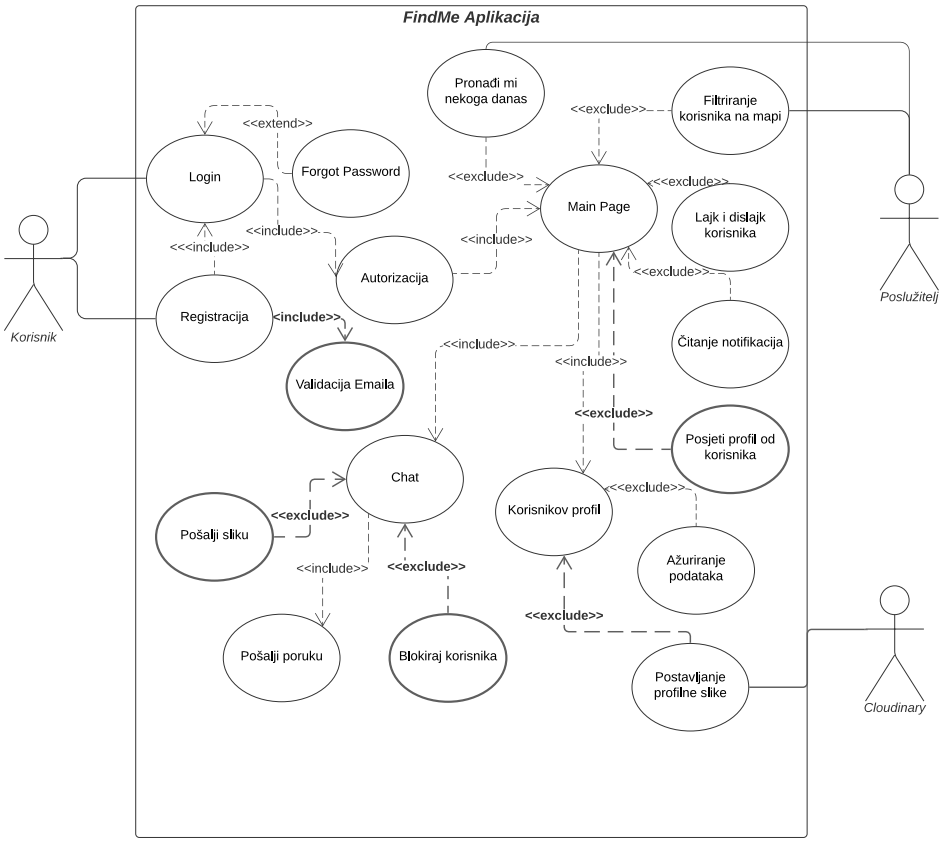
izabrati. Neke od njih su pregled obavijesti, filtriranje korisnika i sl. Sa glavne stranice korisnik odlazi na svoju profilnu stranicu. Također do razgovora može doći preko glavne stranice ili izbornika.

Na javnom profilu korisnika, korisnik može označiti korisnika sa „sviđa mi se“ ili „ne sviđa mi se“ te započeti razgovor s korisnikom ako su se uspješno spojili.

Na korisnikovom profilu korisnik može ažurirati sliku i podatke.

Poruke se šalju korisniku preko „razgovor“ stranice. Ponuđene opcije su blokiranje korisnika i slanje slike.

Sekundarni akteri su poslužitelji koji šalju podatke klijentu. Neke od glavnih zadataka poslužitelja su slanje filtriranih korisnika, slanje najboljih korisnika, slanje notifikacija i dr. Još jedan akter je Cloudinary server koji šalje sve slike na aplikaciju. Slike se prvotno preko zahtjeva spremaju na Cloudinary server, link na njih se sprema na poslužitelja, te se pri učitavanju slike povlače sa Cloudinary servera.



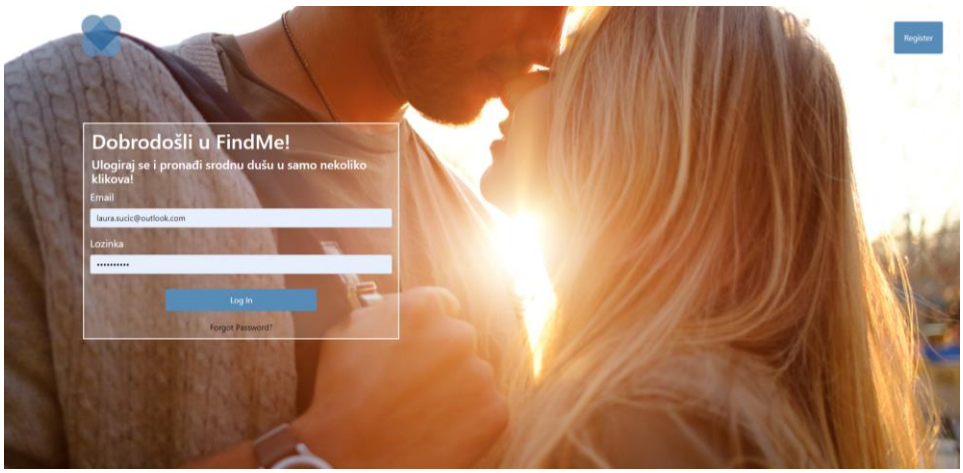
Slika 13. Prikaz dijagrama slučajeva korištenja aplikacije FindMe

5. Arihtektura aplikacije, njeno korisničko sučelje i korisnički zahtjevi aplikacije

Struktura aplikacije je koncipirana na par glavnih stranica od kojih se dalje dolazi do drugih prikaza.

5.1. Uvodna stranica

Uvodna stranica je početna stranica aplikacije koja se prikazuje kada se pokrene aplikacija. Sastoji se od funkcionalnosti prijave i registracije. Sadržava opis aplikacije, korisničkih komentara, moto i vizualni identitet aplikacije. Preko početne stranice korisnik se upoznaje sa aplikacijom, njenim opisom i osnovnim informacijama.



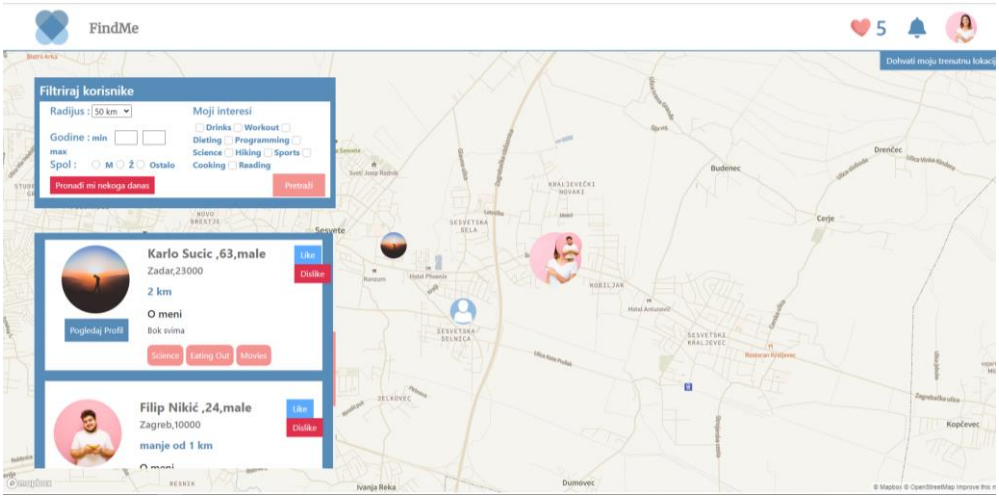
Slika 14. Prikaz početnog dijela uvodne stranice

5.2. Glavna stranica

Glavna stranica sastoji se od zaglavlja aplikacije (engl. Header). U zaglavlju su funkcionalnosti života, obavijesti i gumb sa izbornikom sa korisničkom slikom koje će biti detaljnije objašnjene u poglavlju 6.

Ispod se nalazi mapa s korisnikovim markerom u sredini i markerima ostalih korisnika raspoređenih po lokaciji na mapi. Ovisno o blizini korisnika, toliko će blizu biti markeri.

Na mapi se nalaze filter i kartice s korisnicima na mapi i njihovim skraćenim informacijama. Na karticama korisnik bira koji mu se korisnik sviđa, a koji mu se ne sviđa i shodno tim odlukama odabire određenu akciju. Unutar filtera korisnik odabire jedinice po kojima će filtrirati korisnike. To može biti spol, godine, udaljenost na mapi ili interesi. Ako korisnik ne želi filtrirati sam nego želi da to aplikacija učini umjesto njega, može kliknuti na opciju „Pronađi mi nekoga danas“ gdje će aplikacija umjesto njega odabrati najodgovarajuće korisnika za njega. Više o funkcionalnostima u poglavlju 6. Na karticama se nalaze korisnici sa svojima osnovim informacijama kao što su ime, prezime i udaljenost od korisnika. Ispod inicijala nalazi se kratki opis korisnika i popis njegovih interesa. Sa desne strane korisnika se može označiti sa sviđa mi se ili ne sviđa mi se, te se sa lijeve može kliknuti na gumb „Pogledaj profil“ i detaljnije pogledati profil korisnika.

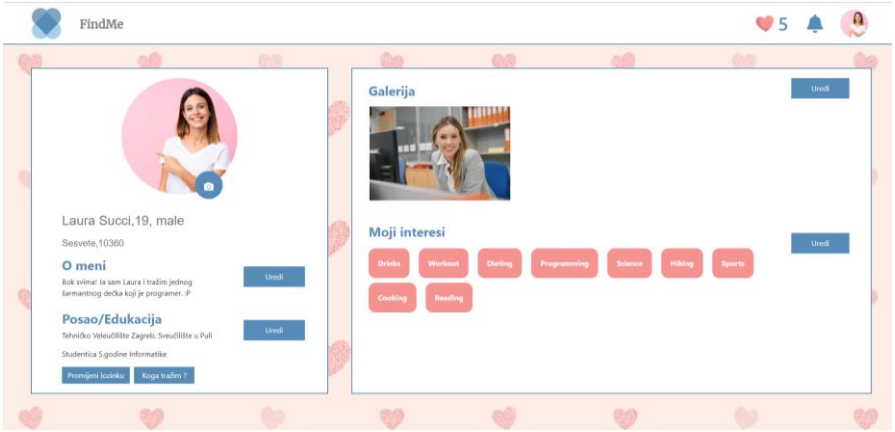


Slika 15. Prikaz glavne stranice aplikacije

5.3. Profilna stranica

Profilna stranica se sastoji od korisnikovih podataka. Na njemu on može izmijeniti svoje podatke, sliku, galeriju i interese. Galerija se sastoji od maksimalno šest fotografija koje se pregledavaju klikom na sliku. Interesi se sastoje od kategorija. Glavne kategorije su priroda, sport, hrana, zabava i učenje. Iz svake kategorije korisnik

bira listu interesa. Na profilu korisnik može izmijeniti lozinku, tako da unese staru, postojeću lozinku i unese novu željenu lozinku. Moguće je uređivanje podataka opisa, edukacije i zanimanja. To se učini tako da se klikne na gumb uredi pored određenog polja koje se želi izmijeniti. Sljedeća slika pod rednim brojem 17. prikazuje proflinu stranicu prijavljenog korisnika.



Slika 17. Profilna stranica

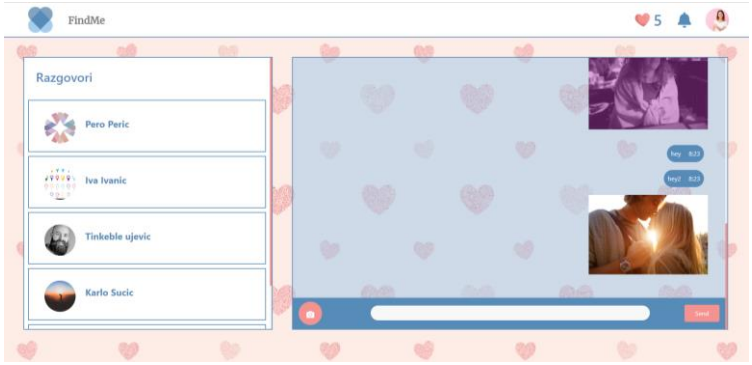
5.4. Korisnikova stranica

Korisnikova stranica je stranica korisnika kojeg pretražujemo. Na profilu se nalaze sve njegove informacije. One informacije koje nije ispunio će prikazivati prikladnu poruku gdje je određeno polje prazno ili ispunjeno. Na profilu možemo označiti korisnika sa sviđa mi se i ne sviđa mi se, te započeti razgovor sa korisnikom preko njegovog profila gdje ako su se uspješno spojili, aplikacija šalje korisnika na stranicu za razgovor.

Slika 16. Korisnikova stranica

5.5. Stranica za razgovor

Stranica za razgovor istoimeno sastoji od razgovora sa korisnicima. Lijevo se nalaze svi dostupni razgovori s korisnicima gdje se jasno vidi koji korisnik je novi, koji korisnik je blokiran i s kojim korisnicima razgovaramo. Na desnoj strani nalzi se glavni razgovor sa korisnikom na kojeg smo kliknuli sa lijeve strane. Unutar razgovora prikazane su poruke koje su izmijenili korisnici, te dolje se nalazi polje gdje se unosi poruka, te dvije tipke. Lijeva tipka prikazuje dodavanje slike, a desni slanje poruke. U gornjem kutu nalazi se tipka Blokiraj korisnika gdje korisnik može blokirati korisnika s kojim priča. Ako se korisnik blokira, on sa lijeve strane prelazi u listu blokiranih korisnika.



Slika 17. Stranica za razgovor

5.6. Korisnički zahtjevi

Korisnički zahtjevi na poslužitelja su:

- Zahtjev za prijavu i registraciju
- Ažuriranje korisničkih podataka
- Prikaz korisnika na mapi
- Filtriranje korisnika na mapi
- Pronalaženje odgovarajućih korisnika

- Komunikacija sa korisnikom
- Oznake za sviđa mi se i ne sviđa mi se korisnik
- Obavještavanja korisnika o novim porukama i sviđanjima

6. Razrada funkcionalnosti klijentskih komponenti

6.1. Prijava

Funkcionalnost prijava se sastoji od procesa prijave korisnika. Kako bi se korisnik prijavio mora unijeti svoju email adresu i svoju zaporku. Korisnik unosi svoj odgovarajući email i zaporku. Ukoliko dođe do greške, korisniku se ispisuje odgovarajuća greška. Zaporka se mora sastojati od minimalno osam znakova, barem jedan broj i barem jedno veliko slovo. Nakon što se korisnik uspješno prijavio, dodjeljuje mu se accessToken, refreshToken i dobiva sve svoje podatke.

Za kontrolu forme koristi se Formik i Yup biblioteka. Formik uzima LoginSchemu u kojoj se nalaze sve varijable koje su nužne za login. Unutar scheme definirane su varijable emial i password te njihova veličina, tip i razina potrebe.

```
const LogInSchema = Yup.object().shape({
  email: Yup.string()
    .email("Invalid email")
    .required("Email adresa je obavezna."),
  password: Yup.string().min(6).max(20).required("Lozinka je obavezna."),
});
```

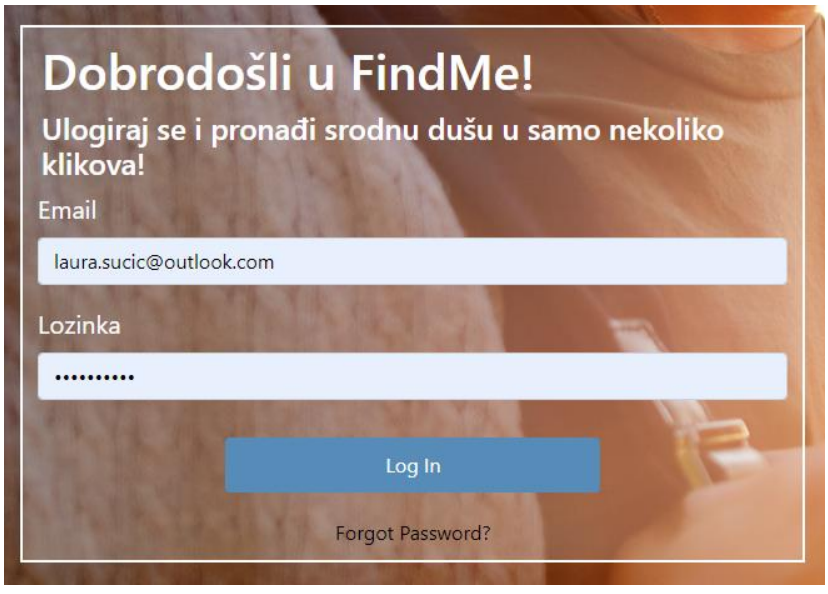
Slika 18. Prikaz LoginSheme

Slika 19. Prikaz Formik inicijalizacije

Slika 21 prikazuje Formik komponentu. Prilikom početka korištenja formika daju mu se inicijalne vrijednosti za email i lozinku. Nakon toga definira se shema koja se koristi u validaciji. Nakon što se pozove onSubmit event, u values objekt stavljaju se vrijednost email-a i zaporke te se na HTTP post zahtjevi šalju na rutu spremljenu u environment okruženju pod nazivom REACT_APP_LOGIN. Nakon logina se postavlja refreshToken iz dobivenih podataka, te se postavlja također i accessToken.Podaci o korisniku se spremaju u getUserData() funkciju.

```
{({ errors, touched }) => (
  <Form>
    <label
      htmlFor="email"
      style={{ color: "white", fontSize: "20px" }}
    >
      Email
    </label>
    <Field
      name="email"
      type="email"
      className="form-control form-group"
    />
    {errors.email && touched.email ? (
      <Alert variant="warning">{errors.email}</Alert>
    ) : null}
    <label
      htmlFor="password"
      style={{ color: "white", fontSize: "20px" }}
    >
      Lozinka
    </label>
    <Field
      name="password"
      type="password"
      className="form-control form-group"
    />
    {errors.password && touched.password ? (
      <Alert variant="warning">{errors.password}</Alert>
    ) : null}

    <div style={{ textAlign: "center", marginTop: "30px" }}>
      <Button
        type="submit"

        size="md"
        style={{ width: "50%", padding: "10px",backgroundColor:"#578BB8",border:"none" }}
      >
        Log In
      </Button>
    </div>
```

Slika 20. Prikaz login forme



Slika 21. Prikaz login korisnika

Slika broj 23 prikazuje korisničko sučelje za ulogiravanje korisnika.

6.2. Zaboravljena lozinka

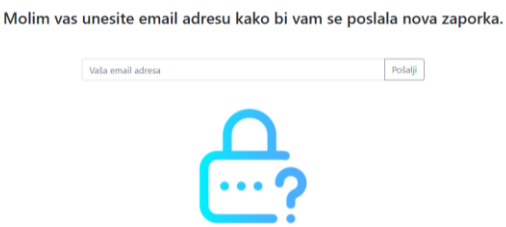
Zaboravili ste lozinku funkcionalnost se aktivira kada se klikne na gumb „Forgot Password“ sa login prozora. Tada korisnik mora unijeti svoj Email i doći će mu pošta sa novom šifrom.

Nakon nove dodijeljene lozinke, korisnik se s njom mora prijaviti i ako ju želi ponovo promijeniti može to napraviti na svom profilu pod opcijom „Promijeni lozinku“.

```
const sendToEmail = async (value) => {

  try {
    const res = await axios.post(
      process.env.REACT_APP_FORGOT_PASSWORD,
      {
        email: value,
      },
      {
        headers: {
          authorization: accessToken,
        },
      }
    );
    console.log(res.data);
  } catch (error) {
    errorHandler(error);
  }

  history.push("/");
};
```

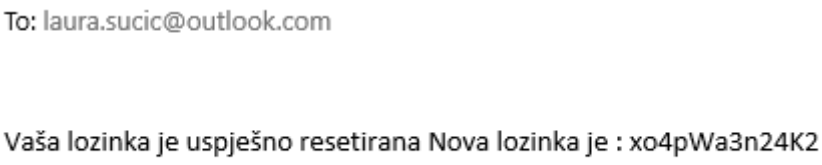
Slika 22. Slanje emaila korisniku preko post zahtjeva

Slika 24 prikazuje funkciju sendToEmail() koja se poziva nakon što korisnik unese email i stisne na gumb Pošalji. Nakon ulaska u funkciju, šalje se post zahtjev na rutu REACT_APP_FORGOT_PASSWORD na poslužiteljski dio gdje poslužiteljski dio (Kožić, 2021.) šalje odgovarajući email korisniku s novom šifrom.



Slika 23. Forgot Password komponenta

Slika broj 25 prikazuje korisničko sučelje za Forgot Password.



Slika 24. Primjer poslane pošte sa novom lozinkom

Slika broj 26 prikazuje uspješno poslan email na laura.sucic@outlook.com i poruku kako je lozinka uspješno resetirana na novu lozinku. Korisnik se sada mora ulogirati sa novom danom lozinkom i kasnije ju može promijeniti u postavkama profila.

6.3. Registracija

Registracija se bavi registriranjem korisnika. Korisnik pri registraciji unosi ime, prezime, email adresu, lozinku, potvrđenu lozinku, grad, zip, datum rođenja i spol.

Sve stavke su obavezne.

Dodatna funkcionalnost je automatski ispis poštanskog broja kod odabira grada.

```
const CustomInputComponent = (props) => (
  <Typeahead
  id="basic-typeahead-single"
  onChange={handleClick}
  options={options.map(i=>i.city)}
  placeholder="Izaberite grad..."
  selected={singleSelections}
  required

/>
);
```

Slika 25. Typehead komponenta za automatsko ispunjavanje poštanskog broja

Slika broj 27 prikazuje Typehead komponentu preuzetu od Bootstrapa koja rješava funkcionalnost automatskog ispunjenja poštanskog broja i biranja grada. Komponenti se pod opcije doda polje „option“ koje su sebi ima grad i njegov zadani poštanski broj. Uzima se samo grad te se selektirani grad sprema u varijablu singleSelections.

```
const RegisterShema = Yup.object().shape({
  firstName: Yup.string().min(2).max(60).required("Ime je obavezno."),
  lastName: Yup.string().min(2).max(60).required("Prezime je obavezno."),
  email: Yup.string()
    .email("Kriva email adresa.")
    .required("Email adresa je obavezna."),
  password: Yup.string()
    .required("Molim vas unesite svoju lozinku.")
    .matches(
      /^.*(?=.{8,})(?=.*\d)((?=.*[a-z]){1})((?=.*[A-Z]){1}).*$/,
      "Lozinka se mora sastojati od barem 8 znakova, jedno veliko slovo i jedan broj."
    ),
  confirmPassword: Yup.string()
    .required("Molim vas potvrdite svoju lozinku.")
    .oneOf([Yup.ref("password"), null], "Loiznke se ne podudaraju."),
  gender: Yup.string().required("Spol je obavezan."),

  day: Yup.string().required("Unesite dan."),
  month: Yup.string().required("Unesite mjesec."),
  year: Yup.string().required("Unesite godinu."),


});
```

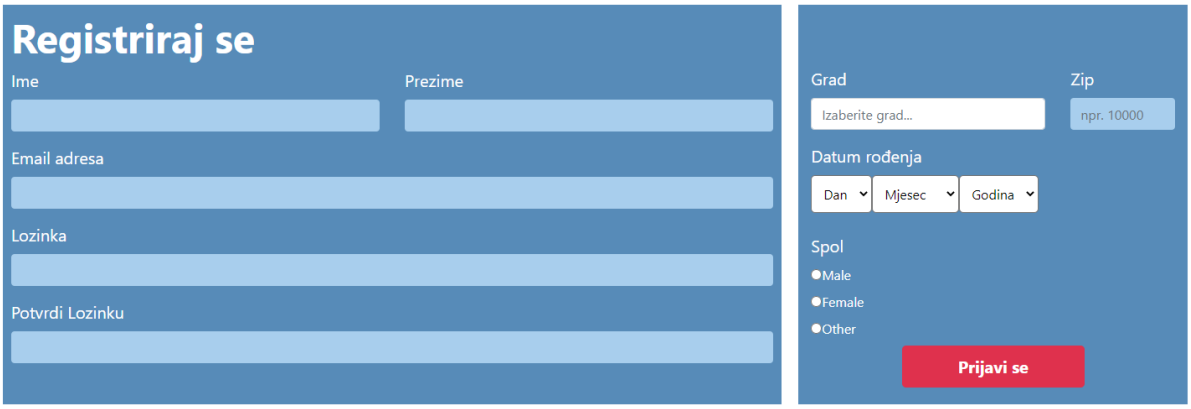
Slika 26. RegisterShema za potrebe validacije registracije

Kao i prethodni primjer za login, slika broj 28 prikazuje Register shemu za potrebe registracije sa svim vrijednostima i ograničenjima. Forma se vrši preko Formik-a i Yup-a na isti način kao i kod logina.

```
onSubmit={async (values) => {


  if(_.isEmpty(singleSelections) || _.isEmpty(zip)) {setCityError(true);return}

  values.dob = reformatDate(values.day, values.month, values.year);

  delete values.day;
  delete values.month;
  delete values.year;
  delete values.confirmPassword;



  try {
    const res = await axios.post(
      process.env.REACT_APP_REGISTER_USER,
      {...values,city:singleSelections[0],zip:zip}
    );
    console.log(res.data);
  } catch (error) {
    errorHandler(error);
  }

  setLoadEmailValidation(true);
}}
```

Slika 27. onSubmit funkcija za registraciju

Nakon danih vrijednosti potrebnih za registraciju, poziva se post zahtjev na rutu REACT_APP_REGISTER_USER te mu se pošalje objekt sa potrebnim vrijednostima. Nakon uspješne registracije učitava se stranica s potvrdom emaila, a poslužitelj unosi korisnika u bazu. (Kožić,2021.)



Slika 28. Prikaz registracije

Slika broj 30 prikazuje korisničko sučelje za registraciju korisnika.

6.4. Dovršavanje registracije nakon prijave

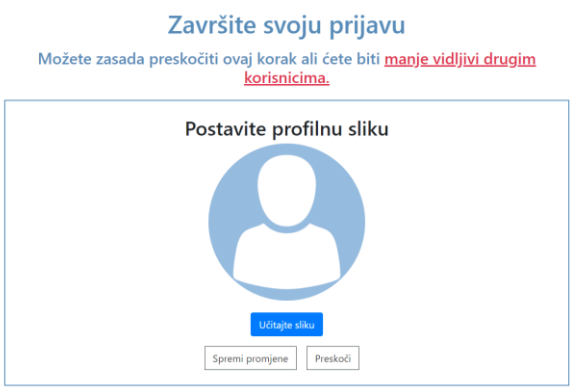
Ova funkcionalnost se aktivira samo pri prvom loginu. Ona omogućuje korisniku brže ispunjavanje dodatnih informacija kako bi odmah mogao početi kvalitetnije koristiti aplikaciju. Sastoji se od unosa slike, opisa i seksualne orijentacije. Svi koraci se mogu preskočiti i naknadno ispuniti na profilu korisnika.

```
function SexualOrientation() {
  return (
    <React.Fragment>
      <Row
        style={{
          justifyContent: "center",
          border: "2px solid black",
          height: "500px",
          textAlign: "center",
        }}
      >
        <Col>
          <h1 style={{ marginTop: "30px" }}>Add Your Sexual Orientation</h1>

          <Row>
            <Col style={{ padding: "20px", height: "300px" }}>

            <div>
            <input type="radio" value="Males" name="gender" ref={ref1}/> {sexes[0]}
            <input type="radio" value="Females" name="gender" ref={ref2}/> {sexes[1]}
            <input type="radio" value="Both" name="gender" ref={ref3} /> {sexes[2]}
          </div>

            </Col>
          </Row>

          <Button
            size="lg"
            variant="outline-dark"
            style={{ marginRight: "20px", borderRadius: "1px" }}
            onClick={setupSexualOrientation}
          >
            Submit
          </Button>
          <Button
            size="lg"
            variant="outline-dark"
            onClick={loadMain}
            style={{ marginRight: "20px", borderRadius: "1px" }}
          >
            Skip
          </Button>
        </Col>
      </Row>
    </React.Fragment>
  );
}
```

Slika 31. Prikaz primjera za ispunjavanje seksualne orijentacije

Slika broj 32 prikazuje komponentu za unos seksualne orijentacije korisnika. Sastoji se od triju vrijednosti označenih radio inputima. Nakon što korisnik odredi vrijednost poziva se funkcija setupSexualOrientation() koja će spremiti vrijednosti korisnika.

Na onClick() funkciji, poziva se loadMain koja učitava glavnu Main stranicu aplikacije.



Slika 32. Prikaz dodavanja profilne slike

6.5. Access token

Kako bi se ispunili svi zahtjevi koji se izvršavaju kada je korisnik ulogiran, u zaglavlje svakog zahtjeva potrebno je poslati accessToken.

Nakon što se korisnik ulogira, on dobiva jedinstveni accessToken i refreshToken. AccessToken služi kako bi se zahtjevi na klijentski dio aplikacije mogli izvršiti, a refreshToken služi kako bi korisnik, bez novog ulogiravanja, mogao dobiti novi accessToken. Nakon 15 minuta accessToken ističe i radi se novi zahtjev na poslužitelju pomoću refreshTokena. Ako refreshToken nije valjan, accessToken se neće promijeniti.

```
const getAcessToken = async ()=>{

  //CHECK IF REFRESH TOKEN
  if(refreshToken != null){

    try{
      const res = await axios.get(
        process.env.REACT_APP_GET_ACCESS,
        {
          headers: {
            "authorization": refreshToken,
          }
        }
      );
      const { data, message, length } = res.data;

      await setAccessToken(data)

      getUserData(data)

    }
    catch(error){
      errorHandler(error)
      //if error 444 call get access

    }

  }
}
```

Slika 29. Postavljanje access tokena

```
const getUserData = async (access = accessToken)=>{
  try{

    const res = await axios.get(
      process.env.REACT_APP_GET_USER_DATA,
      {
        headers: {
          "authorization": access,
        }
      }
    );
    setUserData(res.data)

  }
  catch(error){
    errorHandler(error)
  }
}
```

Slika 30. Dohvaćanje podataka od korisnika pomoću access tokena

6.6. Mapa sa korisnicima u blizini

Mapa je jedna od najglavnijih funkcionalnosti aplikacije. Ona se učitava nakon što se korisnik ulogira i na njoj se sastoje svi korisnici u blizini.

Korisnici su prikazani markerima sa slikama i vidljiva se njihova približna lokacija od korisnika.

Svi korisnici na mapi također se nalaze na karticama s lijeve strane mape tako da se prikažu njihove glavne informacije poput imena, interesa, godina, spol itd.

```
<ReactMapGL
  {...viewport}
  mapboxApiAccessToken={process.env.REACT_APP_MAPBOX_ACCESS_TOKEN}
  onViewportChange={(nextViewport) => setViewport(nextViewport)}
  mapStyle="mapbox://styles/fakkkkkk123/ckq6pcalq3jpd17o6c7kiwz2y"
  style={{ position: "absolute" }}
>
  <Button
    variant="primary"
    onClick={() => getCurrentLocation()}
    style={{
      float: "right",
      borderRadius: "0px",
      backgroundColor: "#578BB8",
      border: "none",
      zIndex: "100",
    }}
  >
    Dohvati moju trenutnu lokaciju
  </Button>
  <div style={{ clear: "both" }}></div>
  <Source id="bull" type="geojson" data={geojson}>
    <Layer
      id="nes"
      type="fill"
      source="bull"
      paint={{
        "fill-color": "#228b22",
        "fill-opacity": 1,
      }}
    ></Layer>
  </Source>
```

Slika 31. Prikaz ReactMapGL komponente

Slika broj 35 prikazuje ReactMapGL komponentu. Ona se sastoji od tokena koji omogućuje prikaz mape, stila mape koji je preuzet sa linka prikazanog na slici, te

onViewPortChange koji mijenja viewport vrijednost cijele mape bilo kakvom promjenom.

```
<Marker
  latitude={userData.lastKnownLocation.coordinates[0]}
  longitude={userData.lastKnownLocation.coordinates[1]}
  offsetLeft={-50}
  zoom={11}
>
  <img
    width="80px"
    id="profilePhoto"
    src={resizeCloudinary(userImage)}
    onClick={(event) => {}}
  />
</Marker>
{usersMarkers.map((user, i) => {
  if (user._id === userData._id) return <></>;

  return (
    <Marker
      key={i}
      latitude={user.lastKnownLocation.coordinates[0]}
      longitude={user.lastKnownLocation.coordinates[1]}
      zoom={11}
      onClick={() => onMarkerClick(user._id)}
      offsetRight={-50}
    >
      <img
        width="40px"
        src={resizeCloudinary(user.imageUrl)}
        id="mapPhoto"
      />
    </Marker>
  );
})}
```

Slika 34. Prikaz markera na mapi

Slika broj 34 prikazuje markere na mapi. Mapa se sastoji od dva markera. Marker od glavnog korisnika koji je ulogiran i njegove pripadajuće slike. Marker se kreira od korisnikovih koordinata. Drugi marker se sastoji od više markera, a to su svi markeri koji nemaju isti id kao i korisnik što označava sve osobe na mapi s kojima korisnik može komunicirati. Nakon učitavanja slika, one promijene svoju veličinu tako da se optimizira čekanje učitavanja na mapi.

6.7. Dohvat trenutne lokacije

Korisnik pri registraciji unosi svoju lokaciju preko grada i poštanskog broja. Ako se u bilo kojem slučaju korisnik pomakne s lokacije a želi upoznati korisnike u blizini onda stisne gumb „Dohvati moju trenutnu lokaciju“ koja uzima lokaciju njegovog uređaja s malim pomakom. Tako korisnik može fleksibilno mijenjati lokaciju gdje god se nalazi.

```
const getCurrentLocation = () => {
  if ("geolocation" in navigator) {
    console.log("Available");
  } else {
    console.log("Not Available");
  }

  navigator.geolocation.getCurrentPosition(async function (position) {
    console.log("Latitude is :", position.coords.latitude);
    console.log("Longitude is :", position.coords.longitude);

    try {
      const res = await axios.patch(
        process.env.REACT_APP_USER_LOCATION,
        {
          latitude: position.coords.latitude,
          longitude: position.coords.longitude,
        },
        {
          headers: {
            authorization: accessToken,
          },
        }
      );
      console.log(res.data);
      setUserData(res.data);
    } catch (error) {
      errorHandler(error);
    }
  });
};
```

Slika 32. Dohvat korisnikove lokacije

Slika broj 35 prikazuje funkciju getCurrentLocation() koja se poziva kada korisnik stisne na odgovarajući gumb na mapi. Na početku se provjerava ima li korisnik geolokaciju upaljenu. Ako ima onda će se dohvatiti, a ako nema neće.

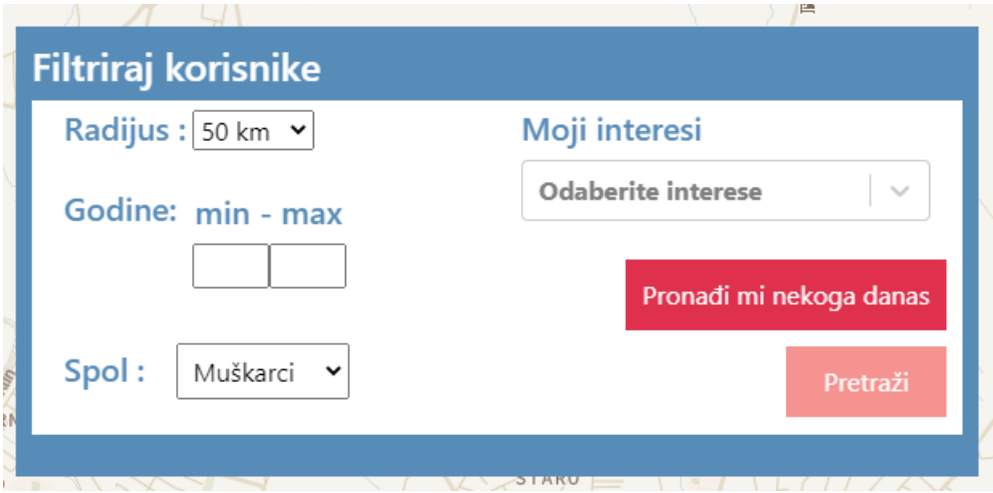
Nakon toga navigator objekt uzima iz preglednika korisnikovu lokaciju te je sprema u vrijednosti koordinata. Nakon toga radi se patch zahtjev koji ažurira korisnikovu lokaciju na novu trenutnu lokaciju. Za potrebe ažuriranja potreban je valjani accessToken korisnika.

Nakon uspješnog zahtjeva ažuriraju se podaci korisnika preko setUserData() i ponovno se renderira komponenta mape.

6.8. Filtriranje korisnika na mapi

Još jedna od glavnih funkcionalnosti je filtriranje korisnika na mapi. Ona omogućuje što detaljniji odabir preferencijalnog korisnika koji se traži. Korisnika se može pretražiti po radijusu, spolu, godinama i korisnikovim interesima.

Filtriranje po interesima omogućuje što veću sličnost i kompatibilnost među korisnicima.



Slika 33. Prikaz filtera za korisnike

Slika 36 prikazuje korisničko sučelje za filtiranje korisnika na mapi.

```
const filterUsers = async () => {
  const obj = {
    range: rangeRef.current.value,
  };

  if (!_.isNull(radioVal)) {
    obj["sex"] = radioVal;
  }

  if (!_.isEmpty(mapHobbies)) {
    obj["interests"] = mapHobbies.map((i) => {
      return { category: "1", interest: i };
    });
  }

  if (!minAgeRef.current.value == "" && !maxAgeRef.current.value == "") {
    console.log(minAgeRef.current.value);
    obj["age"] = {
      min: minAgeRef.current.value,
      max: maxAgeRef.current.value,
    };
  }

  console.log(obj);

  if (Object.keys(obj).length == 1 && Object.keys(obj)[0] == "range") {
    console.log("hej range");

    try {
      const res = await axios.get(
        process.env.REACT_APP_MAP + "?range=" + obj.range,
        {
          headers: {
            authorization: accessToken,
          },
        }
      );

      console.log(res.data);
      setMarkers(res.data);
    } catch (error) {
      errorHandler(error);
    }
```

Slika 37. Provjera objekta prije slanja zahtjeva

Prije nego li se pošalje zahtjev za filtriranje na poslužitelja (Kožić,2021.), radi se provjera koje su vrijednosti označene na slici broj 37. Radijus se uvijek uzima i on je uvijek zadan. Inicijalna vrijednost mu je 50 kilometara. Nakon poslanog radijusa, postavljaju se novi markeri sa setMarkers() kako bi se promijenili podaci.

```
    try {
      const res = await axios.get(
        process.env.REACT_APP_MAP + "?range=" + obj.range,
        {
          headers: {
            authorization: accessToken,
          },
        }
      );

      console.log(res.data);
      setMarkers(res.data);
    } catch (error) {
      errorHandler(error);
    }
  } else {
    try {
      const res = await axios.post(process.env.REACT_APP_MAP, obj, {
        headers: {
          authorization: accessToken,
        },
      });

      console.log(res.data);
      setMarkers(res.data);
    } catch (error) {
      errorHandler(error);
    }
  }
};
```

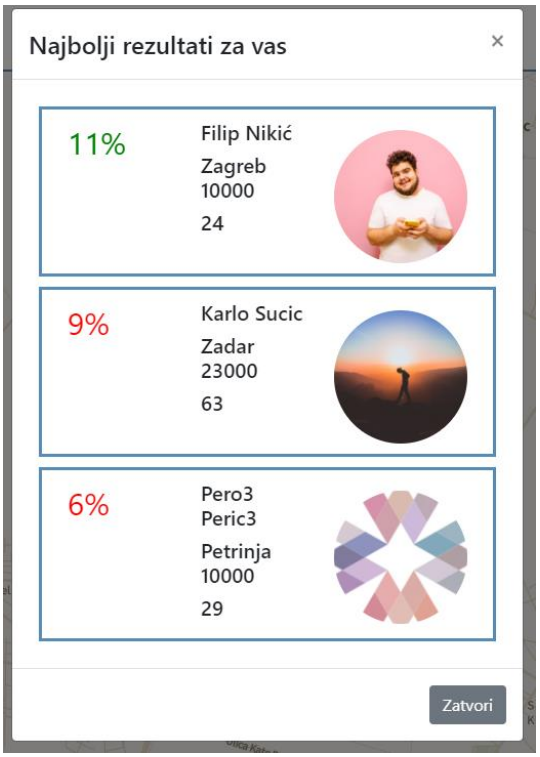
Slika 34. Slanje filtera na poslužitelja

Nakon što su se izbacile vrijednosti koje nisu označene i postavile one koje jesu korisnik može filtrirati korisnike po bilo kojim vrijednostima. Slika broj 38 prikazuje slanje radijusa i slanje ostalih vrijednosti na poslužitelja (Kožić,2021.) Nakon što je određen objekt poslan, poslužitelj ga šalje natrag sa novim korisnicima koji odgovaraju filterima od korisnika.

6.9. Pronalaženje korisnika sa sličnim atributima

„Pronađi mi nekoga danas“ funkcionalnost namijenjena je za korisnike koji ne žele sami ručno filtrirati i tražiti korisnike nego želi da to aplikacija napravi umjesto njih kako bi im našla najoptimalniji „match“. Ona uzima sve podatke od korisnika i pokušava ih spojiti po svim atributima. Dobiveni rezultati su poredani po postocima kompatibilnih bodova. Funkcionalnost „Pronađi mi nekoga danas“ obavlja se isključivo na

poslužiteljskoj strani aplikacije (Kožić,2021.). Nakon što se uspješno dobiju rezultati s klijentskog dijela, dobivaju se odgovarajući rezultati prikazani na slici 37.



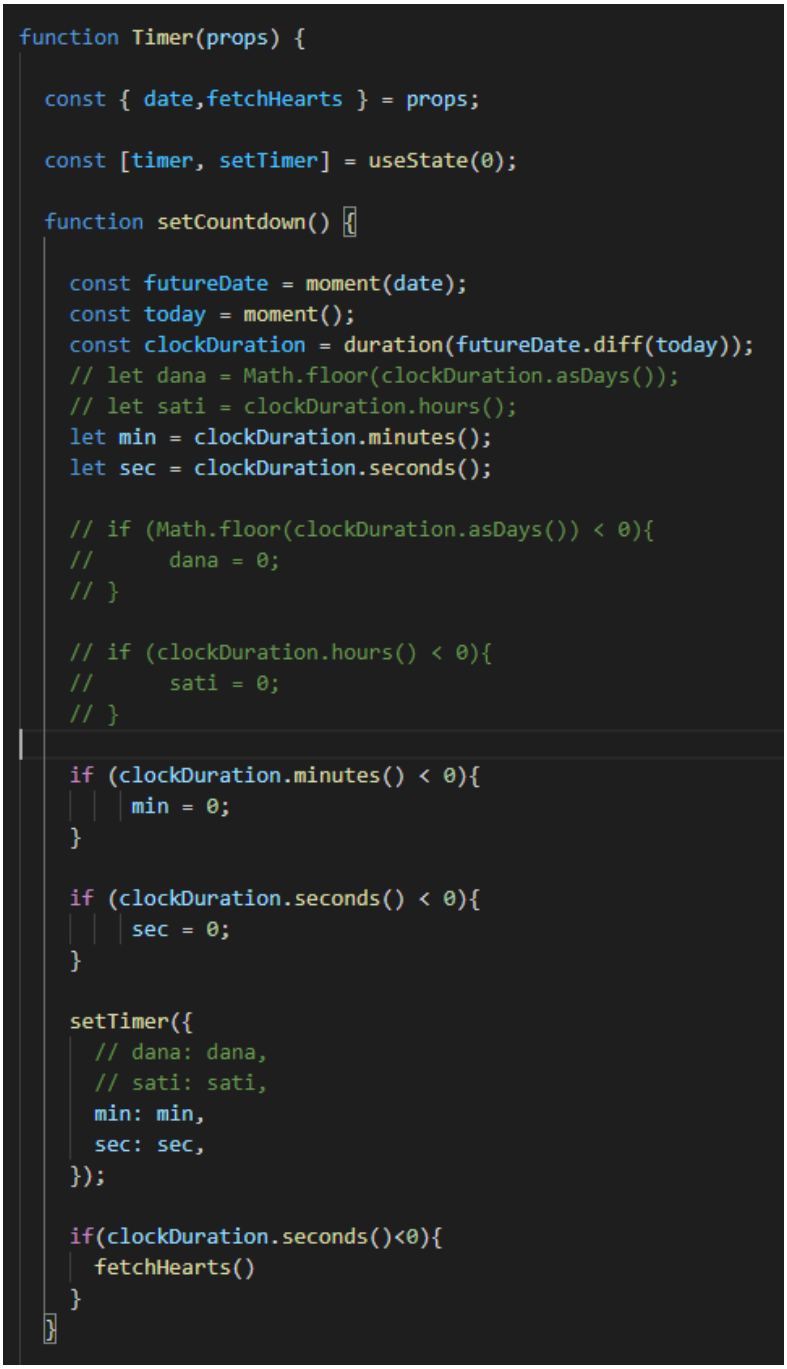
Slika 35. Prikaz najboljih trenutnih rezultata za korisnika

6.10. Sustav života

Korisniku je dodijeljeno pet srca. Nakon svakog sviđa mi se, ne sviđa mi se oznaka i započinjanja razgovora, korisnik gubi po jedno srce. Sustav gamifikacije je napravljen tako da se motivira korisnika da što pametnije odabere koga će odabrati ili odbiti da bi se spriječio osjećaj trake ljudi i tako da korisnici ne bi sve ljude označili sa sviđa mi se i tako bi imali previše ljudi za komuniciranje i manjak fokusa na stvaranje kvalitetnog odnosa. Srca ili životi se pune nakon svakih 15 minuta.

Slika 36. Prikaz života

Slika 37. Prikaz punjenja života



```
function Timer(props) {

  const { date,fetchHearts } = props;

  const [timer, setTimer] = useState(0);

  function setCountdown() {

    const futureDate = moment(date);
    const today = moment();
    const clockDuration = duration(futureDate.diff(today));
    // let dana = Math.floor(clockDuration.asDays());
    // let sati = clockDuration.hours();
    let min = clockDuration.minutes();
    let sec = clockDuration.seconds();

    // if (Math.floor(clockDuration.asDays()) < 0){
    //      dana = 0;
    // }

    // if (clockDuration.hours() < 0){
    //      sati = 0;
    // }

    if (clockDuration.minutes() < 0){
        min = 0;
    }

    if (clockDuration.seconds() < 0){
        sec = 0;
    }

    setTimer({
      // dana: dana,
      // sati: sati,
      min: min,
      sec: sec,
    });

    if(clockDuration.seconds()<0){
      fetchHearts()
    }
  }
```

Slika 38. Prikaz Timer komponente

Kao što je prikazano na slici broj 42, Timer je komponenta koja prati odbrojavanje do novog srca. Za potrebe odbrojavanja koristi se moment biblioteka. Iz props objekta uzimamo funkciju fetchHearts koja nam vraća koliko još imamo vremena do sljedećeg srca. Sa funkcijom setCountdown() postavljaju se minute i sekunde te se započinje odbrojavanje. U trenutku kada su sekunde veće od nule ponovo zovemo fetchHearts() funkciju.

```
return(
    <Container>



        {Object.keys(timer).map((key, i) => (
            <span className="timer" >
                <span className="key">
                    {timer[key]}
                </span>
                <span>
                    {key.toLowerCase()}
                </span>
            </span>



        ))}


    </Container>
)
```

Slika 39. Prikaz Timer render dijela

Slika broj 43 prikazuje renderirani dio timera koji se prikazuje na stranici. Sastoji se od keys-a koji se dodjeljuju timeru i svaki key predstavlja određenu stavku vremena od najmanjeg do najvećeg.

```
const fetchHearts = async () => {
  try {
    const res = await axios.get(process.env.REACT_APP_GET_USER_POINTS, {
      headers: {
        authorization: accessToken,
      },
    });

    console.log("lifes");
    console.log(res.data);
    setUserPoints(res.data);


    setLifes(res.data.lifes);
    setTimeToFill(res.data.nextHeartAt);
    setNotifications(res.data.notifications);
    setChatNotifications(res.data.chat_notifications);

    let arr  = [...res.data.notifications,...res.data.chat_notifications]
    arr = arr.sort((a,b)=> b.date-a.date )
    console.log(arr)
    setNotifications(arr)

  } catch (error) {
    errorHandler(error);
  }
};
```

Slika 40. Prikaz funkcije fetchHearts

Funkcija na slici 44 featchHearts() uzima sa poslužitelja (Kožić,2021.) get zahtjev na rutu REACT_APP_GET_USER_POINTS, te prima odgovarajući accessToken od korisnika da može pristupiti podacima. Kada se zahtjev zaprimi, postavlja se userPoints objekt sa setUserPoints(). Unutar tog objekta nalaze se informacije o broju srca (života), sljedeće vrijeme za novo srce, te notifikacije. Svakom promjenom funkcije fetchHearts() mijenja se timer i tako gubimo ili dobivamo srca.

6.11. Označavanje korisnika sa sviđa mi se

Korisnika se preko gumba Like može označiti sa sviđa mi se te ako se oba korisnika isto označe, dolazi do oglašavanja da se dogodio spoj (engl. „match“).

```
const likeUser = async (id) => {
  try {
    const res = await axios.get(process.env.REACT_APP_LIKE_USER + `/${id}`, {
      headers: {
        authorization: accessToken,
      },
    });
    console.log(res.data);
    setUserPoints(res.data);
    const newUserPoints = { ...userPoints };
    newUserPoints.lifes = res.data.lifes;
    newUserPoints.nextHeartAt = res.data.nextHeartAt;
    setUserPoints(newUserPoints);

    setButtonLiked(true);
    //CHECK MATCH WITH USER

    try {
      const res = await axios.get(process.env.REACT_APP_MATCH + `/${id}`, {
        headers: {
          authorization: accessToken,
        },
      });
      console.log(res.data);
      if (res.data) {
        console.log(setModalID);
        setModalID(id);
        handleShow();
      }
    } catch (error) {
      errorHandler(error);
    }
  } catch (error) {
    errorHandler(error);
  }
};
```

Slika 41. Prikaz likeUser funkcije

Na slici broj 45 prikazana je funckija likeUser. Ona kao argument uzima id od korisnika kojeg smo označili sa sviđa mi se. Nakon oznake radi se get zahtjev na poslužitelja (Kožić,2021.) na rutu REACT_APP_LIKE_USER plus id od korisnika kojeg smo označili. Nakon dobivenih podataka ponovo se ažurira userPoints objekt i gumb se postavlja na označeni gumb sa „like“ koji se onda isključi kada smo korisnika odabrali.

```
const [buttonLiked, setButtonLiked] = useState(
  userPoints.liked.includes(user._id)
);
```

Slika 42. Ovisnost gumba o lajku korisnika

Kao što je prikazano na slici broj 46, gumb se postavlja na isključen ili uključen ako se korisnik nalazi u polju liked unutar userPoints objekta. Ako se nalazi onda se gumb isključi, a ako ne onda se uključi i to nam omogućuje da označimo sa sviđa mi se korisnika.

6.12. Označavanje korisnika sa ne sviđa mi se

Korisnika se može također odbiti i tada se taj korisnik više neće prikazivati na mapi.

```
const dislikeUser = async (id) => {
  try {
    const res = await axios.get(
      process.env.REACT_APP_DISLIKE_USER + `/${id}`,
      {
        headers: {
          authorization: accessToken,
        },
      }
    );
    console.log(res.data);

    setUserPoints(res.data);

    const newU = users.filter((el) => {
      return el._id !== id;
    });

    setUsers(newU);

  } catch (error) {
    errorHandler(error);
  }
};
```

Slika 47. Prikaz funkcije dislikeUser

Kao što je prikazano na slici broj 47, dislikeUser funckija također uzima id od korisnika kojeg smo odbili, no nakon izvršenog zahtjeva get na rutu REACT_APP_DISLIKE_USER/id filtriramo korisnika kojeg smo odbili, tako da ga izbacimo iz polja kartica gdje se podudara id korisnika u tom polju.

Slika 43. Prikaz kartice korisnika sa opcijama lajk,dislajk i pregled profila

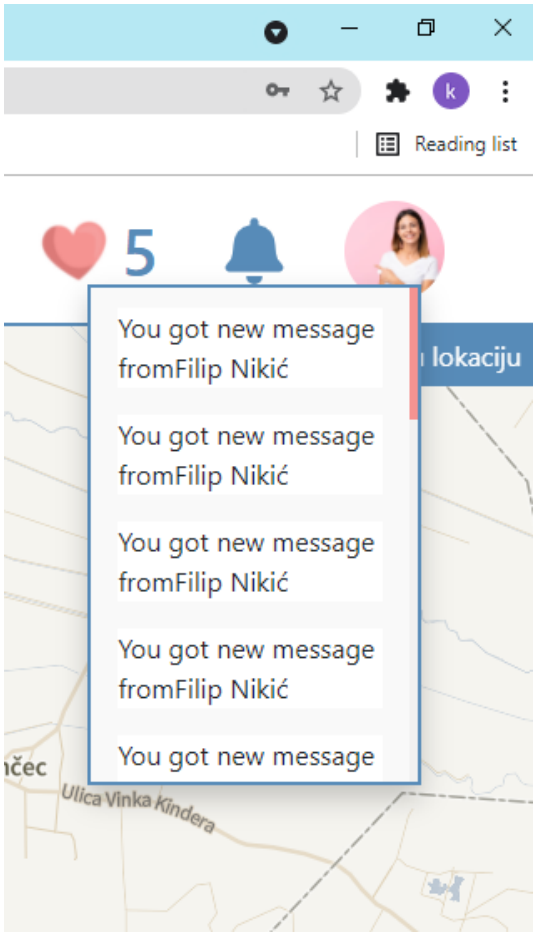
Slika broj 48 prikazuje korisničko sučelje za opcije sviđa mi se, ne sviđa mi se i pregled profila.

6.13.Notifikacijski sustav

Korisnik može dobiti dvije vrste notifikacija. Notifikaciju tko ga je označio sa „sviđa mi se“ i da je dobio poruku od određenog korisnika. Notifikacije se nalaze u zaglavlju aplikacije i dostupne su sa svake stranice aplikacije.

Notifikacijski sustav funkcionira kao i razgovor preko socket.io.client biblioteke na klijentu i funkcionira asinkrono tako da se nova notifikacija odmah šalje korisniku i pritom se na ponovnom učitavanju stranice sprema na bazu i preko nje učitava na klijenta.

Svaki puta kada pošaljemo novu poruku ili označimo korisnika sa „sviđa mi se“, kreira se nova notifikacija koja se šalje korisniku.



Slika 44. Prikaz notifikacija korisnika

```
useEffect(() => {
  socket.on("getNotification", async(data) => {

    try {
      const res = await axios.get(process.env.REACT_APP_GET_USER_DATA+"/"+data.senderId,{
      headers: {
        authorization: accessToken,
      }})


      const noficationText = `Dobili ste novu poruku od ${res.data.firstName} ${res.data.lastName}`
      const obj = data
      obj.text = noficationText
      setArrivalNotification(obj);
    } catch (error) {
      errorHandler(error)
    }


  });
}, []);
```

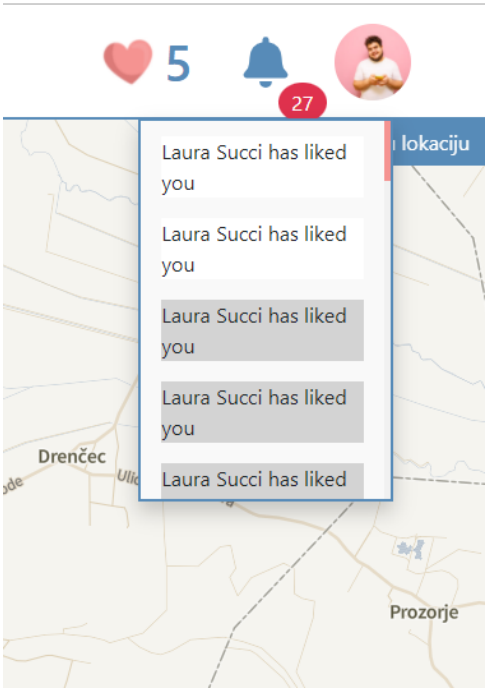
Slika 45. getNotification funkcija na socketu

Pri učitavanju Header komponente na slici broj 50, u useEffect() funkciji poziva se getNotification() funkcija koja dohvaća sve trenutne nove notifikacije za korisnika. Poziva se get zahtjev na poslužitelja (Kožić,2021.) i postavlja se nova nadolazeća notifikacija sa setArrivalNotification().

```
const setSeenNotification = async (n_id, type, senderId, chat_id = null) => {

  console.log( (n_id, type, senderId, chat_id) )

  const txt =
    type == "0"
      ? process.env.REACT_APP_GET_CHAT_NOTIFICATION
      : process.env.REACT_APP_GET_NOTIFICATION;

  const goToPage = (senderId) => {
    if (type == 0) {
      history.push(`/chat/${chat_id}`);
    } else if (type == 1) {
      history.push(`/user/${senderId}`);
    }
  };


  try {
    const res = await axios.get(txt + `${n_id}`, {
      headers: {
        authorization: accessToken,
      },
    });

    console.log(res.data);
    goToPage(senderId);
    setNumberofNotification(numberOfNotifications-1)

  } catch (error) {
    errorHandler(error);
  }
};
```

Slika 46. Prikaz setSeenNotification funkcije

Na slici broj 51 prikazana je funkcija setSeenNotification koja prati vidljivost notifikacije. Notifikacije koje korisnik nije vidio bit će označene blago sivom bojom, a one koje su viđene će se pobijeliti. Klikom na notifikaciju gasimo njenu vidljivost. Ovisno na koju smo notifikaciju kliknuli, aplikacija će nas preustrojiti na razgovro ili na profil korisnika.

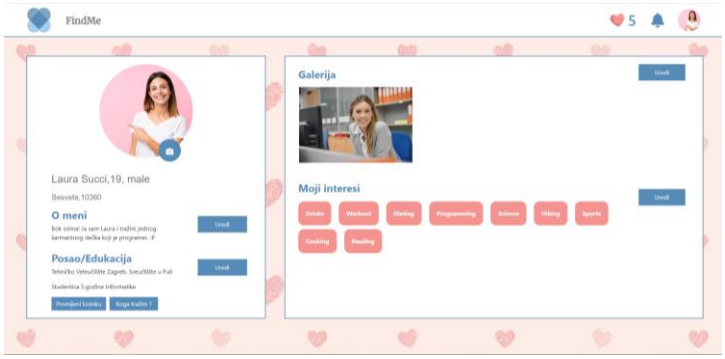


Slika 47. Prikaz neotvorenih notikacija i njihovog broja

6.14.Korisnički profili

6.14.1. Privatni profil

Korisnički profil sastoji se od profilne slike korisnika, njegovog opisa, posla, edukacije, galerije i interesa. Na profilu korisnik može izmijeniti svoje podatke.



Slika 48. Prikaz profila ulogiranog korisnika

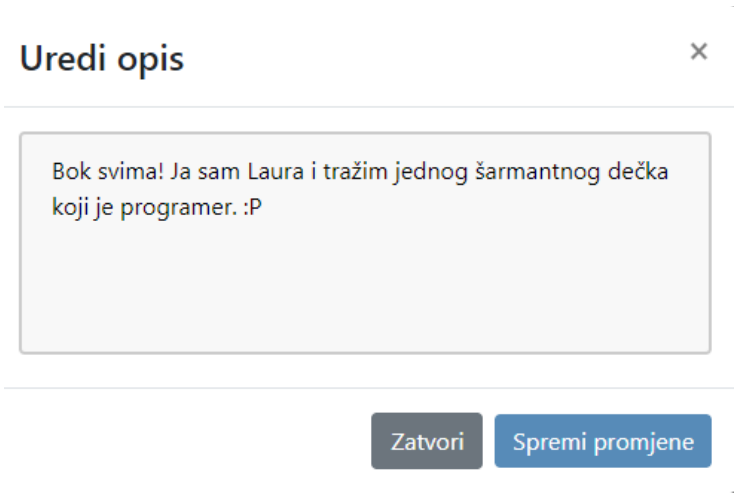
6.14.2. Javni profil

Javni profili su profili korisnika na kojeg kliknemo. Tamo su njegove informacije, može ga se označiti sa „sviđa mi se“, „ne sviđa mi se“ i započeti razgovor s njegovog profila.

Slika 49. Javni profil korisnika

6.15. Promjena osobnih podataka

Sastoji se od promjene profilne slike. Promjene opisa, seksualne orijentacije, lozinke, edukacije, posla, interesa i galerije.



Slika 50. Prikaz promjene opisa korisnika

6.16.Chat

Chat je još jedna od glavnih funkcionalnosti aplikacije. Sastoji se od otvorenih razgovora sa korisnicima i od samog razgovora.

Na razgovor stranici korisnik komunicira s korisnikom s kojim se isključivo morao povezati prije započinjanja razgovora. Korisnici mogu izmjeniti poruke i slike.

Za potrebe razgovora koristi se Socket IO server koji se sastoji od SocketIo server i SocketIo client biblioteke. Socket Io klijent prima eventove s poslužiteljskog Socket Io te emitira na isti nove eventove koje kasnije poslužiteljski dio obrađuje (Kožić, 2021.). Nakon što poslužitelj pošalje event s određenim podacima, klijent ovisno o tim eventovima odrađuje određene funkcionalnosti.

```
useEffect(() => {
  if (socket != null) {

    socket.on("getUsers", (users) => {});

    socket.on("getMessage", (data) => {
      let newMessageArray = Array.from(chatMessages);
      let messObj = {
        message: data.text,
        date: Date.now(),
        senderID: data.senderId,
        imageUrl : data.imageUrl
      };
      setArriveMessage(messObj);
    });
  }
}, [socket]);
```

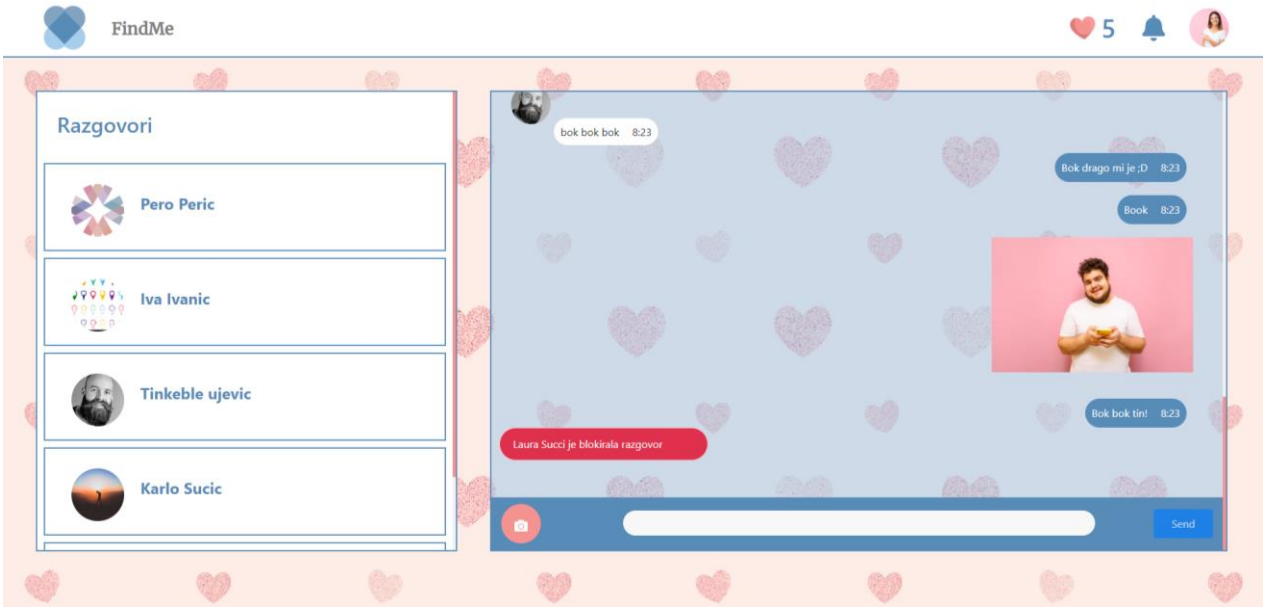
Slika 51. Primjer dohvaćanja poruka

U prikazanom primjeru na event „getMessage“ kao argument se zaprima data koja predstavlja podatke o poruci, te se naknadno korisniku prikazuju novo dospjele poruke.

Poruka se sastoji od teksta, datuma, ID-a od korisnika koji šalje poruku i ID-a od korisnika koji prima poruku.

6.16.1. Blokiranje razgovora sa korisnikom

Postoji također i mogućnost blokiranja razgovora sa korisnikom i odblokiranja kasnije. Kada se razgovor zablokira nije moguće nikako poslati poruku korisniku koji je blokirao razgovor.



Slika 52. Prikaz razgovora sa korisnikom i blokiran chat

U ovom primjeru korisnik je blokirao razgovor sa odabranim korisnikom. U tom slučaju poruke se ne mogu više slati sve dok korisnik ne odblokira korisnike preko razgovora.

7. Zaključak

Kvalitetna aplikacija je esencijalna za stvaranje dugotrajnih i značajnih odnosa između osoba, no taj zadatak nije uvijek lagan za postići. Većina korisnika još uvijek koristi ljubavne aplikacije striktno za veze bez obaveze tako da je potrebno promijeniti kulturu u društvu da bi se rezultati poboljšali. Čak pola populacije Amerikanaca tvrdi da su veze gdje su se partneri upoznali preko stranice za traženje ljubavi jednako uspješne kao i veze koje su započele uživo. Zbog tako velikog postotka uspješnosti na stranicama za stvaranje ljubavnih veza, ne smije se olako shvaćati online upoznavanje pogotovo u današnjem, modernom društvu. Promjenom trenda života i uključivanjem sve više tehnologije u našem društvu, interakcije preko interneta postale su jednako važne ako ne i važnije nego one uživo, te ih se stoga treba shvatiti ozbiljno i omogućiti korisnicima što jasniju, ugodniju i povjerljivu komunikaciju preko ove aplikacije. Važna misija svakog klijentskog dijela aplikacije je jasnoća misije i namjere aplikacije, što jednostavnije i razumljivije korisničko sučelje i očekivani rezultat korisnikovih želja.

Namjera ove aplikacije je natjerati korisnike da razmisle koga žele izabrati, a ne da se samo „prodaju“ na traci mora izbora. Aplikacija omogućuje lagano i detaljno pretraživanje korisnika koji su nam u blizini ili do 200 kilometara.

Dodatna nadogradnja ove aplikacije bi bila kažnjavanje korisnika za neodgovaranje na poruke, što bi dodatno natjeralo korisnike da budu kulturniji, jer u suprotnom ne bi mogli koristiti aplikaciju.

Nadalje, potrebno je uvesti neke osnove smjernice za uspostavu komunikacije kako bi se korisnicima olakšalo početno komuniciranje i stres pri upoznavanju nove osobe. Rečenice bi bile takozvani započinitelji razgovora (engl. ice breakers) koji lakše započinjanju komunikaciju sa odabranim korisnikom. Također, moguće je dodati i status veze s korisnikom tako da se motivira korisnike da skupljaju što više bodova i obogate svoju vezu s odabranim korisnikom.

Dodatno proširenje za stvaranje novih prijateljskih odnosa bi bilo dodavanje javnog zida (engl. feed) preko kojega bi korisnici izmjenjivali svoje statuse, grupne slike s tagovima, komentare i lokacije za sastajanje na kojima bi se mogli naći korisnici koji su voljni upoznati nove osobe.

8. Literatura

Ally, V. S. (16. 9 2021). *VIDA SELECT*. Dohvaćeno iz Badoo Review (2021): What You Should Know & If It's Worth Trying: https://www.vidaselect.com/badoo-review/, [19.rujna.2021]

BBC News. (7. 11 2013). Dohvaćeno iz #BBCtrending: How drugs are offered on Instagram: https://www.bbc.com/news/av/magazine-24849537, [19.rujna.2021]

Center, P. R. (6. Veljača 2020). *The Virtues and Downsides of Online Dating*. Dohvaćeno iz Pew Research Center: https://www.pewresearch.org/internet/2020/02/06/the-virtues-and-downsides-of-online-dating/, [19.rujna.2021]

Dietz, L. (2. 7 2019). *Medium*. Dohvaćeno iz Instagram Is Not a Dating Site: https://medium.com/swlh/instagram-is-not-a-dating-site-7bd93aa05828, [19.rujna.2021]

GCFGlobal . (16. 9 2021). Dohvaćeno iz What is Facebook?: https://edu.gcfglobal.org/en/facebook101/what-is-facebook/1/, [19.rujna.2021]

GitHub. (14. Studeni 2018). Dohvaćeno iz "Search · stars:>1": https://github.com/search?o=desc&q=stars%3A%3E1&s=stars&type=Repositories, [19.rujna.2021]

Good Therapy. (8. 2 2019). Dohvaćeno iz How Many Friends Does the Average Person Have?: https://www.goodtherapy.org/blog/psychology-facts/how-many-friends-does-average-person-have-0208197, [19.rujna.2021]

Instagram. (17. 3 2021). Dohvaćeno iz Continuing to Make Instagram Safer for the Youngest Members of Our Community: https://about.instagram.com/blog/announcements/continuing-to-make-instagram-safer-for-the-youngest-members-of-our-community, [19.rujna.2021]

Kelly, M. (3. 8 2010). *marie daire*. Dohvaćeno iz How Love Makes People Prettier: https://www.marieclaire.com/sex-love/a4178/how-love-makes-people-prettier/, [19.rujna.2021]

Makhija, R. M. (16. 9 2021). *Guru TechnoLabs*. Dohvaćeno iz Guru TechnoLabs: https://www.gurutechnolabs.com/tinder-feature-list/, [19.rujna.2021]

Miller, D. (2011). *Tales from Facebook.*

npm. (2021). Dohvaćeno iz axios: https://www.npmjs.com/package/axios, [19.rujna.2021]

Ortega, J., & Hergovich, P. (2017). The Strength of Absent Ties: Social Integration via Online Dating

Paska, I. (14. 2 2020). *hrčak*. Dohvaćeno iz Fast Choices and Emancipatory Spaces: Complex Reality of Online dating Apps: https://hrcak.srce.hr/238332, [19.rujna.2021]

Peard, L. (16. 9 2021). *NEAL SCHAFFER*. Dohvaćeno iz A List of the Latest Facebook Features: https://nealschaffer.com/facebook-features/, [19.rujna.2021]

React. (16. 9 2021). Dohvaćeno iz Introducing JSX: https://reactjs.org/docs/introducing-jsx.html, [19.rujna.2021]

React. (2021). Dohvaćeno iz React.Component: https://reactjs.org/docs/react-component.html, [19.rujna.2021]

REACT TRAINING / REACT ROUTER. (2021). Dohvaćeno iz https://reactrouter.com/web/guides/quick-start, [19.rujna.2021]

react-map-gl. (2021). Dohvaćeno iz REACT-MAP-GL: https://visgl.github.io/react-map-gl/, [19.rujna.2021]

Short, K. (9. 9 2021). *Best Company*. Dohvaćeno iz Best Company: https://bestcompany.com/online-dating/compare/tinder-vs-badoo#:~:text=Most%20of%20Badoo's%20new%20users,for%20being%20a%20hookup%20app., [19.rujna.2021]

webwise. (16. 9 2021). *webwise*. Dohvaćeno iz Explained: What is Instagram?: https://www.webwise.ie/parents/explained-image-sharing-app-instagram/, [19.rujna.2021]

Wieruch, R. (2021). *The Road to React: Your journey to master React.js in JavaScript (2021 Edition).*

Izvori slika:

1. https://datingappsfree.com/badoo-review/, (Badoo Dating Site Review 2020 – To Badoo or Not to do?, 16. 7. 2020.)
2. https://docs.mapbox.com/mapbox-gl-js/example/image-on-a-map/, (Mapbox, 19.9.2021.)
3. https://getbootstrap.com/docs/5.1/layout/grid/, (Bootstrap, 19.9.2021.)
4. https://louisem.com/180621/how-to-grow-your-instagram,(Louise Mayers,Visual Social Media, 17.8.2021.)
5. https://neilpatel.com/blog/what-should-you-put-in-each-section-of-your-facebook-page-for-maximum-engagement/, (Neil Patel, 19.9.2021.)
6. https://reactjs.org/docs/components-and-props.html, (React, 19.9.2021.)
7. https://reactjs.org/docs/introducing-jsx.html, (React, 19.9.2021.)
8. https://reactrouter.com/web/guides/quick-start, (REACT TRAINING / REACT ROUTER, 19.9.2021.)
9. https://tinder.com/, (Tinder, 19.9.2021.)
10. https://www.npmjs.com/package/axios, (npm, 19.9.2021.)

Diplomski rad:

1. Karlo Dini Kožić (Sveučilište Jurja Dobrile u Puli, Pula, 2021) Razvoj poslužiteljskih komponenti web aplikacije za stvaranje novih prijateljskih i ljubavnih veza

9. Popis slika

Slika 1. Prikaz početne stranice Tinder aplikacije (9) 5
Slika 2. Prikaz sučelja Badoo aplikacije (1) 7
Slika 3. Prikaz profila korisnika na Facebook stranici (5) 9
Slika 4. Prikaz Instagam profila korisnika (4) 11
Slika 5. Prikaz JSX sintakse u React-u (7) 13
Slika 6. Primjer klasne komponente (6) 15
Slika 7. Primjer slaganja kolona u redovima (3) 16
Slika 8. Unos tokena preko environment varijable 17
Slika 9. Prikaz kostimirane mape (2) 18
Slika 10 Instalacija paketa Axios preko yarn 18
Slika 11. Prikaz POST zahtjeva pomoću Axiosa (10) 19
Slika 12. Primjer Router komponente u App komponenti (8) 20
Slika 13. Prikaz dijagrama slučajeva korištenja aplikacije FindMe 21
Slika 14. Prikaz početnog dijela uvodne stranice 22
Slika 15. Prikaz glavne stranice aplikacije 23
Slika 16. Korisnikova stranica 24
Slika 17. Stranica za razgovor 25
Slika 18. Prikaz LoginSheme 27
Slika 19. Prikaz Formik inicijalizacije 28
Slika 20. Prikaz login forme 29
Slika 21. Prikaz login korisnika 29
Slika 22. Slanje emaila korisniku preko post zahtjeva 30
Slika 23. Forgot Password komponenta 31
Slika 24. Primjer poslane pošte sa novom lozinkom 31
Slika 25. Typehead komponenta za automatsko ispunjavanje poštanskog broja 32
Slika 26. RegisterShema za potrebe validacije registracije 32
Slika 27. onSubmit funkcija za registraciju 33
Slika 28. Prikaz registracije 33
Slika 29. Postavljanje access tokena 36
Slika 30. Dohvaćanje podataka od korisnika pomoću access tokena 36

Slika 31. Prikaz ReactMapGL komponente 37
Slika 32. Dohvat korisnikove lokacije 39
Slika 33. Prikaz filtera za korisnike 40
Slika 34. Slanje filtera na poslužitelja 42
Slika 35. Prikaz najboljih trenutnih rezultata za korisnika 43
Slika 36. Prikaz života Slika 37. Prikaz punjenja života 43
Slika 38. Prikaz Timer komponente 44
Slika 39. Prikaz Timer render dijela 45
Slika 40. Prikaz funkcije fetchHearts 46
Slika 41. Prikaz likeUser funkcije 47
Slika 42. Ovisnost gumba o lajku korisnika 47
Slika 43. Prikaz kartice korisnika sa opcijama lajk,dislajk i pregled profila 49
Slika 44. Prikaz notifikacija korisnika 50
Slika 45. getNotification funkcija na socketu 50
Slika 46. Prikaz setSeenNotification funkcije 51
Slika 47. Prikaz neotvorenih notikacija i njihovog broja 52
Slika 48. Prikaz profila ulogiranog korisnika 52
Slika 49. Javni profil korisnika 53
Slika 50. Prikaz promjene opisa korisnika 53
Slika 51. Primjer dohvaćanja poruka 54
Slika 52. Prikaz razgovora sa korisnikom i blokiran chat 55

Sažetak

Od osobnih reklama koje su se počele prikazivati u publikacijama oko 1700te godine do videokazeta koje prodaju usluge za „dating“ koje su počele prije par desetljeća, platforme koje se koriste za traženje ljubavnih i prijateljskih odnosa su se razvile tijekom povijesti. Evolucija je nastavila s rastom online stranica za traženje ljubavnih i prijateljskih odnosa i mobilnih aplikacija za tu namjenu. Ovaj rad bavi se razvojem klijentskih komponenti za aplikaciju koja je namijenjena za stvaranje ljubavnih i prijateljskih odnosa. Aplikacija se sastoji od brojnih funkcionalnosti od kojih su neke: prikaz korisnika na mapi i njihovo filtriranje po interesima,spolu,godinama itd., sustav života kojim se životi (srca) gube svakim novim „lajkom“ korisnika, „dislakom“ korisnika i započinjanjem razgovora sa korisnikom. Ulogirani korisnik ima svoj profil na kojemu dodaje svoj osobni opis, edukaciju i karijeru. Potom ima galeriju svojih slika, te popis svojih interesa po kategorijama. Sve to mu omogućuje da pronađe što bolju polovicu i dugotrajne odnose. Aplikacija nastoji svojim funkcionalnostima natjerati korisnika da što pametnije izabere osobu koja mu odgovara, te da se iluzijom manjeg izbora ne osjeća da ne može izabrati.

Ključne riječi : Javascript,React,Axios, web aplikacija, klijentske komponente, online dating, odnosi, prijateljstvo

Abstract

From personal ads that began appearing in publications around the 1700s to videocassette dating services that sprang up decades ago, the platforms people use to seek out romantic partners have evolved throughout history. This evolution has continued with the rise of online dating sites and mobile apps. This work is based on development od client side components of dating and friendship web app. The app is consisted of many functionalities of which are some: displayed users on map and filtering those users on their interests, sex, age etc., system of lifes on which user loses a heart when he likes, dislikes user or starts a new conversation with him. Logged in user has a profile on which he can edit his personal info, education and job. Then he

has his own gallery photos and list of his interests by category. All of this is helping the user find his better half and create a meaningful relationship. The created web app with its aforementioned functionalities tries to persuade the user to choose the person smartly and help him choose the correct person with the illusion of less choice.

Keywords: Javascript, React, Axios, web application, fronted, dating, online relationships, friendship