Mobilna Android aplikacija za višekriterijsko preporučivanje turističkih odredišta

Gerić, Klara-Iva

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

MOBILNA ANDROID APLIKACIJA ZA VIŠEKRITERIJSKO PREPORUČIVANJE TURISTIČKIH ODREDIŠTA

Završni rad

Klara-Iva Gerić

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 28.08.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Klara- Iva Gerić
Studij, smjer:	Računalno inženjerstvo
Mat. br. Pristupnika, godina upisa:	R 4344, 12.10.2021.
OIB Pristupnika:	06370540936
Mentor:	prof. dr. sc. Goran Martinović
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna Android aplikacija za višekriterijsko preporučivanje turističkih odredišta
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rad:	Uzimajući u obzir postojeća slična rješenja i pristupe, u završnom radu treba analizirati i opisati izazove i mogućnosti preporučivanja turističkih odredišta. Uzimajući u obzir sklonosti korisnika i postojeća turistička odredišta nekog područja, mogućnosti stvaranja preporuka temeljenih na sadržaju, te kolaborativnom filtriranju i rangiranju, treba definirati funkcionalne i nefunkcionalne zahtjeve, kao i predložiti model i
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	28.08.2023.
Datum potvrde ocjene od strane Odbora:	08.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 08.09.2023.

Ime i prezime studenta:	Klara- Iva Gerić
Studij:	Računalno inženjerstvo
Mat. br. studenta, godina upisa:	R 4344, 12.10.2021.
Turnitin podudaranje [%]:	12

Ovom izjavom izjavljujem da je rad pod nazivom**: Mobilna Android aplikacija za višekriterijsko preporučivanje turističkih odredišta**

izrađen pod vodstvom mentora prof. dr. sc. Goran Martinović

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1 Zadatak završnog rada 1

2. PROBLEMI, IZAZOVI I STANJE U PODRUČJU PREPORUČIVANJA TURISTIČKIH ODREDIŠTA 3

2.1 Pojam turizma 3

2.2 Izazovi preporuka turističkih odredišta 3

2.3 Stanje u području i korišteni postupci preporučivanja 4

2.4 Postojeća rješenja za preporuke turističkih odredišta 4

2.4.1 Sygic Travel Maps Offline & Trip Planner 5

2.4.2 Vienna Travel Guide 5

2.4.3 Tripadvisor: Plan & Book Trips 6

3. MODEL APLIKACIJE ZA VIŠEKRITERIJSKO PREPORUČIVANJE TURISTIČKIH ODREDIŠTA 8

3.1 Funkcionalni zahtjevi na mobilnu aplikaciju 8

3.1.1 Prijava korisnika 8

3.1.2 Registracija korisnika 8

3.1.3 Preporučivanje turističkih odredišta 9

3.1.4 Google Maps Karta 9

3.1.5 Detalji turističkog odredišta 9

3.1.6 Izmjena postavki računa 9

3.1.7 Odjava korisnika 9

3.2 Nefunkcionalni zahtjevi mobilne aplikacije 10

3.2.1 Performanse 10

3.2.2 Sigurnost 10

3.2.3 Pristupačnost 10

3.2.4 Kompatibilnost 11

3.3 Slijed korištenja mobilne aplikacije 11

3.4 Sustav preporučivanja 13

4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA VIŠEKRITERIJSKO PREPORUČIVANJE TURISTIČKIH ODREDIŠTA 16

4.1 Korištene tehnologije i alati 16

4.1.1 Canva 16

4.1.2 Android Studio 18

4.1.3 Kotlin 18

4.1.4 XML 18

4.1.5 Firebase 18

4.2 Programsko rješenje na strani korisnika 18

4.2.1 Registracija i prijava korisnika 18

4.2.2 Google Maps i prikaz turističkih odredišta na karti 20

4.2.3 Odabir preferiranih kategorija 21

4.2.4 Preporučena odredišta i odabir filtara preporuke 23

4.2.5 Detalji turističkog odredišta 26

4.2.6 Profil korisnika 27

4.3 Programska rješenja na strani poslužitelja 27

4.3.1 Baza podataka 27

4.3.2 Autentifikacija korisnika 29

4.3.3 Google Maps karta 29

5. NAČIN KORIŠTENJA, ISPITIVANJE MOBILNE APLIKACIJE I ANALIZA 31

5.1 Korištenje ostvarene mobilne Android aplikacije 31

5.2 Ispitivanje mobilne aplikacije 36

5.2.1 Korisnički slučaj – Nema odabranih kategorija 36

5.2.2 Korisnički slučaj – Jedna ili više odabranih kategorija 37

5.2.3 Korisnički slučaj – Utjecaj korisničkih ocjena na rangiranje odredišta 39

5.2.4 Korisnički slučaj – Odabir filtra 39

5.3 Analiza rezultata ispitivanja 40

6. ZAKLJUČAK 41

LITERATURA 42

POPIS SLIKA 44

SAŽETAK 46

ABSTRACT 47

ŽIVOTOPIS 48

PRILOZI 49

1. UVOD

Turizam je postao neizostavan dio ljudskog života. Ako čovjek ne putuje sam, onda vrlo često čuje od drugih ljudi koja su mjesta posjetili ili pomoću različitih medija saznaje o atraktivnostima i razvoju turizma drugih zemalja. Kada osoba odluči posjetiti neku novu lokaciju, najčešće ne zna odakle krenuti istraživati, ali se taj problem može lako riješiti pomoću mobilnih uređaja. Kao suputnik, mobilna aplikacija može korisniku otkriti najbolja turistička odredišta koja možda sam ne bi pronašao na druge načine.

U ovom završnom radu problem velikog broja turističkih sadržaja rješava se višekriterijskim preporučivanjem. Cilj je korisnicima olakšati izbor turističkih odredišta na temelju njihovih sklonosti. Ostvarena mobilna aplikacija zahtijeva registraciju korisnika, zatim traži da korisnik odabere kategorije koje su mu od interesa te mu na temelju tih odabira nudi personalizirani popis turističkih odredišta. Popis se dodatno personalizira izborom filtara i tako rangirani popis daje korisniku uvid u najbolja turistička odredišta za njegove interese. Rangiranje odredišta provodi se na temelju prosječnih ocjena drugih korisnika koji su već ocijenili određeno odredište. Također, omogućen je i pregled odredišta na interaktivnoj karti. Aplikacija je izrađena koristeći razvojno okruženje Android Studio, uz upotrebu programskog jezika Kotlin, a za pohranu podataka korišten je Firebase.

Kroz drugo poglavlje opisani su problemi i izazovi koji se pojavljuju u preporučivanju turističkih odredišta, opisano je i stanje u području te su navedena tri slična postojeća rješenja. Treće poglavlje bavi se funkcionalnim i nefunkcionalnim zahtjevima na mobilnu aplikaciju te definiranjem građe aplikacije, dok četvrto poglavlje prikazuje programsko rješenje mobilne aplikacije prema postavljenim zahtjevima. Peto poglavlje prikazuje upute i način rada aplikacije kao i ispitivanje s analizom rada aplikacije.

1.1 Zadatak završnog rada

Uzimajući u obzir postojeća slična rješenja i pristupe, u završnom radu treba analizirati i opisati izazove i mogućnosti preporučivanja turističkih odredišta. Uzimajući u obzir sklonosti korisnika i postojeća turistička odredišta nekog područja, mogućnosti stvaranja preporuka temeljenih na sadržaju, te kolaborativnom filtriranju i rangiranju, treba definirati funkcionalne i nefunkcionalne zahtjeve, kao i predložiti model i arhitekturu mobilne Android aplikacije. Mobilna aplikacija treba omogućiti prijavu i stvaranje profila s interesima korisnika, pregled i izbor multimedijski

prikazanih turističkih odredišta, ocjenjivanje i rangiranje, te višekriterijsko preporučivanje odredišta. U praktičnom dijelu rada, koristeći predloženi model i arhitekturu, te prikladne programske jezike i tehnologije (Kotlin, Firebase i druge po potrebi) treba programski ostvariti mobilnu aplikaciju s bazom podataka i sustavom stvaranja preporuka. Mobilnu aplikaciju potrebno je ispitati i analizirati za različite slučajeve korištenja na primjerima odredišta grada Virovitice i okoline.

2. PROBLEMI, IZAZOVI I STANJE U PODRUČJU PREPORUČIVANJA TURISTIČKIH ODREDIŠTA

Razvojem pametnih mobilnih uređaja i razvojem prometne mreže, ljudi dobivaju želju za istraživanjem novih i nepoznatih turističkih odredišta. Samim time dolazi i do rasta jedne od ugostiteljskih grana, turizma. Nova odredišta za koje se turisti odluče posjetiti imaju mnogo sadržaja i zanimljivosti koje mogu obići i razgledati, ali im najveći problem stvara nedostatak vremena. Razvojem ove aplikacije korisnicima se pruža mogućnost vrlo brzog pronalaska turističkog odredišta koje bi za njih osobno bilo zanimljivo, a u tom izboru im pomažu i drugi korisnici dodjeljivanjem ocjena turističkim odredištima.

2.1 Pojam turizma

Hunziker i Krapft [1] su definirali turizam kao skup odnosa i pojava, koje proizlaze iz putovanja i boravka posjetitelja nekog mjesta, ako se tim boravkom ne zasniva stalno prebivalište i ako s tim boravkom nije povezana nikakva njihova gospodarska djelatnost. Također, prema [2], turizam uključuje sve aktivnosti proizašle iz putovanja i boravaka osoba izvan njihove uobičajene sredine ne dulje od jedne godine radi odmora, poslovnog putovanja i drugih razloga nevezanih uz aktivnosti za koje bi primili ikakvu naknadu u mjestu koje posjećuju. Može se reći kako je turizam jedna od gospodarskih grana koja najbrže raste i ima velik utjecaj na prihod i razvitak zemlje. Turizam također otvara vrlo velik broj radnih mjesta, stoga je većini zemalja u velikom interesu privući što više turista.

2.2 Izazovi preporuka turističkih odredišta

Vrlo mali broj ljudi u današnje vrijeme odlazi u nepoznat grad bez da su prethodno istražili što se u njemu može posjetiti. Za pronalazak dobrih odredišta često koriste društvene mreže, poznate web stranice za turističke preporuke ili čak preporuke poznanika. Međutim, veliki broj turista i dalje koristi tradicionalne izvore za vrijeme putovanja, kao što su to papirnate karte i knjige s vodičima. S tim ne dobivaju individualno iskustvo koje bi odgovaralo njihovim interesima. Vodiči su osmišljeni za prosječnog turista pa je rezultat toga velika količina informacija koja ponekad može dovesti do tzv. preopterećenja informacijama [3]. Vrlo je teško prikupiti i analizirati mnogo različitih mišljenja za samo jedno turističko odredište bez moderne tehnologije, a za više njih je gotovo nemoguće procijeniti je li odredište dovoljno dobro za osobni ukus turista. Većina turista koji posjećuju neki grad ili će se odlučiti na turističko vodstvo ili na samostalno razgledavanje uz

pomoć karte. Problem je da obje grupe turista gotovo sigurno ne dobivaju iskustvo koje odgovara njihovim interesima jer su prisiljeni pridružiti se grupnom iskustvu ili koristiti kartu koja je namijenjena svim turistima [4]. Pristupi u kojima korisnik sam prikuplja bitne informacije zahtijevaju veliko vremensko ulaganje, a može se dogoditi da takvih informacija u nekim gradovima bude i previše te mogu vrlo brzo zastarjeti i postati nerelevantne. Na temelju toga dolazi do preporučivanja loših i nezanimljivih turističkih odredišta. Svoje negativno iskustvo turist može prenijeti na druge turiste koji taj grad nisu posjetili i ako se ništa ne poduzme, može doći do lošeg razvijanja turizma toga grada. Zato, svaki turist želi dobiti personalizirane preporuke i tako najbolje iskoristiti svoje vrijeme za obilazak nekog mjesta.

2.3 Stanje u području i korišteni postupci preporučivanja

Za preporučivanje turističkih odredišta postoji mnogo stranica kao i aplikacija za mobilne uređaje ili računala. Te aplikacije koriste kontekstualni sustav preporuka. Ideja je da sustav može detektirati i reagirati na korisnikovu situaciju, kao na primjer prilagoditi korisničko sučelje ovisno o različitim iskustvima posjetitelja ili na temelju prethodnih odabira naučiti što bi moglo korisnika kasnije zanimati [5]. U zadnje vrijeme popularizirali su se razgovorni roboti (*engl. chatbot*), a primjer je [5] gdje je kreiran razgovorni robot koji koristi obradu prirodnog jezika i strojno učenje kako bi korisniku dao relevantne informacije za njegovo trenutno stanje.

Koristi se i kolaborativno filtriranje, ovisno o korisniku ili o predmetu koji se preporučuje. U [6] je korišteno kolaborativno filtriranje temeljeno na korisniku, tako da je korisniku ponuđena kratka anketa. Pomoću nje su prikupljeni podaci o korisniku, a zatim i o nekoliko turističkih odredišta koje je morao ocijeniti prema privlačnosti na temelju opisa bez posjećivanja tog odredišta. Strojnim učenjem su mu preporučena odredišta koja još nije vidio ni ocijenio, a svidjela su se grupi ljudi koja ima slične sklonosti kao taj korisnik.

Općenito, prije ili tijekom putovanja, sustavi za preporučivanje turističkih odredišta zahtijevaju unos od turista (implicitno, eksplicitno ili oboje) kako bi kreirali profil korisnika i izračunali preporučene rezultate koje bi zatim poslali nazad korisniku [7]. Sustavi, koji ne primaju unos od korisnika pa čak ni njegovu lokaciju, su previše općeniti i zapravo ne pomažu korisniku.

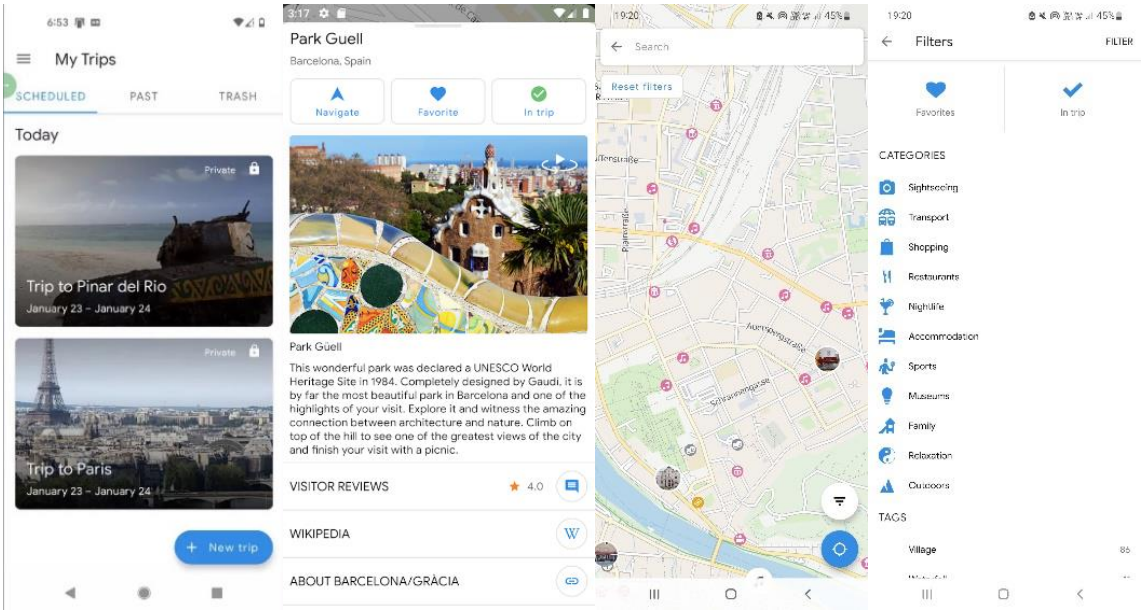
2.4 Postojeća rješenja za preporuke turističkih odredišta

Na Google Play platformi koja je najčešće korištena za preuzimanje aplikacija za Android mobilne uređaje postoje mnogobrojne aplikacije koje su povezane s turizmom, no međutim veliki broj tih

aplikacija služi za pronalazak hotela ili za planiranje putovanja. Prikazano je nekoliko aplikacija kojima je svrha preporuka turističkih odredišta.

2.4.1 Sygic Travel Maps Offline & Trip Planner

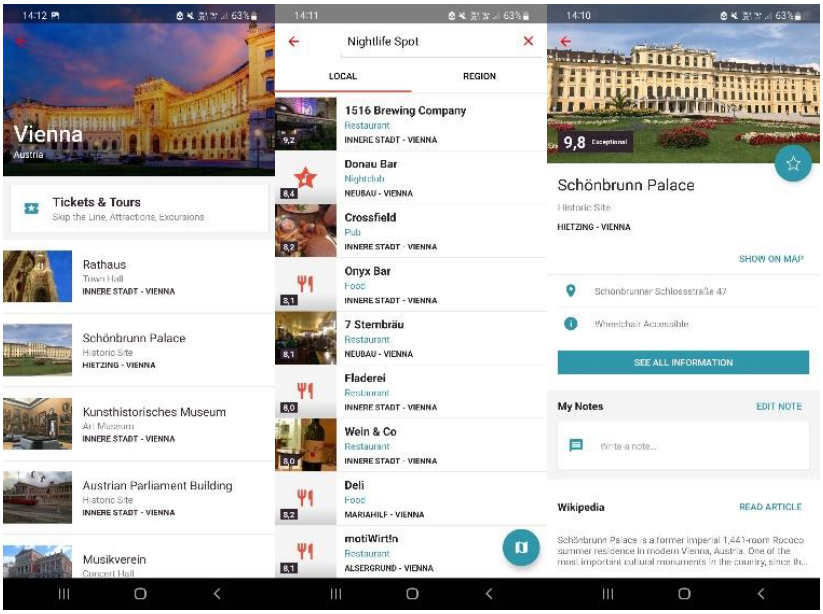
Ova aplikacija korisniku nudi mogućnost planiranja putovanja, ali i pregledavanja odredišta koje oni smatraju zanimljivim pomoću opcije sortiranja. Moguće je odabrati samo jednu kategoriju interesa i zatim će korisnik samo na karti imati mogućnost pregleda koja odredišta pripadaju toj kategoriji. Odabirom nekog odredišta, korisniku se prikazuje više detalja o odabranom odredištu, ponuđena mu je navigacija do odredišta te ga može dodati u omiljene lokacije ili u svoj plan putovanja. Slika 2.1 prikazuje sučelje ove aplikacije.



Slika 2.1 Prikaz sučelja aplikacije Sygic Travel Maps Offline & Trip Planner

2.4.2 Vienna Travel Guide

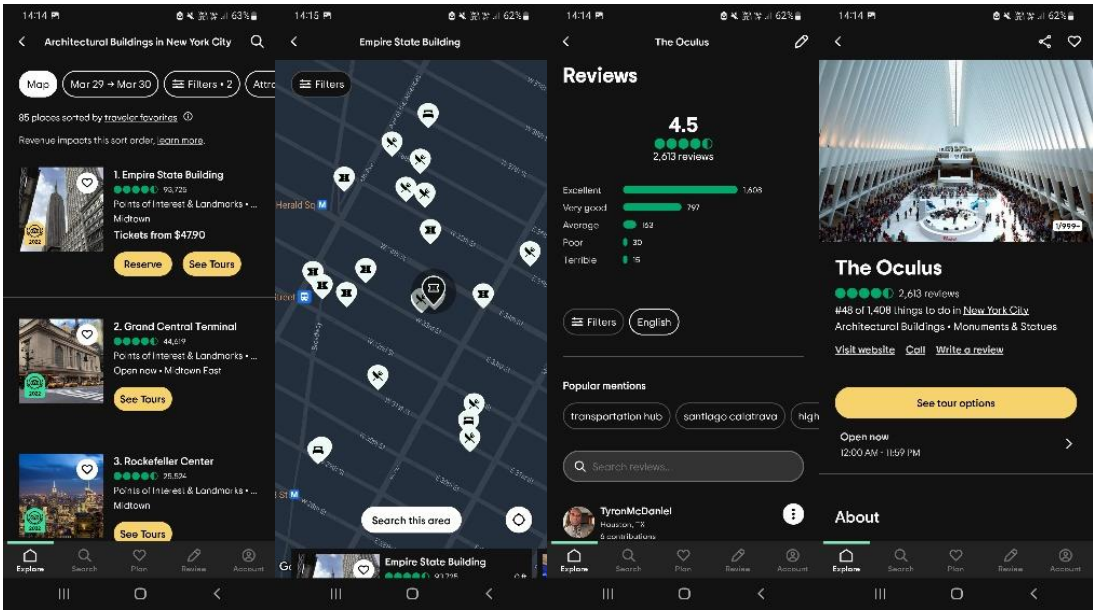
Aplikacija se može koristiti samo za grad Beč, ali ima mogućnost biranja kategorije koja korisnika zanima te mu na temelju toga daje popis turističkih odredišta. Moguće je preporučena odredišta prikazati na mapi. Nadalje, svako turističko odredište je ocijenjeno, ali nije jasno kako je ocjena izračunata. Pretpostavka je da su ocjene preuzete od Google Maps recenzija. Uz kratki opis, postoji i poveznica putem koje se može saznati više informacija na određenoj web stranici te se nude i ponude smještaja koji se nalaze u blizini odabranog odredišta. Također, u aplikaciji postoji i rangirana lista s turističkim odredištima koja imaju najvišu ocjenu neovisno o kategoriji u koju su uvrštena. Sučelje ove aplikacije prikazano je na slici 2.2.



Slika 2.2 Prikaz sučelja aplikacije Vienna Travel Guide

2.4.3 Tripadvisor: Plan & Book Trips

Aplikacija čije sučelje je prikazano na slici 2.3, nudi korisniku odabir grada kojeg želi posjetiti te datum putovanja i jednu od kategorija turističkih odredišta koju želi posjetiti. Na temelju jedne kategorije generiraju se preporuke koje korisnik može posjetiti, a rangiraju se prema ocjeni drugih korisnika ili prema broju korisnika koji su dodali to odredište u svoja omiljena odredišta. Aplikacija nudi i pregled odredišta te osnovne informacije o njima, ali i informacije o vodičima koji bi bili dostupni u vrijeme dolaska korisnika. Također, prikazana je lista dostupnih rezervacija hotela kao i poveznica na web stranicu gdje je moguća daljnja rezervacija. Predloženi su i restorani koji se nalaze u blizini tog hotela, ali se hoteli ne preporučuju ovisno o turističkim odredištima koje korisnika zanimaju.



Slika 2.3 Prikaz sučelja aplikacije Tripadvisor: Plan & Book Trips

3. MODEL APLIKACIJE ZA VIŠEKRITERIJSKO PREPORUČIVANJE TURISTIČKIH ODREDIŠTA

U ovom poglavlju se definiraju funkcionalni i nefunkcionalni zahtjevi na mobilnu aplikaciju i opis modela aplikacije za višekriterijsko preporučivanje turističkih odredišta.

3.1 Funkcionalni zahtjevi na mobilnu aplikaciju

Funkcionalni zahtjevi su izjave o uslugama koje bi sustav trebao pružati, kako bi sustav trebao reagirati na određene ulaze i kako bi se sustav trebao ponašati u određenim situacijama. U nekim slučajevima, funkcionalni zahtjevi mogu eksplicitno navesti što sustav ne bi trebao raditi [8]. Funkcionalni zahtjevi često ovise o tome kakva se aplikacija ili usluga razvija te ovise i o programskom okruženju u kojem se razvija, a bitno je i za koju vrstu korisnika je namijenjena. Oni također definiraju što bi sustav trebao raditi u kratkim natuknicama i često su pisani jednostavnim jezikom.

3.1.1 Prijava korisnika

Otvaranjem aplikacije korisniku se prikazuje zaslon za prijavu s poljima za unos adrese e-pošte i lozinke, tekstualni gumb za zaboravljenu lozinku, gumb za prijavu te gumb koji vodi korisnika na zaslon za registraciju. Ako je korisnik unio registriranu adresu e-pošte s pripadajućom lozinkom uspješno će se prijaviti te će mu se otvoriti početni zaslon aplikacije. U slučaju zaboravljene lozinke, korisnik može poslati automatski generiranu e-mail poruku za ponovno postavljanje lozinke na svoju adresu e-pošte, samo ako ta adresa e-pošte postoji u bazi korisnika. Nakon uspješne prijave korisnik ostaje prijavljen u aplikaciji čak i ako ju u potpunosti zatvori.

3.1.2 Registracija korisnika

Odabirom gumba za registraciju, otvara se zaslon za registriranje u kojem su polja za unos imena, adrese e-pošte i lozinke. Ako su svi parametri popunjeni i važeći, korisnik će biti uspješno registriran prilikom pritiska gumba za registraciju. Firebase Authentication koristi adrese e-pošte kao jedinstvene oznake, stoga može postojati samo jedan račun s jednom adresom e-pošte. Na zaslonu se također nalazi i gumb za prijavu korisnika, u slučaju da korisnik ipak ima korisnički račun te pritiskom na njega mu se otvara ponovno zaslon za prijavu.

3.1.3 Preporučivanje turističkih odredišta

Nakon uspješne prijave novog korisnika, prvo se otvara zaslon s mogućnostima odabira kategorija turističkih odredišta koje ga zanimaju. Kategorije turističkih odredišta su zapravo oznake koje se mogu dodijeliti svakom odredištu kako bi ga pobliže opisale. Bitno je za naglasiti da je odabir kategorije moguće izmijeniti u bilo kojem trenutku. Na temelju korisnikova odabira kategorija, kreirati će se popis preporučenih turističkih odredišta i na njega će odmah biti primijenjen prvi filtar sortiranja. Rangirani popis će biti prikazan u kartici „Rang“ koja je treća kartica na donjoj navigacijskoj traci zaslona aplikacije. Predložena turistička odredišta se prema volji korisnika mogu sortirati i prema drugim filtrima kao na primjer filtar zanimljivosti ili fotogeničnosti nekog turističkog odredišta. Turistička odredišta se rangiraju od najbolje do najlošije ocjene.

3.1.4 Google Maps Karta

Na zaslonu sa kartom su pomoću markera prikazana su sva odredišta koja se nalaze u bazi podataka turističkih odredišta. Odabirom nekog markera korisniku se otvara zaslon s detaljima odabranog turističkog odredišta. Karta je interaktivna, što znači da ju korisnik može zumirati, zakrenuti i slično. Cilj je da korisnik može vidjeti na karti gdje se nalaze sva turistička odredišta.

3.1.5 Detalji turističkog odredišta

Zaslon na kojem se može pronaći više informacija o odabranom turističkom odredištu kao što su to naziv, opis i multimedijski sadržaj. Na istom zaslonu se također nalaze ljestvice za ocjenjivanje svakog turističkog odredišta. Ocjenjivanje je u rangu od 1.0 do 5.0 s korakom od 0.5, a kriteriji po kojima se ocjenjuje su zapravo filtri za rangiranje odredišta s popisa preporučenih turističkih odredišta.

3.1.6 Izmjena postavki računa

Korisnik može poslati zahtjev za promjenu lozinke, a time će mu na adresu e-pošte biti poslan e-mail s uputama kako to učiniti. Na istom zaslonu može se vidjeti adresa e-pošte s kojom je korisnik prijavljen te postoji i mogućnost jednosmjernog slanja poruke korisničkoj podršci u slučaju pitanja ili problema s aplikacijom.

3.1.7 Odjava korisnika

Korisnik se može odjaviti na zaslonu gdje se nalaze podatci o njegovom profilu, a svi njegovi podaci ostaju spremljeni u bazi. Prijavom bilo kada i sa bilo kojeg drugog ili istog uređaja korisnik ima pristup svim svojim informacijama.

3.2 Nefunkcionalni zahtjevi mobilne aplikacije

Prema [8], nefunkcionalni zahtjevi su ograničenja na usluge ili funkcije koje nudi sustav. Oni uključuju vremenska ograničenja, ograničenja na razvojni proces i ograničenja nametnuta standardima. Nefunkcionalni zahtjevi se često odnose na cijeli sustav, a ne na pojedinačne značajke ili usluge sustava. Korisnicima je lakše pronaći način kako koristiti aplikaciju, iako neki od funkcionalnih zahtjeva nije dobro implementiran, ali ako jedan od nefunkcionalnih zahtjeva nije dobro implementiran to znači da postoji mogućnost da je cijeli sustav nesiguran za korištenje.

3.2.1 Performanse

Performanse mobilne aplikacije odnose se na različite zahtjeve, kao što su vrijeme pokretanja mobilne aplikacije i vrijeme potrebno za njeno potpuno učitavanje. Aplikacija treba efikasno upravljati dohvaćanjem podataka, što znači da aplikacija ne treba učitati sve podatke odmah, već tek kada su potrebni. Performanse također podrazumijevaju da aplikacija treba imati odzivno korisničko sučelje bez kašnjenja tijekom interakcije s korisnikom, te stabilnost aplikacije bez rušenja i sposobnost oporavka od neočekivanih situacija i pogrešaka.

3.2.2 Sigurnost

Aplikacija treba osigurati zaštitu od neželjenih vanjskih utjecaja, kao što su potencijalni hakerski napadi. Konkretno, aplikacija treba osigurati da haker koji pokušava pristupiti korisničkom profilu ne može saznati ima li stvarno korisnik račun samo s unesenom adresom e-pošte te da li je unesena lozinka pogrešna za unesenu adresu e-pošte. Ovim se postupkom informacije o korisničkim računima čuvaju u tajnosti, a samo stvarni vlasnik računa ima ovlasti i znanje o tim podacima. Također, aplikacija treba osigurati da programeri nemaju ovlasti za promjenu lozinke korisniku, već da jedino korisnik sam ima pravo izmijeniti svoju lozinku.

3.2.3 Pristupačnost

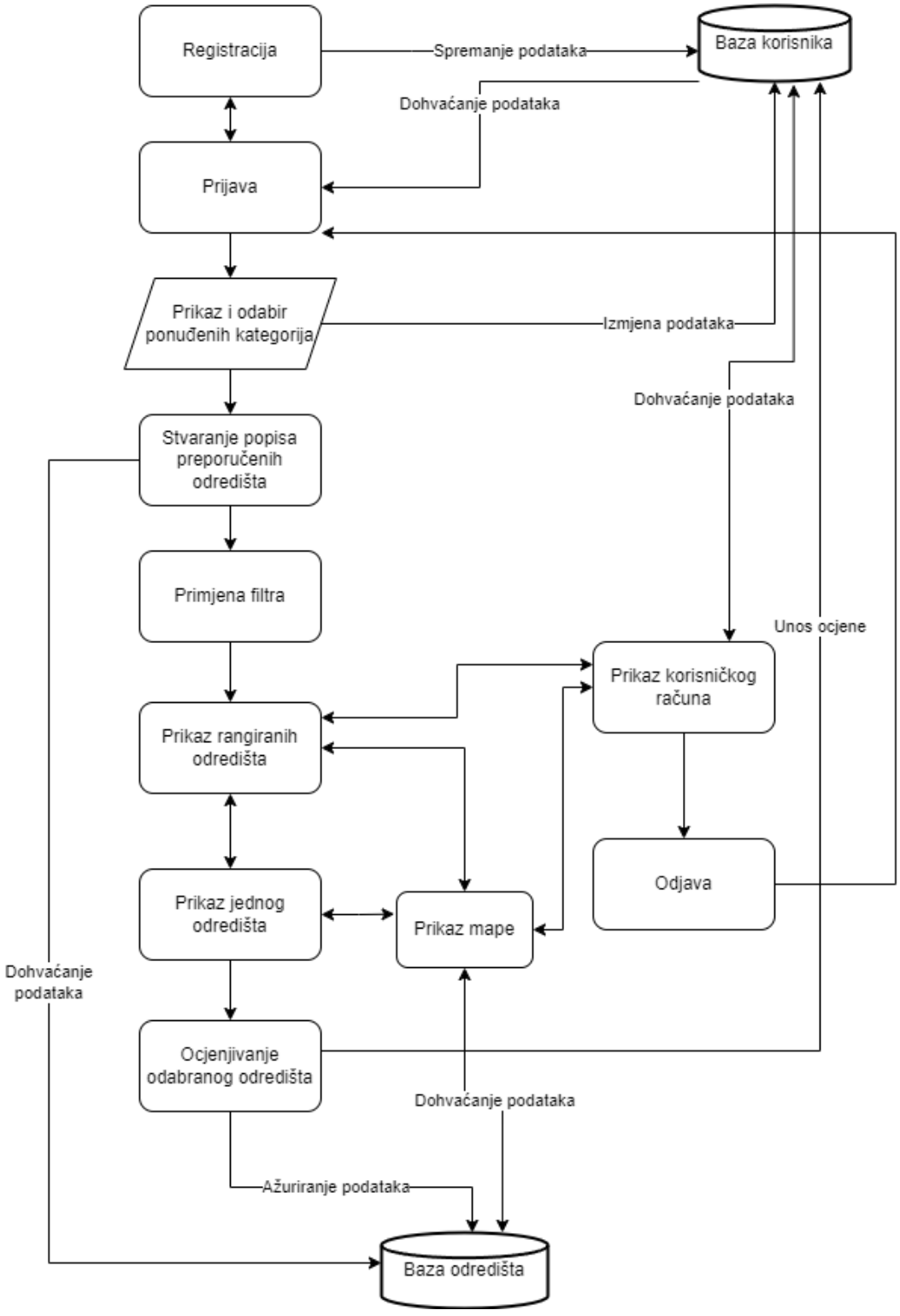
Aplikacija mora biti prilagođena različitim skupinama korisnika, to se odnosi i na skupinu osoba s invaliditetom. Potreban je dovoljno veliki kontrast boja i lako čitljivi fontovi. Pristupačnost se također odnosi i na prilagođavanje aplikacije promjeni teme sustava, iz svijetle u tamnu temu i obrnuto, a odnosi se i na zakretanje ekrana i aplikacija mora ispunjavati sve navedene uvjete.

3.2.4 Kompatibilnost

Aplikacija mora biti u mogućnosti raditi na različitim Android uređajima. Oni mogu biti različitih specifikacija, sklopovskih karakteristika, inačica operacijskih sustava i veličine zaslona. Minimalna zahtijevana inačica alata za razvoj programa (SDK) na kojoj aplikacija treba biti kompatibilna je 25, poznata pod nazivom „Nougat“, dok se preporučuje korištenje verzije 33, poznate pod nazivom „Tiramisu“.

3.3 Slijed korištenja mobilne aplikacije

Arhitektura mobilne Android aplikacije za višekriterijsko preporučivanje turističkih odredišta sastoji se od korisničke i poslužiteljske strane. Poslužiteljska strana obuhvaća korištenje dvije baze podataka, a to su baza korisnika i baza podataka o turističkim odredištima. Osim toga, ova strana koristi i Google Maps API za geolokacijske funkcionalnosti. Logika aplikacije s korisničke strane prikazana je dijagramom tijeka na slici 3.1. Na dijagramu tijeka je prikazan slijed korištenja mobilne android aplikacija za višekriterijsko preporučivanje turističkih odredišta. Prikazano da pri registriranju korisnika aplikacija sprema podatke u bazu, a pri prijavi koristi spremljene podatke kako bi autentificirala korisnika. Nakon prijave se od korisnika očekuje odabir preferiranih kategorija da bi se zatim stvorio popis preporučenih odredišta, na koji će se primijeniti odabrani filtar. Korisnik ima mogućnost odabira između različitih kartica na navigacijskoj traci što je prikazano strelicama koje idu u oba smjera. Dijagram tijeka također prikazuje komunikaciju s bazom podataka o turističkim odredištima. Podaci se dohvaćaju iz ove baze prilikom prikaza informacija o odredištima i ažuriraju kada korisnik dodjeljuje ocjene.



Slika 3.1 Dijagram tijeka mobilne Android aplikacije za višekriterijsko preporučivanje turističkih odredišta

3.4 Sustav preporučivanja

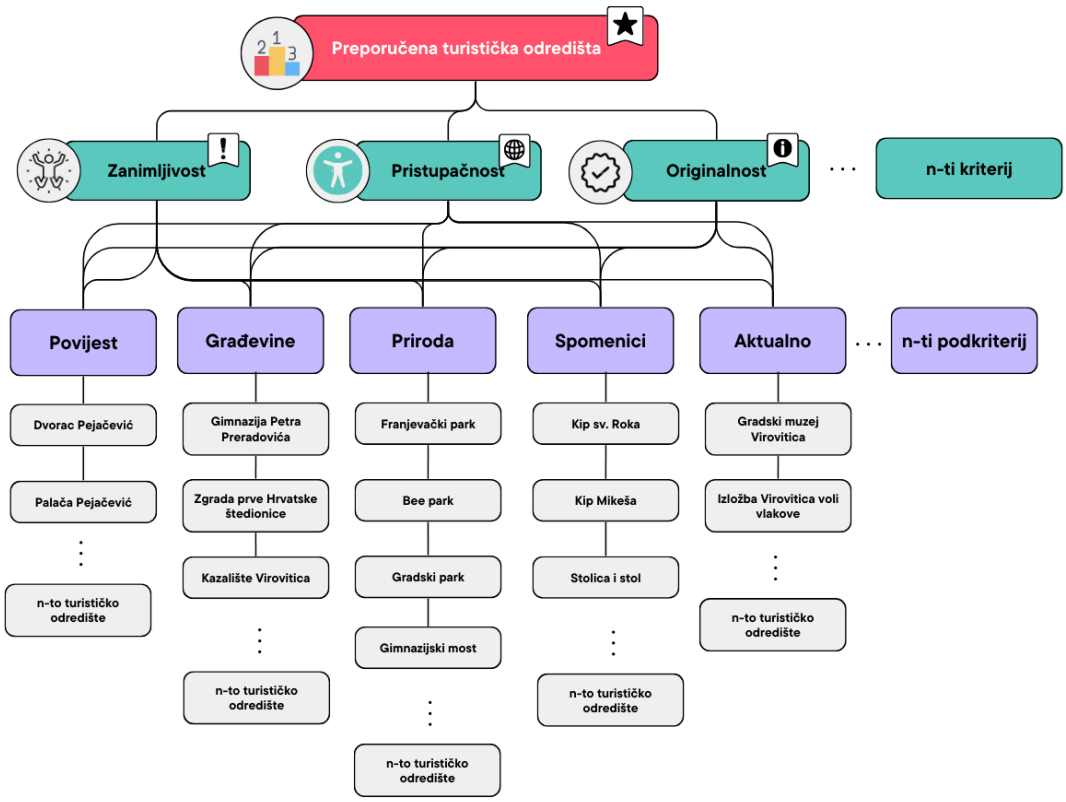
Prema [9], kolaborativno filtriranje je najčešće korišten pristup u sustavima preporučivanja. Prvenstveno se temelji na ocjenama koje korisnici daju predmetima u danoj domeni. Računa sličnosti između profila više korisnika na temelju njihovih danih ocjena i generira nove preporuke temeljene na usporedi korisničkih ocjena. Kolaborativno filtriranje stvara pretpostavke ili preporuke tako da je predviđanje numerička vrijednost, a preporuka je lista od gornjih N predmeta koje će se najviše svidjeti korisniku [6].

Kolaborativno filtriranje može biti korisnički i predmetno orijentirano. Cilj korisničkog orijentiranog filtriranja je preporučiti predmete temeljene na ocjenama koje daju drugi članovi koji su već pogledali te predmete. Ako član taj predmet smatra zanimljivim, bit će automatski preporučen drugim korisnicima koji su izrazili slična mišljenja u prošlosti [6]. Nadalje, predmetno orijentirano filtriranje se oslanja na opise sadržaja predmeta i relevantne korisničke profile da bi generirao personalizirane preporuke. Predstavlja korisnikov profil temeljen na prethodno odabranim predmetima korisnika. Na taj način generira preporuke temeljene na pronalasku predmeta koji su slični onim koje je korisnik preferirao u prošlosti [9]. Također, postoji i hibridna kategorija kolaborativnog filtriranja koja koristi obje potkategorije i smatra se boljom od korištenja samo jedne potkategorije.

U većini sustava za preporuke koristi se jednokriterijska vrijednost koja predstavlja ukupnu ocjenu predmeta koju je dao korisnik, ali ta jedna ocjena ne prikazuje dovoljno dobro koliko su se korisniku svidjela sva svojstva tog predmeta. Uključivanje više kriterija može rezultirati preciznijim preporukama [9]. To znači da je korisniku neko turističko odredište bilo vrlo zanimljivo, ali vrlo nepristupačno i nije siguran koju ocjenu treba dati tom turističkom odredištu. Proširenjem kriterija korisnik može svakom kriteriju dati posebnu ocjenu, a time će drugi korisnici dobiti stvarnije informacije što mogu očekivati od tog turističkog odredišta i hoće li im biti pri vrhu popisa ako odaberu taj kriterij kao filtar. Drugim riječima, ključ učinkovitog sustava preporučivanja je sposobnost razumijevanja ne samo što se korisniku sviđa, već i zašto mu se sviđa [9].

U ovom završnom radu koristi se višekriterijsko preporučivanje. Svakom turističkom odredištu dodijeljena je jedna ili više kategorija, što znači da se provodi kategorizacija svakog turističkog odredišta prije nego što može biti preporučeno korisniku. Na slici 3.2 odredišta su prikazana ispod kategorije, to jest podkriteija kojem pripadaju. Višekriterijski sustav preporučivanja kao jedan od

parametara koristi korisnikov odabir podkriterija koji ga zanimaju, a oni mogu biti: aktualno, duhovnost, eksperimenti, građevine, interaktivno, spomenici, povijest, priroda, umjetnost i sport. Na temelju odabira slaže se lista turističkih odredišta koja se podudaraju s odabirom korisnika, a to je poznato kao predmetno orijentirano filtriranje. Preporuke se dodatno sužavaju kada korisnik odabere filtar za rangiranje preporučenih turističkih odredišta kao još jedan parametar višekriterijskog preporučivanja, a ponuđeni filtri su: zanimljivost, pristupačnost, originalnost, fotogeničnost i vremenska zahtjevnost odredišta. Na slici 3.2 filtri su navedeni kao kriteriji i oni su zadnji odabir prije nego se korisniku prikaže rangirana lista preporučenih turističkih odredišta. Nakon primjene filtra, sustav rangira odredišta prema zadanim kriterijima. Ovdje se koristi prosječna ocjena kao glavni faktor za rangiranje. Korištenje ocjena drugih korisnika je poznato kao korisnički orijentirano filtriranje. Odredišta koja su bolje ocijenjena prikazuju se s većim prioritetom.



Slika 3.2 Prikaz kriterija i podkriterija za višekriterijsko preporučivanje

Pojednostavljeni pseudo-kod algoritma aplikacije za višekriterijsko preporučivanje turističkih odredišta je prikazan na slici 3.2. Svakom turističkom odredištu se dodjeljuje jedna ili više kategorija, a zatim se za preferirane kategorije korisnika izdvajaju odredišta koja odgovaraju tim uvjetima. Lista izdvojenih turističkih odredišta se sortira prema filtru kojeg korisnik odabere kako bi korisnik na kraju imao personaliziranu rangiranu listu.

```
Za svaku lokaciju u popisu lokacija:
    lokacija.kategorija = dodijeljenaKategorija

Za svaku stavku u omiljenimKategorijamaKorisnika:
    Za svaku lokaciju u popisu lokacija:
        Ako stavka.kategorija == lokacija.kategorija:
            lista.dodaj(lokacija)

lista.sortiraj(premaOdabranomFilteruKorisnika)
prikažiListuKorisniku(lista)
```

Slika 3.3 Pseudo-kod algoritma za višekriterijsko preporučivanje turističkih odredišta

4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA VIŠEKRITERIJSKO PREPORUČIVANJE TURISTIČKIH ODREDIŠTA

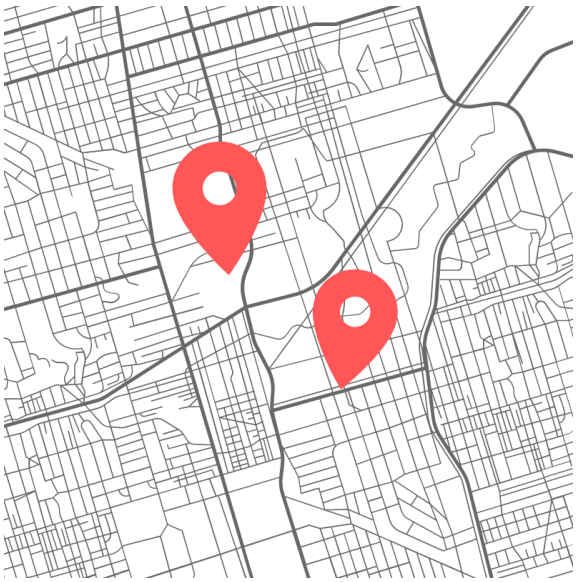
U ovom poglavlju su prikazane korištene tehnologije za ostvarivanje aplikacije za višekriterijsko preporučivanje turističkih odredišta, a zatim su opisani i prikazani najvažniji dijelovi programskog rješenja na strani korisnika i poslužitelja.

4.1 Korištene tehnologije i alati

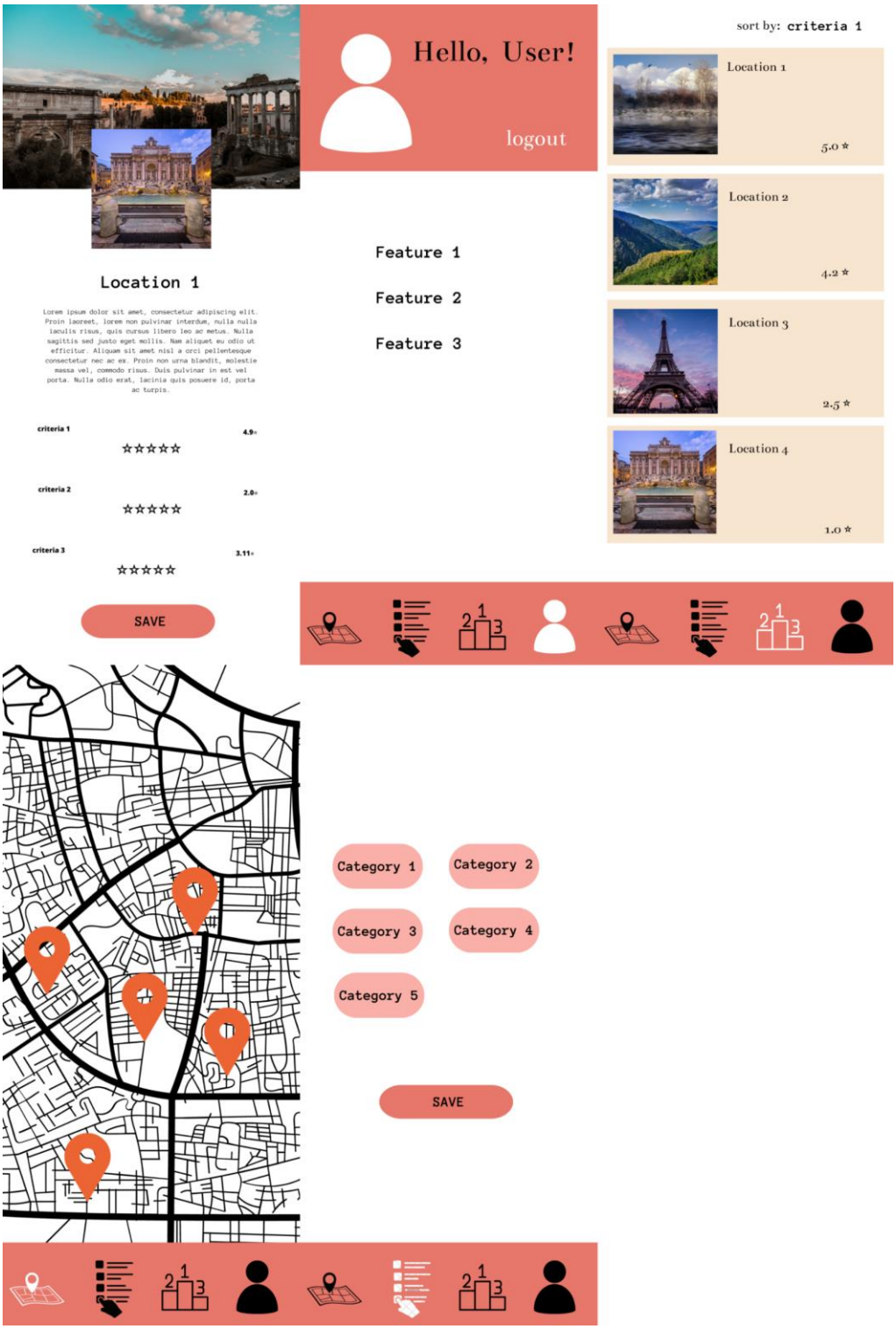
Prvotni dizajn aplikacije za višekriterijsko preporučivanje turističkih odredišta napravljen je u online alatu Canvi. Korištenjem programskog jezika Kotlin i XML za korisničko sučelje, aplikacija je razvijena u programskog okruženju Android Studio. Za pohranu podataka u oblaku korišten je Firebase. Svi navedeni alati bit će dodatno objašnjeni u slijedećim poglavljima.

4.1.1 Canva

Canva [10] je on-line alat koji korisnicima omogućava dizajniranje raznih grafika, kao na primjer: poslovne prezentacije, infografike, plakate, medijske objave za društvene mreže i slično. Sadrži mnogo predložaka, alata za uređivanje teksta i slika, kao i samu biblioteku slika i ikona koju korisnici mogu upotrijebiti za vlastiti dizajn. Mnogi korisnici preferiraju Canvu jer omogućava laku izradu dizajna bez potrebe za učenjem kako određeni alati funkcioniraju ili pravila dizajna. Za potrebe ovog rada korištena je za izradu logotipa prikazanog na slici 4.1 i za izradu prvotnog prototipa izgleda aplikacija koji je prikazan na slici 4.2.



Slika 4.1 Logotip aplikacije



Slika 4.2 Ideja izgleda aplikacije

4.1.2 Android Studio

Android Studio je službeno integrirano razvojno okruženje za razvoj Android aplikacija prema [11]. Temeljen na snažnom uređivaču koda i alata za razvojne programere razvojnog okruženja IntelliJ IDEA, Android Studio nudi mnogo značajki koje povećavaju produktivnost pri izradi Android aplikacija. Jedne od tih značajki su vizualni uređivač korisničkog sučelja, upravljanje izgradnjom i pakiranjem aplikacije, virtualni emulator za testiranje aplikacije. Nudi i podršku za programiranje na više programskih jezika, uključujući Java, Kotlin i C++.

4.1.3 Kotlin

Prema [12], Kotlin je programski jezik koji se izvodi na Java Virtual Machine te se njegov kod vrlo lako može pretvoriti u JavaScript kod. Dizajniran je da bude više izražajan, ali sažetiji nego Java kod, dok u isto vrijeme rješava dio stvari koje su problematične kod Jave. Stvorila ga je tvrtka JetBrains koja stoji iza razvojnog okruženja IntelliJ IDEA.

4.1.4 XML

Prema [13], XML je kratica za Extensible Markup Language, odnosno jezik za označavanje podataka. Koristi se za spremanje i prijenos podataka te je dizajniran da bude razumljiv sam po sebi. Često je svrstan u programske jezike, ali XML kod nema nikakvu funkcionalnost obavljanja neke radnje, već je transportni jezik, tj. sastoji se od informacija koje su obavijene oznakama (engl. *tags*).

4.1.5 Firebase

Firebase je prema [14] platforma koja pomaže u razvijanju aplikacija i igrica. Pruža različite alate i usluge kao što je autentifikacija korisnika, baza u stvarnom vremenu, pohranu podataka i slika u oblak i još mnoge druge.

4.2 Programsko rješenje na strani korisnika

U ovom poglavlju bit će opisano programsko rješenje zadatka uz slike programskog koda.

4.2.1 Registracija i prijava korisnika

Kao što je već objašnjeno, prvim pokretanjem aplikacije korisniku se otvara zaslon za prijavu, u slučaju da korisnik nema račun, odabrat će gumb „registriraj se“ te unijeti tražene podatke i pritisnuti gumb „registracija“. Za registriranje korisnika korišten je Firebase Authentication te metoda „register“, prikazana na slici 4.3, koja kao parametre prima adresu e-pošte, lozinku i ime

kao slijed znakova (engl. *string*). Firebase Authentication se koristi za provjeru ispravnosti unesene adrese e-pošte, a provjera za lozinku zahtjeva od korisnika da unese lozinku minimalne dužine od osam znakova, koja mora sadržavati barem jedno slovo i barem jedan broj. Ako su zadovoljeni svi uvjeti, u Firebaseu se kreira korisnik s automatskom i jedinstvenom oznakom i unesenim podacima. Ako već postoji korisnik s navedenom adresom e-pošte, na zaslonu se prikazuje poruka sa skočnim prozorom (engl. *Toast message*) „Email već postoji ili je neispravan“. U slučaju kada je adresa e-pošte ispravna, a lozinka ne ispunjava uvjete, korisniku će se na zaslonu prikazati *Toast* poruka što treba učiniti kako bi se mogao uspješno registrirati.

```
fun register(email: String, password: String, name: String){
    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                Toast.makeText(
                    baseContext, text: "Račun uspješno kreiran!",
                    Toast.LENGTH_SHORT
                ).show()
                val user = auth.currentUser
                val data = hashMapOf("name" to name)
                user?.uid?.let {  it: String
                    db.collection( collectionPath: "users").document(it)  DocumentReference
                        .collection( collectionPath: "documents")  CollectionReference
                        .document( documentPath: "user-info").set(data)
                }
                val initalData = hashMapOf(
                    "selectedChips" to "exists"
                )
                user?.uid?.let { it1 ->
                    db.collection( collectionPath: "users")  CollectionReference
                        .document(it1).collection( collectionPath: "documents")
                        .document( documentPath: "categoryPreferences")  DocumentReference
                        .set(initalData)
                }
                val chipselected:ArrayList<String> = arrayListOf()
                chipselected.add("exists")
                user?.let {  it: FirebaseUser
                    db.collection( collectionPath: "users")
                        .document(it.uid).collection( collectionPath: "documents")
                        .document( documentPath: "categoryPreferences").update( field: "selectedChips", chipselected)
                }
                val intent = Intent( packageContext: this, MainActivity::class.java)
                intent.putExtra( name: "registration", value: "true" )
                SavedStates.setnavigationBarIndex(1)
                startActivity(intent)
                finish()
            }
            else {
                Toast.makeText(baseContext,  text: "Email već postoji ili je neispravan", Toast.LENGTH_SHORT).show()
            }
        }
    return
}
```

Slika 4.3 Programski kod metode register

Ako korisnik već ima račun, a nije se prethodno prijavio na tom uređaju, onda će u predviđena polja unijeti adresu e-pošte i lozinku s kojom se registrirao. Uneseni podaci se predaju metodi login, prikazanoj na slici 4.4, koja pomoću Firebase Autentification provjerava ispravnost podataka. Ako su podaci ispravni korisniku se prikaže *Toast* poruka „Uspješna prijava“. U

suprotnom, prikazat će se Toast poruka „Pogrešni email ili lozinka“, što ne daje do znanja postoji li već korisnički račun s tom adresom e-pošte ili je korisnik samo unio krivu lozinku te se na taj način štiti od potencijalnih hakerskih napada.

```
fun login(email: String, password: String){
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                val intent = Intent( packageContext: this, MainActivity::class.java)
                startActivity(intent)
                Toast.makeText( context: this, text: "Uspješna prijava",Toast.LENGTH_SHORT).show()
                finish()
            }
            else {
                Toast.makeText(baseContext,  text: "Pogrešni email ili lozinka",
                    Toast.LENGTH_SHORT).show()
                val email = findViewById<EditText>(R.id.editTextTextEmailAddress)
                val password = findViewById<EditText>(R.id.editTextTextPassword)
                email.text.clear()
                password.text.clear()
            }
        }
}
```

Slika 4.4 Programski kod metode login

Ako se korisnik na istom mobilnom uređaju već prijavio, neće se morati ponovno prijaviti pri svakom novom pokretanju aplikacije. U slučaju da korisnik zaboravi lozinku, može odabrati gumb „Zaboravili ste lozinku“. Otvorit će mu se skočni prozor (engl. *Pop-up*) u kojem je potrebno unijeti adresu e-pošte s postojećim korisničkim računom te pritisnuti gumb „POŠALJI” pri čemu će mu upute za ponovno postavljanje lozinke biti poslane na unesenu adresu e-pošte, samo ako postoji račun s tom adresom e-pošte.

4.2.2 Google Maps i prikaz turističkih odredišta na karti

Registriranim korisnicima će se otvoriti drugačiji zaslon za razliku od onih koji su se samo prijavili. Prijavljenim korisnicima se pri otvaranju aplikacije otvara zaslon s kartom na kojoj se nalaze markeri. Dohvaćanje karte odvija se pomoću Google Cloud platforme i koristi se Google Maps SDK for Android API za koji se s prijavljenim Google računom dohvaća API ključ.

Iz Firebase baze podataka povlače se podaci o svim turističkim odredištima, ali se uzimaju samo ID i koordinate svakog odredišta u obliku objekata klase MapMarker i prikazuju se kao markeri na karti. Korisnici mogu odabrati bilo koji marker te će im se nakon odabira otvoriti novi prozor s više informacija o odabranom odredištu. Dio programskog koda za dohvaćanje podataka je prikazan na slici 4.5, a klasa MapMarker prikazana je na slici 4.6.

```
override fun onMapReady(googleMap: GoogleMap) {
    mMap = googleMap
    mMap.moveCamera(CameraUpdateFactory.newCameraPosition(CameraBounds.getCameraPosition()))
    mMap.setOnCameraMoveListener { CameraBounds.setCameraPosition( mMap.cameraPosition) }
    val locations = mutableListOf<MyMarker>()
    val docRef = db.collection( collectionPath: "places")
    docRef.get()
        .addOnSuccessListener { documents ->
            for (document in documents.documents) {
                val coordinates = LatLng(
                    document.data!!["latitude"].toString().toDouble(),
                    document.data!!["longitude"].toString().toDouble()
                )
                locations.add(MyMarker(document.id, coordinates))
            }
        }
        .addOnCompleteListener{ it: Task<QuerySnapshot!>
            for (location in locations) {
                val myMarker = mMap.addMarker(MarkerOptions().position(location.cordinates))
                myMarker!!.tag = location.id
                markers.add(myMarker)
            }
```

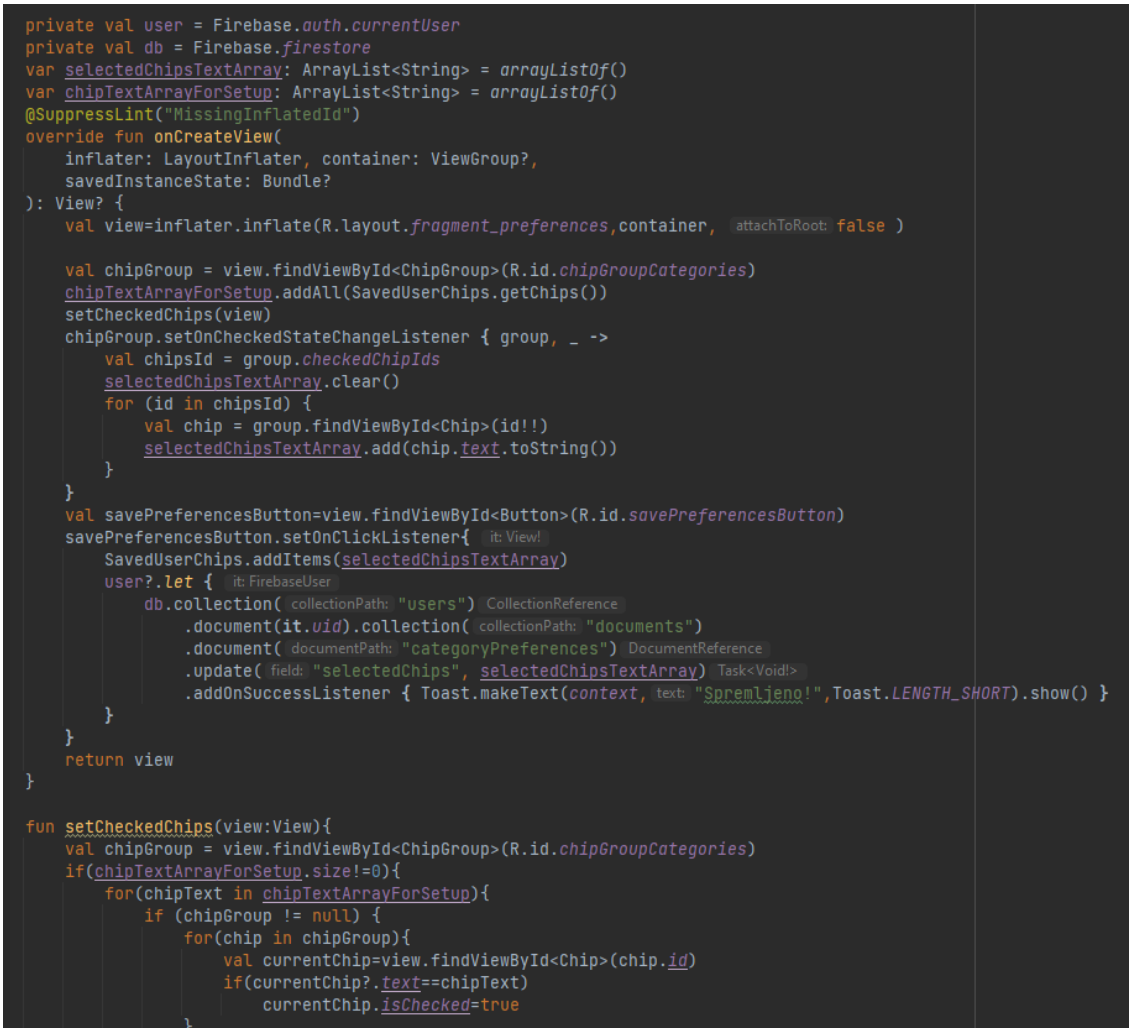
Slika 4.5 Isječak programskog koda za prikazivanje markera turističkih odredišta na karti

```
data class MapMarker(
    var id: String,
    var cordinates: LatLng
)
```

Slika 4.6 Prikaz klase MapMarker

4.2.3 Odabir preferiranih kategorija

Za tek registrirane korisnike, prva aktivnost nakon registracije je aktivnost s mogućnosti biranja preferiranih kategorija. Prijavljenim korisnicima ovo je druga kartica na navigacijskoj traci pri dnu ekrana. Kategorije odredišta su prikazane kao čipovi (engl. c*hips*). Korisnik može odabrati nijedan, jedan ili više čipova te će se oni spremiti u bazu nakon što korisnik odabere „SPREMI“. Broj izmjena preferiranih čipova nije ograničen. Na temelju tog odabira kreiraju se preporuke odredišta na kartici Rang. Programski kod za spremanje odabranih čipova je prikazan na slici 4.7.



```
private val user = Firebase.auth.currentUser
private val db = Firebase.firestore
var selectedChipsTextArray: ArrayList<String> = arrayListOf()
var chipTextArrayForSetup: ArrayList<String> = arrayListOf()
@SuppressLint("MissingInflatedId")
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    val view=inflater.inflate(R.layout.fragment_preferences,container, attachToRoot: false )

    val chipGroup = view.findViewById<ChipGroup>(R.id.chipGroupCategories)
    chipTextArrayForSetup.addAll(SavedUserChips.getChips())
    setCheckedChips(view)
    chipGroup.setOnCheckedStateChangeListener { group, _ ->
        val chipsId = group.checkedChipIds
        selectedChipsTextArray.clear()
        for (id in chipsId) {
            val chip = group.findViewById<Chip>(id!!)
            selectedChipsTextArray.add(chip.text.toString())
        }
    }
    val savePreferencesButton=view.findViewById<Button>(R.id.savePreferencesButton)
    savePreferencesButton.setOnClickListener{ it: View!
        SavedUserChips.addItems(selectedChipsTextArray)
        user?.let { it: FirebaseUser
            db.collection( collectionPath: "users") CollectionReference
                .document(it.uid).collection( collectionPath: "documents")
                .document( documentPath: "categoryPreferences") DocumentReference
                .update( field: "selectedChips", selectedChipsTextArray) Task<Void!>
                .addOnSuccessListener { Toast.makeText(context, text: "Spremljeno!",Toast.LENGTH_SHORT).show() }
        }
    }
    return view
}

fun setCheckedChips(view:View){
    val chipGroup = view.findViewById<ChipGroup>(R.id.chipGroupCategories)
    if(chipTextArrayForSetup.size!=0){
        for(chipText in chipTextArrayForSetup){
            if (chipGroup != null) {
                for(chip in chipGroup){
                    val currentChip=view.findViewById<Chip>(chip.id)
                    if(currentChip?.text==chipText)
                        currentChip.isChecked=true
                }
```

Slika 4.7 Programski kod za spremanje odabranih čipova

S obzirom da se popis odabranih kategorija koristi u više od jedne aktivnosti, napravljen je objekt koji sadrži te kategorije spremljene kako se ne bi moralo pri svakom pokretanju neke aktivnosti iznova dohvaćati podatke iz baze. Dohvaćanje se radi samo pri prvom pokretanju ili izmjeni podataka. Programski kod za objekt je prikazan na slici 4.8. Izmjena i dohvaćanje podataka se odvija pomoću metoda addItems i getChips koje su također prikazane na slici 4.8.

```
object SavedUserChips {
    var list: ArrayList<String> = arrayListOf()

    fun addItems(items:ArrayList<String>) {
        list.clear()
        list.addAll(items)

    }

    fun getChips(): List<String> = list
}
```

Slika 4.8 Objekt s listom odabranih čipova s pripadnim metodama

4.2.4 Preporučena odredišta i odabir filtara preporuke

Prilikom svakog otvaranja treće kartice na navigacijskoj traci, dohvaćaju se podatci o svim turističkim odredištima u obliku klase LocationData koja je prikazana na slici 4.9. Zatim, uzima se indeks padajućeg izbornika koji prikazuje dostupne filtre, a pomoću kojeg će se korisniku rangirati lista. Uzimaju se spremljeni čipovi, a zatim se prolazi kroz listu svih turističkih odredišta i izdvajaju se odredišta koja odgovaraju korisnikovoj sklonosti. Postupak se obavlja unutar metode updateArrayWithChips koja je prikazana na slici 4.10. Ako je lista izdvojenih turističkih odredišta prazna, to jest nema odabranih čipova, na ekranu će biti prikazan tekst koji upućuje korisnika da odabere neke čipove kako bi mu se turistička odredišta mogla preporučiti. U slučaju da lista nije prazna, pozvat će se metoda performSort, prikazana na slici 4.11, koja na temelju odabranog indeksa padajućeg izbornika sortira listu prema odabranom filtru te poziva metodu showList, koja je prikazana na slici 4.12, koja će izdvojenu i rangiranu listu prikazati korisniku. Svaka kartica liste predstavlja jedno turističko odredište te se može kliknuti i time se otvara novi zaslon s detaljima odabranog turističkog odredišta. Kartice koje su pri vrhu imaju najveće ocijene i time pored sebe imaju drugačiju oznaku kako bi korisniku bilo jasnije da su upravo ta turistička odredišta najbolji izbor za njega.

```
data class LocationData(
    var id:String="",
    var name:String="",
    var category:ArrayList<String> = arrayListOf(),
    var latitude:Double?=null,
    var longitude:Double?=null,
    var image1:String="",
    var image2:String="",
    var accessibilityAverage:Double?=null,
    var excitementAverage:Double?=null,
    var originalityAverage:Double?=null,
    var photogenicAverage:Double?=null,
    var timeWorthAverage:Double?=null,
    var overallGrade:Double?=null
)
```

Slika 4.9 Prikaz klase LocationData

```
fun updateArrayWithChips(view:View) {
    val lista: ArrayList<MyLocation> = ArrayList()
    for (chip in selectedChips) {
        for (data in initialLocationList) {
            for(data2 in data.category){
                if (data2==chip) {
                    if (!lista.contains(data))
                        lista.add(data)
                }
            }
        }
    }

    val nothingSelectedTextView = view.findViewById<TextView>(R.id.onNothingSelected)
    val addIcon = view.findViewById<ImageView>(R.id.add_icon)
    if (lista.isEmpty()) {
        nothingSelectedTextView?.visibility = View.VISIBLE
        addIcon?.visibility = View.VISIBLE
    } else {
        changingLocationList = lista
        performSort()
        nothingSelectedTextView?.visibility = View.GONE
        addIcon?.visibility = View.GONE
    }
}
```

Slika 4.10 Programski kod metode updateArrayWithChips

```
fun performSort(){

    val position=spinnerIndex
    var sortedList: ArrayList<LocationData> = ArrayList()
    when(position){
        0->{
            sortedList =
                ArrayList(changingLocationList.sortedWith(compareByDescending { it: LocationData
                    it.overallGrade
                }))

        }
        1 -> {
            sortedList=
                ArrayList(changingLocationList.sortedWith(compareByDescending{ it: LocationData
                    it.excitementAverage
                }))
        }
        2 -> {
            sortedList=
                ArrayList(changingLocationList.sortedWith(compareByDescending{ it: LocationData
                    it.accessibilityAverage
                }))
        }
        3 -> {
            sortedList=
                ArrayList(changingLocationList.sortedWith(compareByDescending{ it: LocationData
                    it.originalityAverage
                }))
        }

        4 -> {
            sortedList =
                ArrayList(changingLocationList.sortedWith(compareByDescending { it: LocationData
                    it.photogenicAverage
                }))
        }
        5 -> {
            sortedList =
                ArrayList(changingLocationList.sortedWith(compareBy { it: LocationData
                    it.timeWorthAverage
                }))
        }

    }
    if(SavedUserChips.list.isNotEmpty()){
            showList(sortedList)
    }
}
```

Slika 4.11 Programski kod metode performSort

```
fun showList(list:ArrayList<MyLocation>) {
    val recyclerView = view?.findViewById<RecyclerView>(R.id.recycledviewer)
    recyclerAdapter = LocationRecyclerAdapter(list,spinnerIndex)
    recyclerView?.apply { this: RecyclerView
        layoutManager = LinearLayoutManager(context)
        this.adapter = recyclerAdapter
    }
}
```

Slika 4.12 Programski kod metode showList

4.2.5 Detalji turističkog odredišta

Odabirom markera na mapi ili kartice turističkog odredišta na zaslonu s preporučenim odredištima, otvara se aktivnost s detaljima. To uključuje prikaz slika, ime, kratki opis i, ako postoji, poveznica tog odredišta. Za pohranu slika koristi se Firebase Storage. Postoji i gumb „Prikaži na karti“ koji će korisniku na kartici mape prikazati lokaciju odredišta. Pri dnu aktivnosti nalaze se četiri mjerne skale za ocjenjivanje, po jedna za svaki filtar i jedna grupa radio gumbova. Pomoću njih korisnik može ocijeniti koliko mu je na primjer bilo zanimljivo ili fotogenično odredište ocjenom od jedan do pet, s tim da ocjena jedan predstavlja najmanje zadovoljstvo, a ocjena pet najveće zadovoljstvo određenim kriterijem. Metoda updateRating, koja ažurira vrijednosti i računa novi prosjek ocjena, nalazi se na slici 4.13. Metoda kao parametre prima vrijednost mjerne skale, jedinstvenu oznaku turističkog odredišta i ime kriterija kojeg je korisnik ocijenio. Pri svakom pozivu metode iz baze će se povući podatci o odredištu, provjeriti jesu li podatci različiti od nula i ako jesu na njih će se dodati ocjena korisnika, a zatim izračunati prosjek i svi podatci će se spremiti nazad u bazu. Metoda će pod korisnikov profil spremiti ocjenu koju je korisnik dodijelio tom odredištu.

```
fun updateRating(rating:RatingBar, id:String, filterName:String){
    if (rating.rating.toString() != "0.0" && !rating.isIndicator) {
        db.collection( collectionPath: "places").document(id).get().addOnSuccessListener { document ->
            var Count = document.data!!["${filterName}"+ "Count"].toString().toDouble()
            var Sum = document.data!!["${filterName}"+"Sum"].toString().toDouble()
            if(Sum==0.0){
            checkIfSumRatingIsZero(rating,id,filterName,Sum)
            }
            else {
                Sum += rating.rating.toDouble()
                Count += 1
                var average: Double = Sum / Count
                user?.let {  it: FirebaseUser
                    db.collection( collectionPath: "users")  CollectionReference
                        .document(it.uid).collection( collectionPath: "documents")
                        .document( documentPath: "${id}")  DocumentReference
                        .update( field: "${filterName}" + "Rating", rating.rating.toDouble())
                }
                rating.setIsIndicator(true)
                db.collection( collectionPath: "places").document(id).update(
                     field: "${filterName}" + "Sum", Sum,
                     ...moreFieldsAndValues: "${filterName}" + "Count", Count,
                    "${filterName}" + "Average", average
                )
                    .addOnSuccessListener {  it: Void!
                        Toast.makeText(
                             context: this,
                             text: "Spremljeno!",
                            Toast.LENGTH_SHORT
                        ).show()
                    }
                updateAllAverages(id)
            }
        }
    }
}
```

Slika 4.13 Programski kod metode updateRating

4.2.6 Profil korisnika

Na kartici korisničkog profila nalazi se gumb za odjavu kao i gumb koji će korisniku poslati e-mail s uputama za resetiranje lozinke ako ju je korisnik zaboravio. Postoji još i gumb za fragment u kojem piše nešto više o aplikaciji i gumb koji korisniku otvara fragment u kojem može napisati ako se suočio s nekim problemima u radu aplikacije. Programski kod za sve gumbove prikazan je na slici 4.14. Na istoj kartici korisniku je informativno prikazan broj turističkih odredišta kojima je dao barem jednu ocjenu.

```
val logout=view.findViewById<Button>(R.id.logout)
logout.setOnClickListener{ it: View!
    SavedStates.setnavigationBarIndex(0)
    FirebaseAuth.getInstance().signOut()
    val intent = Intent(context, LoginActivity::class.java)
    startActivity(intent)
    val activity = context as Activity?
    activity!!.finish()
}

val resetPasswordButton=view.findViewById<Button>(R.id.resetPasswordButton)
resetPasswordButton.setOnClickListener{ it: View!
    FirebaseAuth.getInstance().sendPasswordResetEmail(user?.email.toString())
    Toast.makeText(context, text: "Provjerite sandučić pošte!",
        Toast.LENGTH_LONG).show()
}

val aboutAppText=view.findViewById<TextView>(R.id.aboutApp)
aboutAppText.setOnClickListener{ it: View!
    val fragmentTransaction: FragmentTransaction?= activity?.supportFragmentManager?.beginTransaction()
    fragmentTransaction?.replace(R.id.container, AboutAppFragment())
    fragmentTransaction?.commit()
}

val helpCenter=view.findViewById<TextView>(R.id.helpCenter)
helpCenter.setOnClickListener{ it: View!
    val fragmentTransaction: FragmentTransaction?= activity?.supportFragmentManager?.beginTransaction()
    fragmentTransaction?.replace(R.id.container, HelpCenter())
    fragmentTransaction?.commit()
}
```

Slika 4.14 Isječak programskog koda kartice s detaljima korisnika

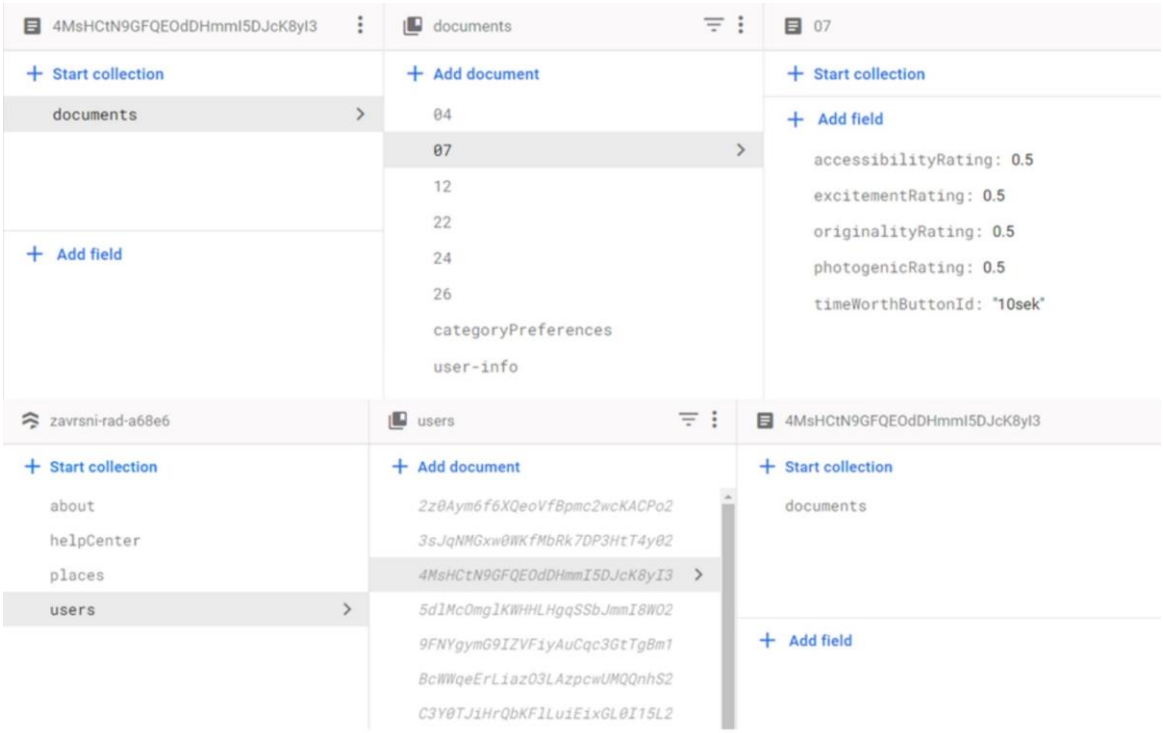
4.3 Programska rješenja na strani poslužitelja

4.3.1 Baza podataka

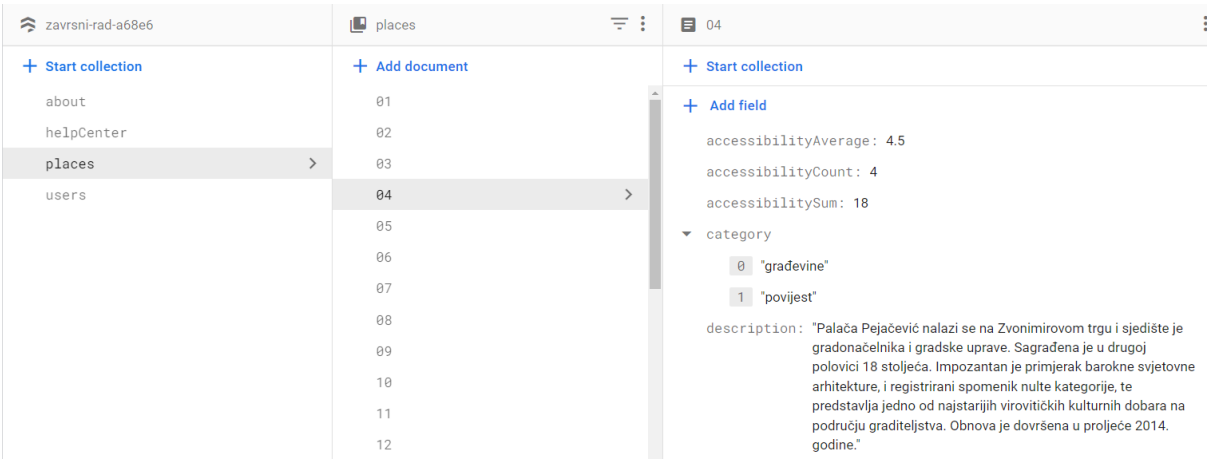
Kako bi aplikacija bila u potpunosti funkcionalna korištene su dvije baze podataka i obje su dio Cloud Firestore. Kao dio usluga koje nudi Firestore, korišten je i Storage za spremanje slika.

Jedna baza podataka ima spremljene sve korisnike i u nju se spremaju kategorije koje korisnik označi, kao i sve ocjene koje korisnik dodijeli određenom odredištu te su spremljeni i svi korisnički podaci. Svaki korisnik ima svoju jedinstvenu oznaku koja se preuzima iz Firebase Authentication, a korisniku je nasumično dodijeljena pri registraciji.

Druga baza podataka je baza sa svim odredištima. Svako odredište ima ID i pod njega se sprema ime odredišta, kategorije kojima odredište pripada, opis, poveznice na mjesta gdje su slike spremljene, poveznicu na web stranicu (ako postoji), koordinate, i za svaki kriterij ocjenjivanja broj osoba koje su ocijenile, ukupni zbroj ocjena i prosjek svih ocjena. Organizacija baze podataka s korisnicima je prikazana na slici 4.15, a za bazu podataka odredišta na slici 4.16.



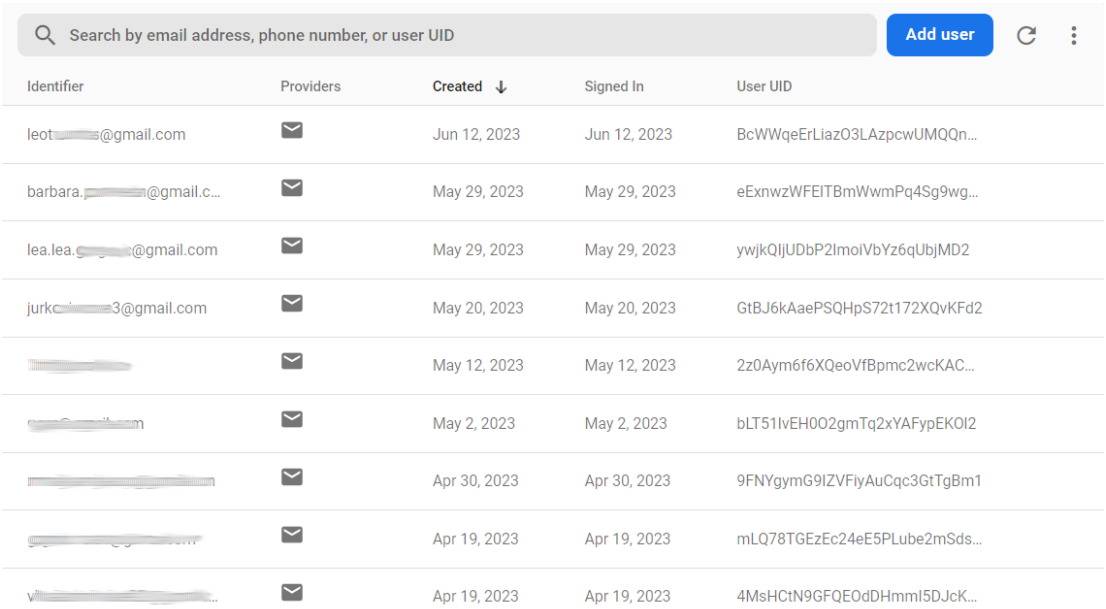
Slika 4.15 Baza podataka korisnika



Slika 4.16 Baza podataka odredišta

4.3.2 Autentifikacija korisnika

Za autentifikaciju korisnika koristi se Firebase Authentication koji ima već gotove metode za provjeru unesenih podataka. Nekoliko spremljenih i aktivnih korisničkih računa unutar Firebase Authentication prikazani su na slici 4.17. Firebase Authentication sprema podatke kao što je adresa e-pošte, jedinstvena korisnička oznaka, datum kreiranja i prijave i lozinku koju nije moguće vidjeti niti izmijeniti iz baze podataka, već korisnik to mora učiniti sam tako da ponese zahtjev za ponovno postavljanje lozinke direktno iz aplikacije ili da se obrati vlasniku baze koji to može učiniti iz Firebase Authentication. Taj postupak će poslati korisniku e-poštu s daljnjim uputama.



Slika 4.17 Registrirani korisnici u sustavu

4.3.3 Google Maps karta

Unutar Google Cloud postoji Google Maps Platforma koja korisnicima nudi API Maps SDK for Android. Aplikacija koristi taj API zajedno s njegovim funkcijama kao kartu s pinovima, a inače se može koristiti i za iscrtavanje ruta. Korisnik dobiva jedinstveni ključ koji mu omogućuje dohvaćanje karte i interaktivnost s njom. Dio programskog koda za učitavanje karte unutar aplikacije kada se otvori kartica s mapom je prikazan na slici 4.18. Stvara se instanca GoogleMap, a zatim se dodaju na kartu markeri koji zemljopisnu širinu i duljinu učitavaju iz baze podataka turističkih odredišta.

```
class MapFragment:Fragment(), OnMapReadyCallback {
    private lateinit var mMap: GoogleMap
    private val db = Firebase.firestore
    var marker: Marker? = null
    val markers = mutableListOf<Marker?>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {...}

    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap
        mMap.moveCamera(CameraUpdateFactory.newCameraPosition(CameraBounds.getCameraPosition()))
        mMap.setOnCameraMoveListener { CameraBounds.setCameraPosition( mMap.cameraPosition) }
        val locations = mutableListOf<MapMarker>()
        val docRef = db.collection( collectionPath: "places")
        docRef.get()
            .addOnSuccessListener { documents ->
                for (document in documents.documents) {
                    val coordinates = LatLng(
                        document.data!!["latitude"].toString().toDouble(),
                        document.data!!["longitude"].toString().toDouble()
                    )
                    locations.add(MapMarker(document.id, coordinates))
                }
            }
```

Slika 4.18 Programski kod za rad s kartom

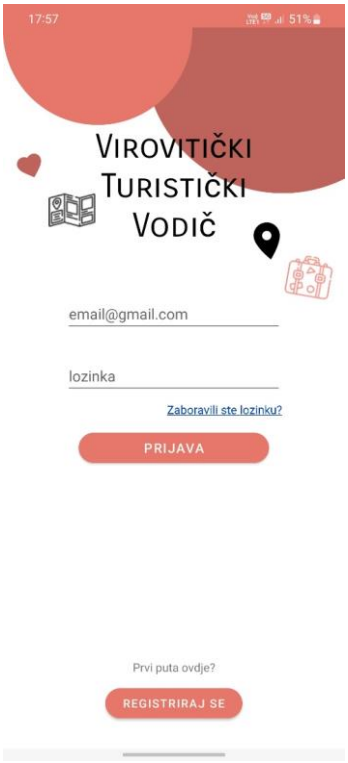
5. NAČIN KORIŠTENJA, ISPITIVANJE MOBILNE APLIKACIJE I ANALIZA

U ovom dijelu rada opisan je rad mobilne Android aplikacije za višekriterijsko preporučivanje turističkih odredišta.

5.1 Korištenje ostvarene mobilne Android aplikacije

Kako bi aplikaciju bilo moguće koristiti, prvo se mora instalirati, a zatim nakon pokretanja aplikacije korisniku se prikazuje početni zaslon, prikazan na slici 5.1, gdje se nalaze polja za prijavu. Ukoliko korisnik već ima račun potrebno je u odgovarajuća polja unijeti adresu e-pošte i lozinku. Ako korisnik zaboravi lozinku, ponuđena mu je opcija ponovnog postavljanja lozinke. Klikom na tekst „Zaboravili ste lozinku?“ prikazati će mu se skočni prozor (engl. *Pop-up window*) gdje korisnik može unijeti adresu e-pošte koju je koristio prilikom registracije te će mu biti poslan e-mail s uputama za ponovno postavljanje lozinke.

Ako korisnik nema račun pritisnut će gumb „REGISTRIRAJ SE“ koji ga vodi na zaslon za registriranje koji je prikazan na slici 5.2.



Slika 5.1 Sučelje za prijavu

Slika 5.2 Sučelje za registraciju

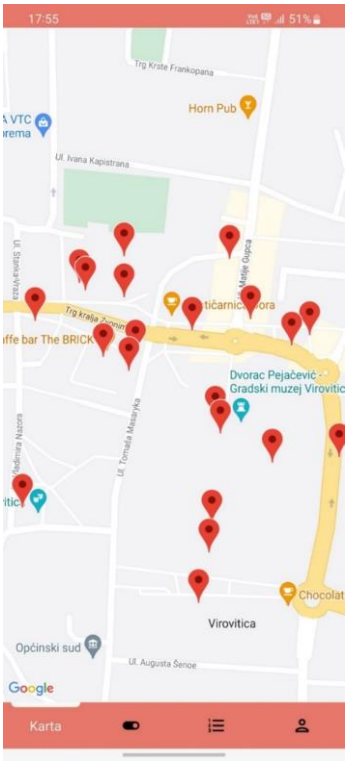
Za korištenje aplikacije, korisnik mora izraditi korisnički račun. Pri registraciji u za to predviđena polja, potrebno je unijeti ime, adresu e-pošte i lozinku čija minimalna dužina iznosi osam znakova. Lozinka mora sadržavati barem jedno slovo i barem jedan broj. U suprotnom, korisniku će se prikazati *Toast* poruka da lozinka ne zadovoljava uvjete.

Nakon registracije, korisniku se prikazuje skočni prozor s porukom dobrodošlice, a nakon njega zaslon na kojem treba odabrati kategorije turističkih odredišta koje su od njegovog interesa. Zaslon s čipovima je prikazan na slici 5.3. Odabrane kategorije neće se spremiti sve dok korisnik ne pritisne „SPREMI“, a moguće ih je promijeniti bilo kada.

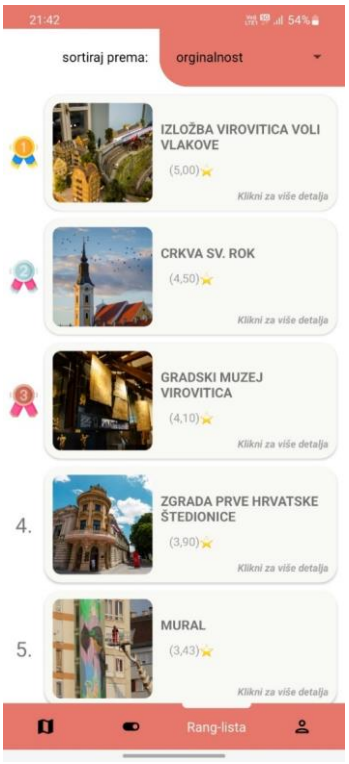
Slika 5.3 Sučelje za odabir kategorija

Prva kartica navigacijske trake će korisnika odvesti na zaslon s mapom na kojoj su pomoću pinova prikazana sva unesena turistička odredišta. To je ujedno i zadani početni zaslon za korisnika koji već ima postojeći korisnički račun i pri zadnjem ulasku u aplikaciju se prijavio i ostao prijavljen. Isječak karte prikazan je na slici 5.4 i klikom na bilo koji pin korisniku će se otvoriti novi zaslon s detaljnim informacijama o odabranom odredištu.

Treća kartica navigacijske trake korisniku prikazuje zaslon s preporučenim turističkim odredištima. Po zadanom su odredišta sortirana prema općoj ocjeni, a na korisniku je da odabere prema kojem filtru želi da mu se preporuče odredišta. Promjena filtra radi se klikom na „najbolje ocijenjeno“ u gornjem desnom kutu. Primjer preporučenih odredišta s primijenjenim filtrom „originalnost“ prikazan je na slici 5.5.



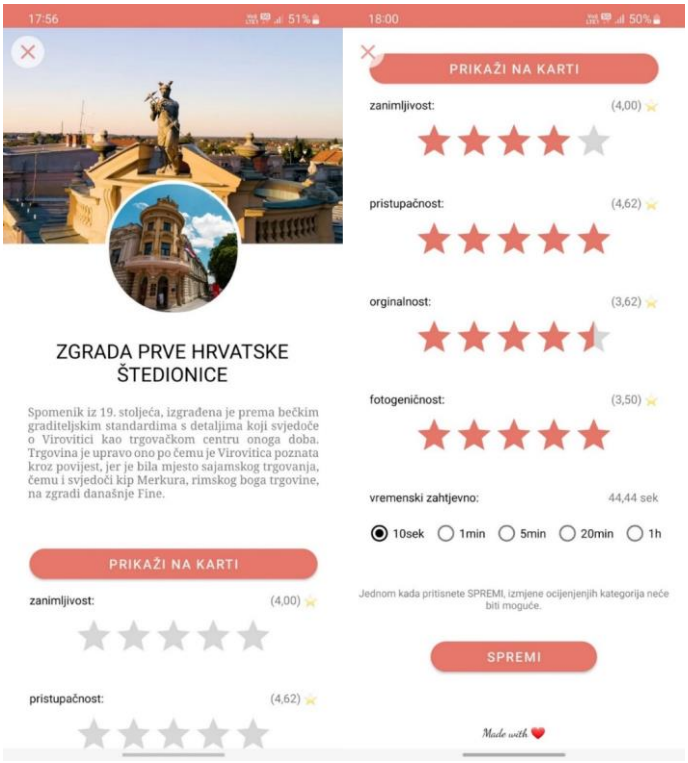
Slika 5.4 Prikaz karte



Slika 5.5 Sučelje s preporučenim odredištima

Korisnik također može pritisnuti na bilo koje odredište, a primjer zaslona koji će mu se otvoriti prikazan je na slici 5.6. Zaslon s informacijama određenog turističkog odredišta sadrži slike tog odredišta, njegov naziv, opis, gumb „PRIKAŽI NA KARTI“ koji će korisnika vratiti na prvu karticu i istaknuti mu odredište za koje je htio znati gdje se nalazi. Ako postoji web stranica o odredištu, prikazat će se poveznica koji će klikom na njega korisnika odvesti na tu web stranicu.

Ispod prethodno navedenih stvari nalaze se četiri mjerne skale i jedna grupa radio gumbova. Od korisnika se očekuje da dodjeli ocjene ponuđenim filtrima prema osobnom mišljenu. Jednom dodijeljene ocjene se neće moći izmijeniti nakon što korisnik pritisne „SPREMI“.



Slika 5.6 Sučelje detalja nekog odredišta

Četvrta kartica navigacijske trake korisniku prikazuje zaslon s njegovim profilom koji je prikazan na slici 5.7. Korisnik se na ovom zaslonu može odjaviti se pomoću gumba „ODJAVA“, zatražiti promjenu lozinke i vidjeti broj turističkih odredišta kojima je dao bar jednu ocjenu. Postoje i dva gumba u obliku teksta koji korisniku daju mogućnost da sazna nešto više o samoj aplikaciji ili da mu se otvori zaslon na kojemu može prijaviti probleme u aplikaciji.

Slika 5.7 Sučelje korisničkog profila

5.2 Ispitivanje mobilne aplikacije

Cilj ispitivanja mobilne aplikacije je dokazati njezinu ispravnost i funkcionalnost. Korisnički slučaj kada nema odabranih kategorija pokazuje da aplikacija neće korisniku preporučivati općenita turistička odredišta dok drugi korisnički slučaj pokazuje da izmjenom odabranih kategorija korisniku se preporučuju drugačija odredišta. Treći korisnički slučaj pokazuje da korisnik uistinu ima utjecaj na rangiranje odredišta, a četvrti korisnički slučaj pokazuje da se rang odredišta mijenjaju promjenom odabranog filtra.

5.2.1 Korisnički slučaj – Nema odabranih kategorija

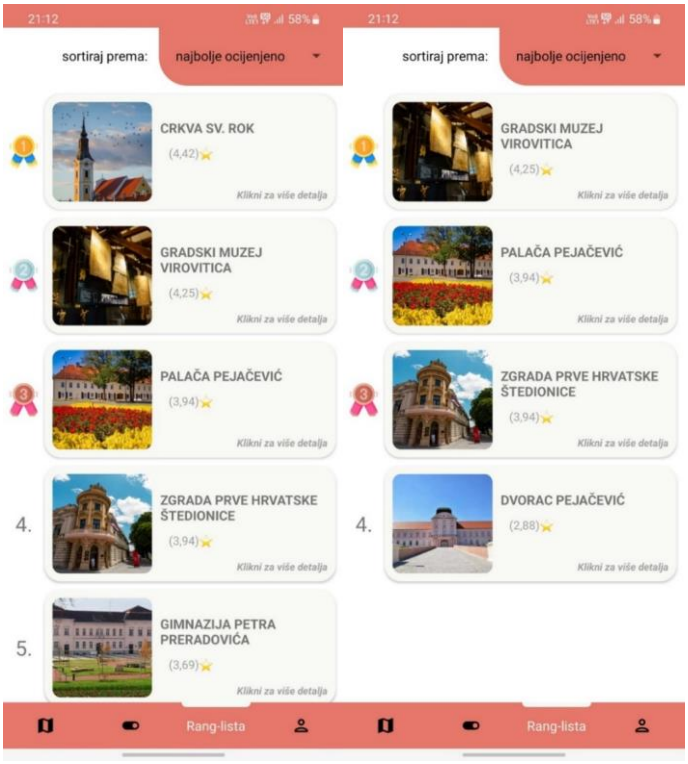
Kada korisnik izradi novi profil, početno stanje odabranih kategorija od interesa je nula, a može se dogoditi da je odlučio ne odabrati nijednu kategoriju turističkih odredišta. Cilj aplikacije je korisniku preporučiti turistička odredišta na temelju više kriterija i iz tog razloga mu neće biti prikazano nijedno odredište na kartici „Rang-lista“. Na ekranu će biti prikazan tekst koji korisnika usmjerava na drugu karticu „Preferencije“ da odabere kategorije koje su mu od interesa kako bi mu se prikazala preporučena turistička odredišta. Izgled aktivnosti je prikazan na slici 5.8. Aplikacija se neće srušiti niti raditi neočekivane ishode ako nema odabranih kategorija, ali neće ni preporučiti turistička odredišta korisniku.

Slika 5.8 Prikaz zaslona preporučenih odredišta kada nije odabrana nijedna kategorija

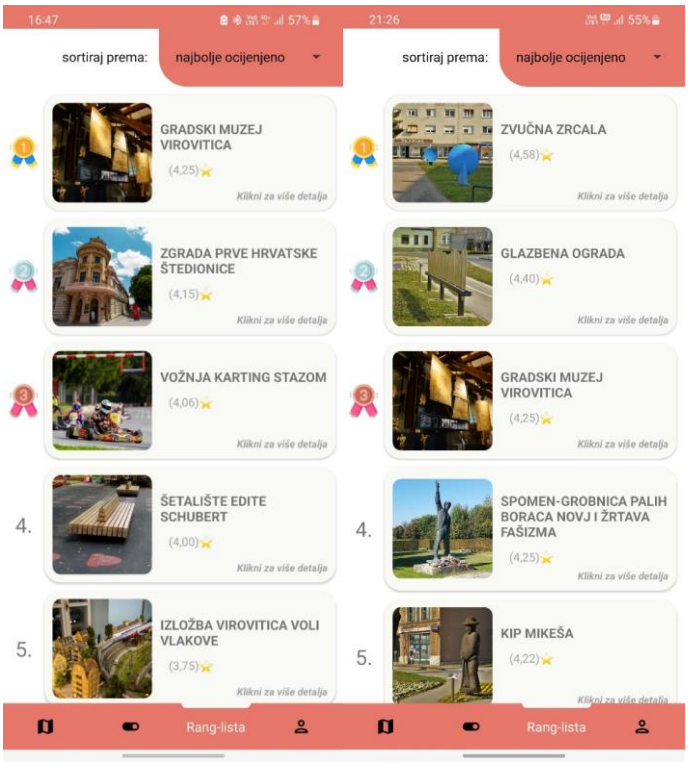
5.2.2 Korisnički slučaj – Jedna ili više odabranih kategorija

Korisnika možda zanima samo jedna kategorija odredišta, a možda želi odabrati nekoliko srodnih kategorija ili će ih označiti sve jer ga zanima baš sve što se može vidjeti. Spremljena turistička odredišta nisu ograničena da će se preporučiti samo pod jednom kategorijom već se pojavljuju u više kategorija, ali time ne dolazi do dupliciranja odredišta na popisu. Na slici 5.9 su prikazana turistička odredišta kada je odabrana kategorija „građevine“ na lijevoj strani slike, a na desnoj strani su prikazana preporučena turistička odredišta kada je odabrana samo kategorija „povijest“. Iz slika se može zaključiti da čak tri turistička odredišta su zajednička tim dvjema kategorijama.

Odabirom više od jedne kategorije korisniku se prikazuju odredišta kao na slici 5.10. Izmjenom odabranih kategorija tako da nije odabrana nijedna kategorija iz prve grupe odabranih kategorija mijenjaju se i preporučena turistička odredišta. Na lijevom dijelu slike 5.10 odabrane su kategorije: aktualno, umjetnost i sport, a na desnom dijelu slike 5.10 odabrane kategorije su: eksperimenti, spomenici i umjetnost. Može se primijetiti da je većina preporučenih odredišta različita i time se može zaključiti da preporučivanje radi prema očekivanju.



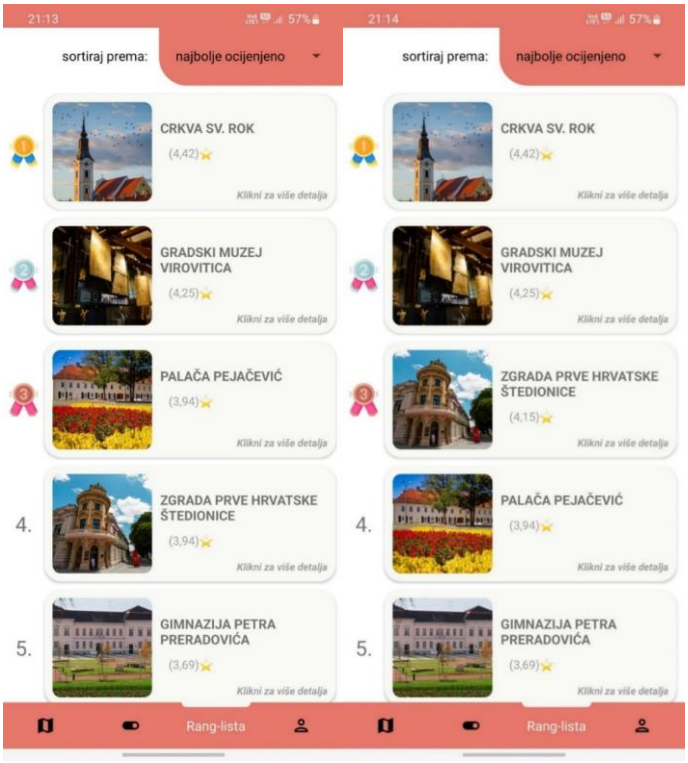
Slika 5.9 Prikaz preporučenih odredišta za kategoriju "građevine" lijevo i „povijest“ desno



Slika 5.10 Preporučena turistička odredišta prve grupe kategorija lijevo i druge grupe kategorija desno

5.2.3 Korisnički slučaj – Utjecaj korisničkih ocjena na rangiranje odredišta

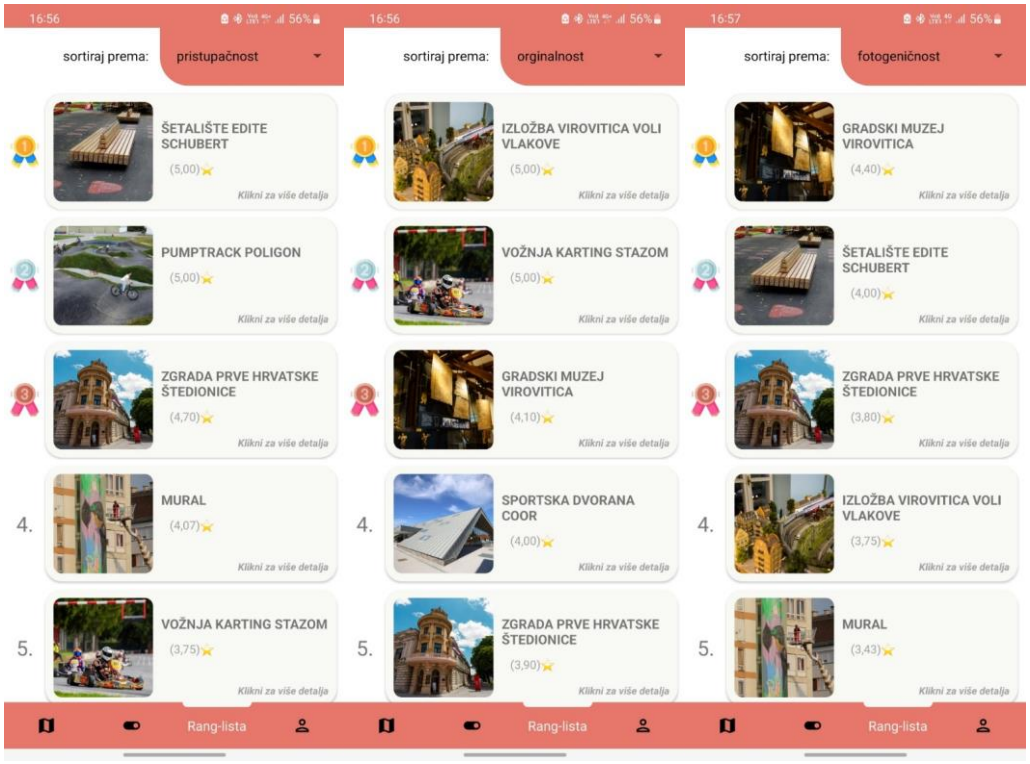
Korisnik svakom svojom ocjenom pridonosi u poboljšanju preporuka drugim korisnicima. Na lijevom dijelu slike 5.11 vidi se prosječna ocjena 3,94 i pozicija na rang-listi zgrade Prve hrvatske štedionice koja je na četvrtom mjestu. Nakon što je korisnik ocijenio promatrano turističko odredište, na desnom dijelu slike 5.11 vidi se promjena prosječne ocjene koja sada iznosi 4,15 i promjena pozicije na treće mjesto na rang-listi.



Slika 5.11 Prikaz promjene prosječne ocjene i pozicije zgrade Prve hrvatske štedionice

5.2.4 Korisnički slučaj – Odabir filtra

Zadani filtar koji se primjenjuje na listu preporučenih turističkih odredišta je „najbolje ocijenjeno“. Odabir filtra koji je od interesa korisniku je iznimno bitan jer mu se time nudi personaliziranija preporuka. Na slici 5.12 je prikaz turističkih odredišta koja pripadaju kategorijama: aktualno, umjetnost i sport. Primijenjeni su filtri pristupačnost, originalnost i fotogeničnost na ista odredišta kako bi se vidjelo da promjena filtra ima stvaran utjecaj u preporučivanju i da se time mijenja poredak preporučenih turističkih odredišta. Zato je bitno da korisnik odabere onaj filtar za koji smatra da najbolje opisuje ono što on sam traži u turističkim odredištima, na primjer, ako korisnik želi da su odredišta vrlo fotogenična onda treba izabrati filtar „fotogeničnost“.



Slika 5.12 Promjena pozicije odredišta ovisno o filtru

5.3 Analiza rezultata ispitivanja

Nakon provedenog ispitivanja, svaki od ključnih korisničkih slučajeva je imao očekivani ishod. Korisniku koji ne odabere nijednu kategoriju neće se preporučivati turistička odredišta, već će biti usmjeren da odabere preferirane kategorije i time popuni svoj korisnički profil, a korisniku koji odabere jednu ili više kategorija preporučiti će se odgovarajuća turistička odredišta, bez dupliciranja odredišta koja spadaju u više od jedne kategorije. Stvarni utjecaj svakog korisnika može se vidjeti u trećem korisničkom slučaju koji je dokaz da je kolaborativno filtriranje funkcionalno, gdje se već samo jednom ocjenom jednog kriterija mijenja pozicija tog turističkog odredišta na rang-listi zajedno s njegovom prosječnom ocjenom. Četvrti korisnički slučaj potvrđuje da primjenom različitih filtara korisnik dobiva različiti poredak preporučenih turističkih odredišta. Ovi korisnički slučajevi potvrđuju da aplikacija za višekriterijsko preporučivanje turističkih odredišta preporučuje odredišta po više kriterija kao što su to kategorije odredišta, sklonost korisnika, odabir filtra i ocjene korisnika.

6. ZAKLJUČAK

Pametni telefoni su neophodni u današnjem društvu i zato su izvrsna sredstva za promidžbu turizma nekog mjesta ili države. Izradom i promoviranjem aplikacija, turizam tog mjesta može se plasirati na visoko mjesto. Cilj ovog završnog rada bio je izraditi aplikaciju koja bi, korisniku koji posjećuje grad Viroviticu, mogla preporučiti turistička odredišta na temelju njegove sklonosti pomoću višekriterijskog ocjenjivanja i kolaborativnog filtriranja.

Korisniku je omogućena izrada korisničkog računa unutar aplikacije koju kasnije koristi za prijavu, a postoji i opcija u slučaju zaboravljene zaporke. Odabirom kategorija koje ga zanimaju korisnik dobiva listu preporučenih turističkih odredišta koja su sortirana po najvišim ocjenama, a on sam odabire filtar prema kojem će mu se odredišta rangirati. Svako odredište sadrži kratak opis i bitne informacije uz slike i gumb za pronalazak odredišta na karti, a od korisnika bi bilo poželjno da ocjeni posjećeno odredište po svim ponuđenim parametrima.

Nakon provedenog ispitivanja aplikacije za četiri ključna različita slučaja korištenja može se zaključiti da je aplikacija uspješno izrađena i ispunjava sve postavljene uvjete. Za daljnji razvoj aplikacije bilo bi dobro omogućiti veću personalizaciju profila korisnika kao što je dodavanje profilne slike, omogućiti dodavanje komentara na svako odredište i dodati još jedno sučelje koje bi korisnicima dalo mogućnost da podnesu zahtjev za dodavanje još nekog turističkog odredišta.

LITERATURA

[1] W. Hunziker, K. Krapf, „Fremdenverkehr“ [Tourism]. Handbuch der Schweizerischen Volkswirtschaft (Bern: Benteli-Verlag), Schweizerische Gesellschaft fur Statistik und Volkswirtschaft, 1955.

[2] Turizam, https://www.enciklopedija.hr/natuknica.aspx?id=62763, pristupljeno 22.08.2023.

[3] K. Meehan, T. Lunney, K. Curran, A. Mccaughey, Context-Aware Intelligent Recommendation System for Tourism, In the Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications, San Diego, CA, USA, 18-22 Ožujak, 2013, pp. 328.-331.

[4] K. Meehan, T. Lunney, K. Curran, A. McCaughey, A Social Media Based Tourist Information System, In the Proceedings of the International Conference on Tourism and Events: Opportunities, Impacts and Change, Belfast, Northern Ireland, 20-22 Lipanj, 2012, pp. 1-7

[5] F. Clarizia, F. Colace, M. De Santo, A Context-Aware Chatbot for tourist destinations, 15th International Conference on Signal-Image Technology & Internet-Based Systems, Sorrento, Italy, 26-29 Listopad, 2019, pp. 348-354

[6] W. Supanich, S. Kulkarineetham, Personalized Tourist Attraction Recommendation System Using Collaborative Filtering on Tourist Preferences, 19th International Joint Conference on Computer Science and Software Engineering, Bangkok, Thailand, 22-25 Lipanj, 2022, pp 1-6

[7] P. Thiengburanathum, S. Cang, H. Yu, A Decision Tree based Recommendation System for Tourists, Proceedings of the 21th International Conference on Automation & Computing, Glasgow, UK, 11-12 Rujan, 2015, pp. 1-7

[8] I. Sommerville, Software Engineering, Pearson Education Limited, 10. izdanje, 2016

[9] Q. Shambour and J. Lu, Integrating Multi-Criteria Collaborative Filtering and Trust filtering for personalized Recommender Systems, 2011 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM), Paris, France, 11-15 Travanj, 2011., pp. 44-51.

[10] About Canva, https://www.canva.com/about/, pristupljeno 31.03.2023.

[11] Meet Android Studio, https://developer.android.com/studio/intro, pristupljeno 31.03.2023.

[12] D. Jemerov, S. Isakova, Kotlin in Action, Manning Publications, 2017., str. 1.

[13] Introduction to XML, https://www.w3schools.com/xml/xml_whatis.asp, pristupljeno 01.04.2023.

[14] Firebase, https://firebase.google.com/, pristupljeno 01.04.2023.

POPIS SLIKA

Slika 2.1 Prikaz sučelja aplikacije Sygic Travel Maps Offline & Trip Planner 5
Slika 2.2 Prikaz sučelja aplikacije Vienna Travel Guide 6
Slika 2.3 Prikaz sučelja aplikacije Tripadvisor: Plan & Book Trips 7
Slika 3.1 Dijagram tijeka mobilne Android aplikacije za višekriterijsko preporučivanje turističkih odredišta 12
Slika 3.2 Prikaz kriterija i podkriterija za višekriterijsko preporučivanje 14
Slika 3.3 Pseudo-kod algoritma za višekriterijsko preporučivanje turističkih odredišta 15
Slika 4.1 Logotip aplikacije 16
Slika 4.2 Ideja izgleda aplikacije 17
Slika 4.3 Programski kod metode register 19
Slika 4.4 Programski kod metode login 20
Slika 4.5 Isječak programskog koda za prikazivanje markera turističkih odredišta na karti 21
Slika 4.6 Prikaz klase MapMarker 21
Slika 4.7 Programski kod za spremanje odabranih čipova 22
Slika 4.8 Objekt s listom odabranih čipova s pripadnim metodama 23
Slika 4.9 Prikaz klase LocationData 24
Slika 4.10 Programski kod metode updateArrayWithChips 24
Slika 4.11 Programski kod metode performSort 25
Slika 4.12 Programski kod metode showList 25
Slika 4.13 Programski kod metode updateRating 26
Slika 4.14 Isječak programskog koda kartice s detaljima korisnika 27
Slika 4.15 Baza podataka korisnika 28
Slika 4.16 Baza podataka odredišta 28
Slika 4.17 Registrirani korisnici u sustavu 29
Slika 4.18 Programski kod za rad s kartom 30
Slika 5.1 Sučelje za prijavu 31
Slika 5.2 Sučelje za registraciju 32
Slika 5.3 Sučelje za odabir kategorija 33
Slika 5.4 Prikaz karte 34
Slika 5.5 Sučelje s preporučenim odredištima 34
Slika 5.6 Sučelje detalja nekog odredišta 35

Slika 5.7 Sučelje korisničkog profila 36
Slika 5.8 Prikaz zaslona preporučenih odredišta kada nije odabrana nijedna kategorija 37
Slika 5.9 Prikaz preporučenih odredišta za kategoriju "građevine" lijevo i „povijest“ desno 38
Slika 5.10 Preporučena turistička odredišta prve grupe kategorija lijevo i druge grupe kategorija desno 38
Slika 5.11 Prikaz promjene prosječne ocjene i pozicije zgrade Prve hrvatske štedionice 39
Slika 5.12 Promjena pozicije odredišta ovisno o filtru 40

SAŽETAK

U ovom završnom radu, koristeći Android Studio uz upotrebu programskog jezika Kotlin i XML, izrađena je mobilna Android aplikacija za višekriterijsko preporučivanje turističkih odredišta. Za izradu korisničkog računa i spremanje podataka o korisniku koristi se Firebase Autentification, dok se Firebase Cloud koristi za spremanje osobnih sklonosti korisnika i ocjena koje on dodjeljuje turističkim odredištima te svih informacija o odredištima. Višekriterijsko preporučivanje uključuje pružanje personaliziranih preporuka korisniku temeljenih na odabranim kategorijama odredišta, a svako odredište pripada barem jednoj kategoriji. Preporučena odredišta rangiraju se pomoću izabranog filtra, a ocjene po kojima se rangiraju temelje se na povratnim informacijama drugih korisnika. Ocjene se zatim računaju kao prosjek svih ocjena koje su dodijeljene, uz primjenu tehnike kolaborativnog filtriranja. Ispitivanjem četiri različita slučaja korištenja potvrđena je ispravnost funkcionalnosti aplikacije te njezina sposobnost pružanja kvalitetnih preporuka turističkih odredišta.

Ključne riječi:, kolaborativno filtriranje, mobilna Android aplikacija, turistička odredišta, višekriterijsko preporučivanje.

ABSTRACT

Title: Mobile Android application for multi-criteria recommendation of tourist destinations

In this final thesis, a mobile Android application for multi-criteria recommendation of tourist destinations was developed using Android Studio, utilizing the Kotlin programming language and XML. Firebase Authentication is used for creating user accounts and storing user-related data, while Firebase Cloud is employed for storing users' personal preferences, ratings assigned to tourist destinations, and all destination-related information. The multi-criteria recommendation involves providing personalized suggestions to the user based on selected destination categories, with each destination belonging to at least one category. The recommended destinations are ranked using a chosen filter, and the ratings used for ranking are derived from feedback given by other users. These ratings are then computed as the average of all assigned ratings, applying collaborative filtering techniques. The functionality of the application was validated through testing four different use cases, confirming its correctness and its capability to offer high-quality recommendations of tourist destinations.

Keywords: collaborative filtering, mobile Android application, tourist destinations, multi-criteria recommending.

ŽIVOTOPIS

Klara-Iva Gerić rođena je 1. rujna 2000. godine u Virovitici. Nakon Osnovne škole Vladimira Nazora u Virovitici upisuje Gimnaziju Petra Preradovića Virovitica 2015. godine. Opći smjer gimnazije završava 2019. godine kada upisuje FERIT Osijek. Također, kao višegodišnja članica likovnog kluba Nikola Trick ima mnogo umjetničkih radova, a voli se baviti i fotografiranjem s kojima ima nekoliko osvojenih nagrada. Sudjelovala je na mnogim natjecanjima u kategorijama matematike, robotike, GLOBE i slično. Govori dva strana jezika, engleski i njemački, te posjeduje vozačku dozvolu B kategorije.

PRILOZI

Prilog 1. Završni rad „Mobilna android aplikacija za višekriterijsko preporučivanje turističkih odredišta“ u *.docx* formatu.

Prilog 2. Završni rad „Mobilna android aplikacija za višekriterijsko preporučivanje turističkih odredišta“ u *.pdf* formatu.

Prilog 3. Programsko rješenje izrađene mobilne aplikacije