

Razvoj mobilne aplikacije za pronalazak atraktivnih turističkih lokacija

Močibob, Eros

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:740993>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-13**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

EROS MOČIBOB

**RAZVOJ MOBILNE APLIKACIJE ZA PRONALAZAK ATRAKTIVNIH TURISTIČKIH
LOKACIJA**

Završni rad

Pula, rujan, 2020.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

EROS MOČIBOB

RAZVOJ MOBILNE APLIKACIJE ZA PRONALAZAK ATRAKTIVNIH TURISTIČKIH LOKACIJA

Završni rad

JMBAG: 0303061991 , izvanredni student

Studijski smjer: Informatika

Predmet: Informacijska tehnologija i društvo

Mentor: Mario Radovan

Komentor: Nikola Tanković

Pula, rujan, 2020.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani EROS MOČIBOB, kandidat za prvostupnika INFORMATIKE ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega 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 Završnog rada nije napisan na nedozvoljeni 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

Eros Močibob

U Puli, 7.09.2020



IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, EROS MOČIBOB dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj Završni rad pod nazivom RAZVOJ MOBILNE APLIKACIJE ZA PRONAŠALAK ATRAKTIVNIH TURISTIČKIH LOKACIJA 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, 7.09.2020

Potpis

Eros Močibob

Sadržaj

Uvod.....	1
Što je Flutter.....	2
Prednosti Fluttera.....	4
Usporedba sa drugim alatima za izradu cross-platform mobilnih aplikacija.....	5
Flutter i Xamarin.....	7
Flutter i React Native.....	8
Flutter i Ionic.....	9
Baza podataka aplikacije.....	10
Firestore.....	11
Aplikacija.....	11
Opis aplikacije.....	11
Dijelovi aplikacije.....	12
Arhitektura aplikacije.....	22
Zaključak.....	23
Literatura.....	25
Popis slika.....	27
Sažetak.....	28

Uvod

Cross-platform softver je softver koji se može pokretati na više platformi (npr. na Windowsu, Linuxu i macOSu). Možemo ga podijeliti u dvije kategorije: softver koji se izrađuje ili kompajlira za svaku platformu posebno i softver koji se direktno može pokretati na više platformi bez dodatnih koraka [1]. U nastavku rada fokus je na drugoj kategoriji.

Pomoću Fluttera izrađena je cross-platform mobilna aplikacija kojom korisnici mogu dijeliti fotografije omiljenih lokacija. Aplikacija, između ostalog, sadrži formu za registraciju, interaktivne mape, automatsku detekciju GPS lokacije, označavanje omiljenih lokacija unutar aplikacija (tzv. lajkanje), itd.

Prije detaljnijeg opisa prethodno navedene aplikacije, daje se više informacija o Flutteru te nabrajaju se njegove prednosti i mane. Flutter se u radu također uspoređuje sa tri druge popularne tehnologije za izradu cross-platform aplikacija.

Što je Flutter

Flutter je skup alata za izradu korisničkih sučelja koji omogućuje izradu mobilnih, web i desktop aplikacija iz istog koda. Google stoji iza Fluttera te ga aktivno razvija. Flutter je besplatan i otvorenog koda (engl. open source) tako da bilo tko može pridonijeti njegovom razvoju. Glavna prednost Fluttera je što smanjuje vrijeme razvoja te smanjuje kompleksnost procesa pisanja aplikacija za više platformi (npr. pisanje iste aplikacije za mobilne uređaje i za web). Dizajneri aplikacija također mogu koristiti Flutter za prototipiranje. [2]

Pod Flutter spadaju [2]:

- Optimizirani 2D engine za crtanje tj. prikazivanje aplikacije
- Moderni framework inspiriran Reactom
- Razni widgeti i teme u više stilova (npr. Googleov Material Design i iOS stilovi)
- Programsko sučelje (engl. application programming interface, skraćeno API) za testove
- Mogućnost spajanja na sustave uređaja (npr. datotečni sustav) i SDK-ove treće strane (engl. 3rd-party SDKs)
- Alat za pokretanje testova bez grafičkog sučelja (Window, Linux i Mac)
- Dodatni alati za testiranje, debugiranje, profiliranje, izradu i kompajliranje aplikacija

Flutter je relativno pristupačan i lak za naučiti. Svi programeri koji poznaju koncepte objektno-orientiranih jezika (klase, metode, varijable, itd.) i koncepte imperativnog programiranja (petlje, uvjetno grananje, itd.) ne bi trebali imati problema sa učenjem i korištenjem Fluttera. Valja napomenuti da se Flutter može koristiti na Windowsu, Linuxu i Macu. [2]

Jedna of glavnih primjena Fluttera je izrada 2D mobilnih aplikacija za Android i iOS. Flutter se naravno može koristiti i za izradu raznih interaktivnih web i desktop aplikacija. U trenutku pisanja ovog rada podrška za web i desktop aplikacije je u testnoj fazi. Flutter dopušta veliku slobodu kod definiranja izgleda aplikacije. Moguće je izraditi aplikacije sa potpuno generičkim izgledom (npr. izgled native Android aplikacije) ili aplikacije sa personaliziranim dizajnom. Personalizirani izgled je bitan kada se gradi aplikacija za nekog

klijenta koji želi grafičko sučelje konzistentno sa vizualnim identitetom brenda, npr. poznata banka ili lanac brze hrane. Pristup kameri, geolokacija, pristup mreži, pristup datotečnom sustavu, SDK-ovi trećih strana su samo neke od funkcionalnosti koje Flutter pruža programerima. [2]

Widgeti su srž Fluttera. Na widgete možemo gledati kao na sastavne elemente korisničkog sučelja. Flutter aplikacije se grade „slaganjem” widgeta. Pojedini widget može predstavljati strukturalni element (npr. gumb ili meni), stilski element (npr. font ili paletu boja), element za raspored u prostoru (npr. margine), itd. Flutter ne koristi native widgete (tj. widgete koje definira mobilni operacijski sustav) nego vlastite. Jedna mana tog pristupa je da se novi widgeti moraju dodati Flutteru svaki put kada npr. Android operacijski sustav dobije nove native widgete. U tom slučaju Googleov Flutter tim ili članovi šire Flutter zajednice dodaju widgete u Flutter. Naravno, programeri mogu definirati vlastite widgete. [3]

Flutter je relativno novi komad tehnologije. Verzija 1.0 izdana je 4. prosinca 2018. Neki dijelovi Fluttera, kao npr. puna podrška za desktop aplikacije, nisu još pristigli u zadnju stabilnu verziju. Unatoč tome, već je tisuće Flutter aplikacija izdano i instalirano na stotine milijuna uređaja [2]. Google, eBay, BMW, Groupon, Nubank i Tencent su samo neke od kompanija koje koriste Flutter [4].

Kod pisanja Flutter aplikacija koristi se Dart programski jezik. Flutter se na Windowsu, macOS-u i Linuxu izvršava unutar Dart virtualne mašine [5]. Dart, kojeg razvija Google, ima sintaksu sličnu C programskom jeziku i posebno je optimiziran za pisanje aplikacija za više platformi [6].

Prednosti Fluttera

Neke od prednosti u nastavku odnose se i generalno na razvoj cross-platform mobilnih aplikacija.

Spomenuto je u prvom poglavlju kako Flutter smanjuje vrijeme i kompleksnost razvoja aplikacija pošto se iz istog Flutter koda može izvesti softver za više platformi. Flutter može pomoći menadžerima kojima vode timove za razvoj softvera time što je za razvoj više aplikacija dovoljan samo jedan tim. Korištenjem Fluttera nije potrebno imati posebne resurse za pojedine platforme (npr. jedan tim/resurs za desktop aplikaciju i jedan tim/resurs za Android mobilnu aplikaciju). Osim što takav cross-platform pristup smanjuje potrebu za ljudskim resursima, također povećava brzinu dodavanja novih funkcionalnosti softverskom proizvodu. Brže je dodati funkcionalnost jednom u kodu nego npr. u tri različita koda za tri različite platforme. Korištenjem Fluttera postiže se konzistentnost na više platformi pošto se koristi jedinstveni kod. Korisnici na više platformi imaju pristup istom korisničkom sučelju i istom skupu funkcionalnosti. Naravno, sve navedene prednosti rezultiraju i manjim troškovima razvoja aplikacija: manji broj potrebnih ljudskih resursa za razvoj aplikacije, održavanje samo jednog koda te brže vrijeme razvoja tj. brže plasiranje novih verzija aplikacije na tržište. Sa podrškom za više platformi automatski se dobiva pristup većem broju potencijalnih korisnika na različitim uređajima. [2]

Jedna od glavnih razlika Fluttera i konkurentnih rješenja za izradu mobilnih aplikacija je ta šta Flutter koristi vlastiti optimizirani engine za crtanje tj. prikazivanje elemenata aplikacije. Takav pristup dopušta više performanse u odnosu na npr. korištenje WebViewa¹. Tanak sloj C/C++ koda omogućuje bolje performanse. Još jedna prednost Fluttera je ta što koristi Dart programski jezik za implementaciju većinu svog sustava (animacije, pokreti na mobilnim uređajima, grafički elementi, itd). Korištenje Darta smanjuje potrebno programersko znanje za manipulaciju većine sustava (npr. pokreti na mobilnim uređajima). [2]

Flutter podržava tzv. vruće učitavanje (engl. hot reloading). Pomoću vrućeg učitavanja programer može u stvarnom vremenu vidjeti promjene u kodu bez da se cijela aplikacija

¹ WebView je ugrađeni (engl. embedded) Internet pretraživač kojeg aplikacije mogu koristiti za prikaz web sadržaja. [7]

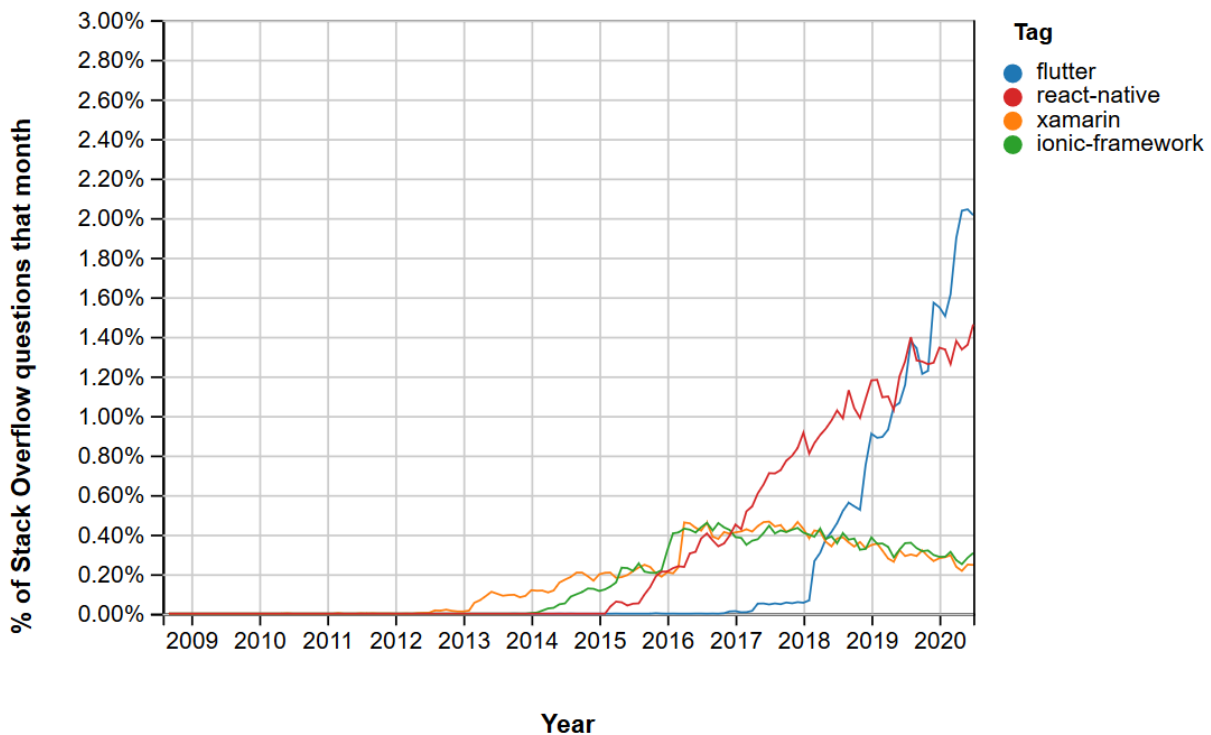
ponovno učitava. Učitava se samo dio koji je bio promijenjen. Vruće učitavanje je posebno korisno u razvoju većih aplikacija koje se sporo kompajliraju. [2]

Spomenimo i da Flutter ima kvalitetnu dokumentaciju te da Dart programski jezik olakšava pisanje aplikacija pošto asinkroni kod možemo pisati u stilu sinkronog koda. [8]

Usporedba sa drugim alatima za izradu cross-platform mobilnih aplikacija

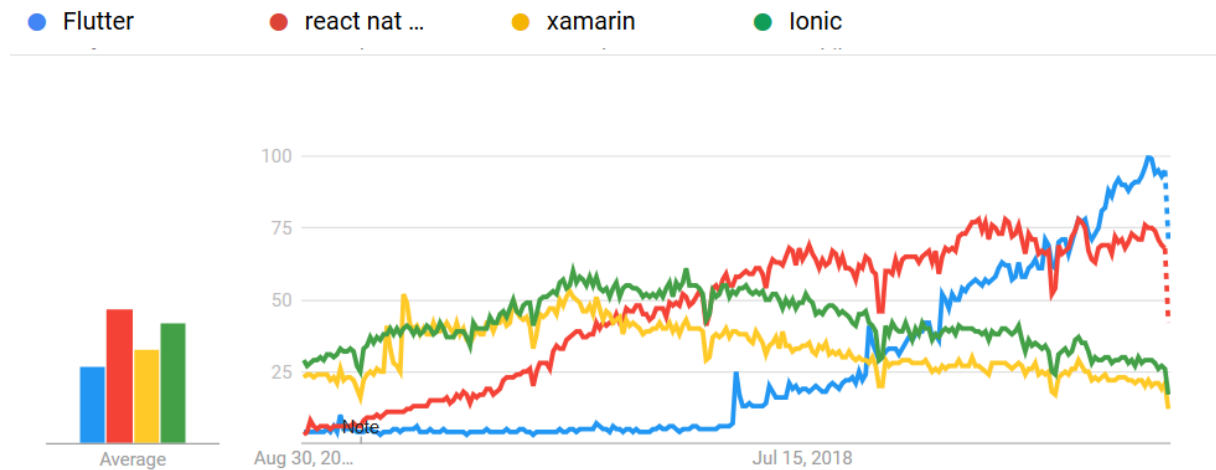
U sljedećim poglavljima Flutter se uspoređuje sa konkurentnim softverom za izradu cross-platform aplikacija. Fokus je tehnologijama koje su primarno namijenjene za izradu mobilnih aplikacija. Odabrane su sljedeće tehnologije koje su dovoljno zrele i popularne da budu ozbiljna konkurencija Flutteru: Xamarin, React Native i Ionic.

Dolje se navodi graf popularnosti prethodno navedenih tehnologija na stranici StackOverflow [9]. Iz grafa možemo vidjeti da je Flutter imao nagli porast popularnosti unazad nekoliko godina i da danas vodi po broju mjesečnih pitanja. Iako je StackOverflow jako popularan među programerima, statistike nisu nikakva službena metrika korištenja određene tehnologije u praksi. No trendovi su očiti: Flutter i React Native su značajno popularniji od Xamarina i Ionica.



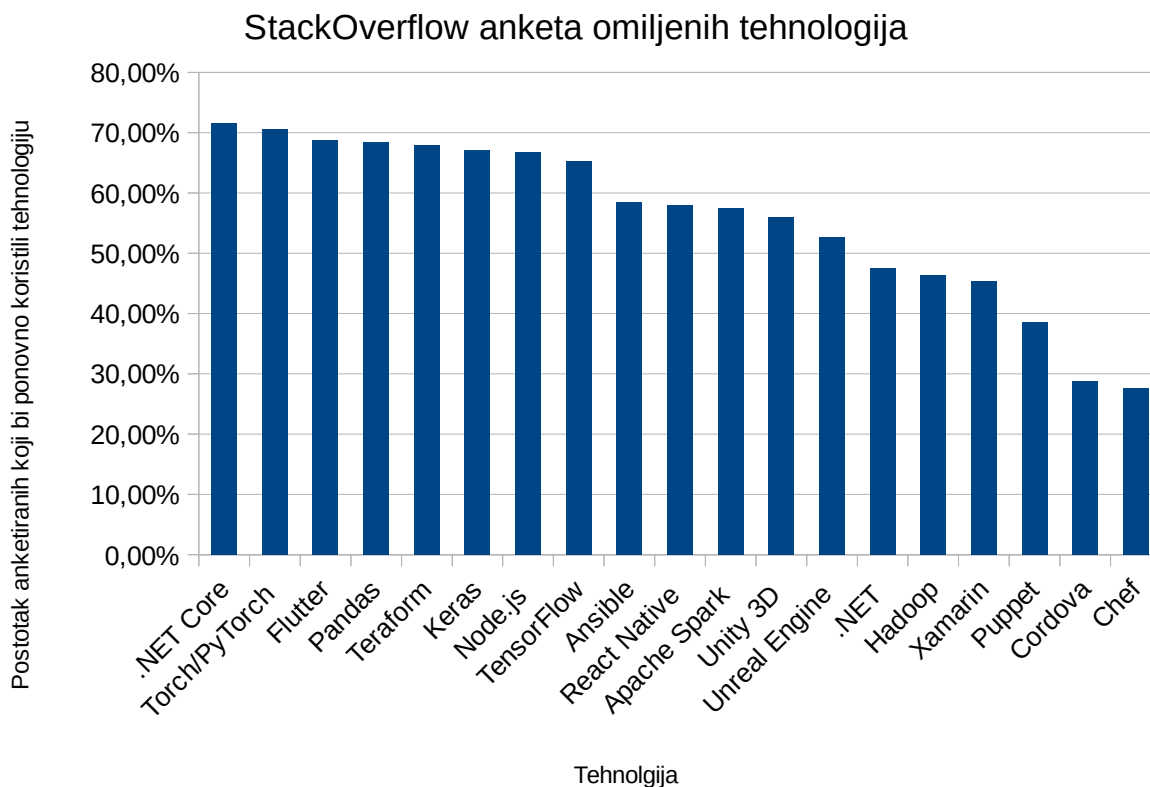
Slika 1: Postotak mjesečnih pitanja na StackOverflowu po tehnologiji (vertikalna os) kroz vrijeme (horizontalna os).

Slične rezultate daje i Google Trends. Flutter je danas najpopularniji a za njim slijedi React Native. U donjoj slici vidi se relativna popularnost sljedećih pojmova na Google tražilici: Flutter (softver), React Native, Xamarin i Ionic (softver). Graf sadrži globalne podatke za zadnjih pet godina. [10]



Slika 2: Relativna popularnost pojmova na Google tražilici (vertikalna os) kroz vrijeme (horizontalna os). Lijevo od glavnog grafa nalazi se prosječna popularnost.

StackOverflow vrši godišnje anketiranje u kojem sakupljaju podatke o korištenju raznih programskih jezika i tehnologija. Sljedeći graf sadrži podatke za najomiljenije tehnologije 2020. godine [11]. Flutter je i u ovom slučaju popularniji od React Nativea. Xamarin i Ionic nisu sakupili dovoljno glasova da zasluže mjesto na grafu.



Slika 3: Rezultati StackOverflow ankete omiljenih tehnologija. Graf prikazuje postotak anketiranih koji su koristili određenu tehnologiju (vertikalna os) i htjeli bi je koristiti ponovno.

Flutter i Xamarin

I Flutter i Xamarin se koriste za izradu cross-platform mobilnih aplikacija iz jednog koda. Tvrtka Xamarin (čiji softver nosi isto ime) osnovana je 2011. sa fokusom na softver za razvoj mobilnih aplikacija. Xamarin je najstariji framework za razvoj cross-platform aplikacija. Godine 2016. Microsoft je kupio Xamarin i otvorio kod frameworka za javnost. Xamarin SDK je danas dio Microsoftove .NET platforme. [12]

Xamarin koristi C# programski jezik koji je puno popularniji i rašireniji od Darta. No uz C# nekada se koriste i nativni jezici za određene izmjene grafičkog sučelja. To nije slučaj sa Flutterom. [12]

Flutter tj. Dart aplikacije mogu se pisati u bilo kojem integriranom razvojnom okruženju, tzv. IDE-u (engl. integrated development environment). Sa druge strane, Xamarin

aplikacije se mogu razvijati samo u Microsoftovom Visual Studio IDE-u. To može stvoriti probleme kod programera koji su navikli na korištenje određenih alata. Unatoč tome, Visual Studio je zreo softver sa mnogo integriranih alata koji pomažu u razvoju aplikacija. Neki od tih alata omogućuju brže refaktoriranje, debugiranje i pregled aplikacije tijekom pisanja njezina koda. Xamarin ima lošiju podršku za vruće učitavanje. [12]

Xamarinova dokumentacije je opsežnija i detaljnija od dokumentacije za Flutter. Xamarinom se uz mobilne aplikacije za Android i iOS mogu razvijati i aplikacije za Windows, macOS, tvOS i watchOS. Prisjetimo se, Flutter uz mobilne aplikacije djelimično podržava i razvoj web aplikacija. Pošto Xamarin postoji puno duže od Fluttera, tehnička zajednica oko Xamarina je veća od Flutterove zajednice. Unatoč tome, Flutter rangira više u anketama zadovoljstva. [12]

Za kraj spomenimo da je Xamarin besplatan samo za individualce i manje timove. Flutter je potpuno besplatan. [12]

Flutter i React Native

React Native je jedna od najpopularnijih tehnologija za razvoj cross-platform mobilnih aplikacija danas. Facebook je izdao React Native u javnost 2015. i održava ga od tada. O samoj popularnosti React Nativea govori i činjenica da njegov javni repozitoriji ima drugi najviši broj suradnika na cijelom GitHubu. Kao što samo ime kaže, React Native se temelji na React frameworku. Neke od tvrtki koje koriste React Native su: Facebook, Instagram, Salesforce, Tesla, Bloomberg, Discord i Walmart. [13]

Jedna od najvećih prednosti React Nativea pred Flutterom je ta što se React Native aplikacije pišu u JavaScriptu. JavaScript je jedan od najpopularnijih i najkorištenijih programskih jezika današnjice. Kompanije koje razmišljaju o razvoju cross-platform mobilne aplikacije moraju uzeti u obzir da će lakše naći JavaScript programera nego Dart programera. [14]

Za razliku od Fluttera, React Native koristi nativne komponente za korisničko sučelje. To znači da će komponente React Native aplikacije biti automatski ažurirane nakon što se

nativne komponente mobilnog operacijskog sustava ažuriraju. Treba uzeti u obzir da automatsko ažuriranje može u rijetkim slučajevima slomiti korisničko sučelje. Flutter nema takvih problema. Postoje React Native biblioteke trećih strana (engl. 3rd-party) koje omogućavaju izradu konzistentnog korisničkog sučelje za više platformi. React Native Paper je primjer jedne takve biblioteke koje koristi Googleov Material Design dizajn. [15]

Prisjetimo se kako uz mobilne platforme Flutter namjerava u budućnosti podržavati razvoj aplikacija za desktop i web. Za usporedbu, postoje biblioteke za React Native kojima se mogu razvijati aplikacije za web i Windows 10. [15]

React Native ima svoju verziju Flutterovog vrućeg učitavanja koja se zove hot refresh. U srži, funkcionalnost je ista: promjene u kodu rezultiraju promjenom aplikacije u stvarnom vremenu. [15]

Aplikacije izrađene u React Nativeu i Flutteru su u pravilu veće nego aplikacije izrađene u nativnim tehnologijama. Što se performansi tiče, React Native je u srži omot oko nativnih metoda tako da zahtjeva „most” koji će prevoditi React Native API pozive u nativni kod. To može stvoriti problem kada treba prevesti velik broj poziva. Flutter zahtjeva manje takvih poziva što rezultira boljim performansama. [15]

Valja spomenuti da React Native ima veću tehničku zajednicu i više materijala za učenje samim time što je stariji od Fluttera. Također, React Native ima veći broj biblioteka trećih strana. [15]

Flutter i Ionic

Pod Ionic spadaju Ionic framework i skup alata za izradu korisničkih sučelja. Ionic koristi standarde otvorenog weba te web tehnologije kao CSS, HTML i JavaScript za izradu desktop i mobilnih aplikacija. Prva verzija Ionica izdana je 2013. [16]

Jedna od većih prednosti Ionica u odnosu na Flutter je ta da Ionic u potpunosti podržava web kao platformu. Još jedna prednost je ta da se Ionic aplikacije mogu pisati JavaScriptom. Kao što je ranije spomenuto, JavaScript je puno rašireniji od Darta. Nadalje, moguće je koristiti React, Angular i Vue web frameworke unutar Ionica što dodatno olakšava prijelaz iz web u mobilni razvoj aplikacija velikom broju programera. [16]

Ionic aplikacije rade tako da se izvršavaju unutar nativnog kontejnera pomoću Cordove ili Capacitatora. Cordova i Capacitator omogućuju puni pristup nativnom sustavu uređaja. Za prikaz grafičkog sučelja na mobilnim uređajima Ionic koristi WebView. Na WebView možemo gledati kao na Internet pretraživač za kojeg korisnik ni ne zna da postoji. Ionic aplikacije na desktop uređajima se izvršavaju unutar kontejnera kao što je npr. Electron. Aplikacije se mogu i direktno prikazivati na bilo kojem mobilnom uređaju ili Internet pretraživaču kao Progressive Web App. [17]

Ni Ionic ni Flutter ne koriste ugrađene nativne komponente za prikaz korisničkog sučelja. Što se veličina aplikacija tiče, Ionic aplikacije su u pravilu manje. To je zato što Ionic koristi ugrađene mogućnosti postojećih Internet pretraživača. Flutter aplikacije, u drugu ruku, moraju sadržavati softver koji će pokretati i prikazivati aplikaciju. To vrijedi i za Flutter web aplikacije. Za kraj spomenimo i performanse; za aplikacije sa puno animacija Flutter nudi bolje performanse od Ionica. [17]

Baza podataka aplikacije

Aplikacija podržava registraciju i prijavu. Prijavljeni korisnici mogu dodati svoje fotografije u aplikaciji. U bazi se spremaju informacije o korisnicima i fotografijama. Kod izbora baze podataka postojala su dva izbora: koristiti SQL tj. relacijsku bazu (npr. Microsoft SQL Server, MySQL ili PostgreSQL) ili koristiti NoSQL tj. nerelacijsku bazu podataka (npr. MongoDB ili Firestore).

Postoji nekoliko ključnih razlika između SQL i NoSQL baza podataka. SQL baze koriste poseban jezik za izvršavanje upita (kao npr. SQL jezik) i imaju predefinirane sheme. NoSQL baze podataka, u drugu ruku, koriste dinamičke sheme za polustrukturirane i

nestrukturirane podatke (kao npr. JSON). SQL baze se skaliraju vertikalno što znači da za spremanje većih volumena podataka treba povećati performanse hardvera (npr. zamijeniti CPU ili dodati više RAM memorije). NoSQL baze podataka se skaliraju horizontalno. Kod horizontalnog skaliranja performanse sustava se povećavaju dodavanjem hardvera tj. računala postojećem skupu računala. Dok SQL baze koriste tablice, NoSQL baze koriste dokumente, ključ-vrijednost (engl. key-value) strukture, grafove i slične alternative tablicama. [18]

Iako NoSQL baze postaju sve popularnije, i dalje nisu popularne kao relacijske baze [19]. U praksi to znači da postoji manje primjera i materijala za učenje.

Firestore

Firestore NoSQL baza odabrana je kao baza podataka za aplikaciju iz nekoliko razloga. NoSQL baze podataka ne zahtijevaju predefinirane sheme što ih čine odličnim za prototipiranje, pogotovo kada unaprijed ne znamo finalnu strukturu podataka.

Firestore je relativno nova NoSQL baza koju je Google pustio u javnost 2017. Baza je posebno namijenjena za mobilne i web aplikacije. Firestore ima odličnu integraciju sa ostalim Googleovim proizvodima. Tako postoji i biblioteka za rad sa Firestorom u Flutteru. Jedna od prednosti Firestora je da se baze automatski skaliraju ovisno o volumenu zahtjeva. Firestore je serverless servis što znači da ne moramo brinuti o administriranju servera. Firestore ima besplatnu i plaćenu verziju. Besplatna verzija je dovoljna za učenje i prototipiranje. [20]

Aplikacija

Opis aplikacije

Aplikacija omogućava korisnicima dijeljenje fotografija turističkih ili generalno zanimljivih lokacija. Sadrži galeriju gdje korisnici mogu pregledavati sve podijeljene fotografije. Moguće je filtriranje galerije po kategorijama. Pojedine fotografije lokacija se mogu lajkati tj. dodati u favorite. Podijeljene fotografije tj. lokacije mogu se pretraživati putem

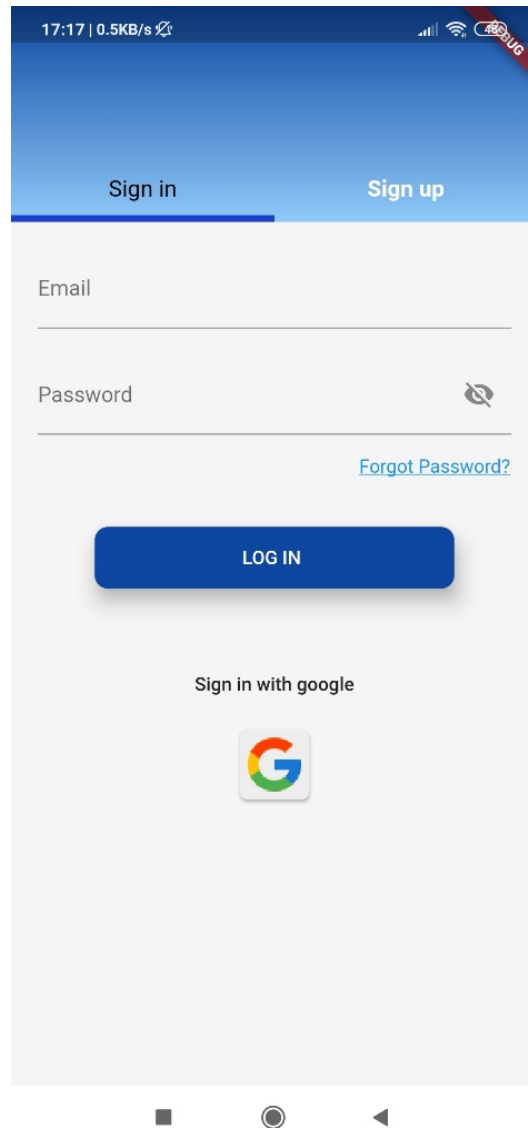
interaktivne mape (u slučaju da korisnika zanimaju interesantne lokacije za specifičnu regiju). Aplikacija ima nekoliko svojstava koja olakšavaju njeno korištenje kao npr. prijava preko postojećeg Google računa (nije potrebna registracija) i automatsko određivanje GPS koordinata fotografije preko trenutne lokacije uređaja. Aplikaciju bi primarno mogla zanimati turiste koji žele osobne preporuke, možda manje poznatih, mjesta za posjećivanje.

Kod aplikacije javno je dostupan na sljedećoj stranici:
github.com/erosmocibob/mobileApp_places.

Dijelovi aplikacije

Slike zaslona u ovom poglavlju snimljene su na Android mobilnom uređaju.

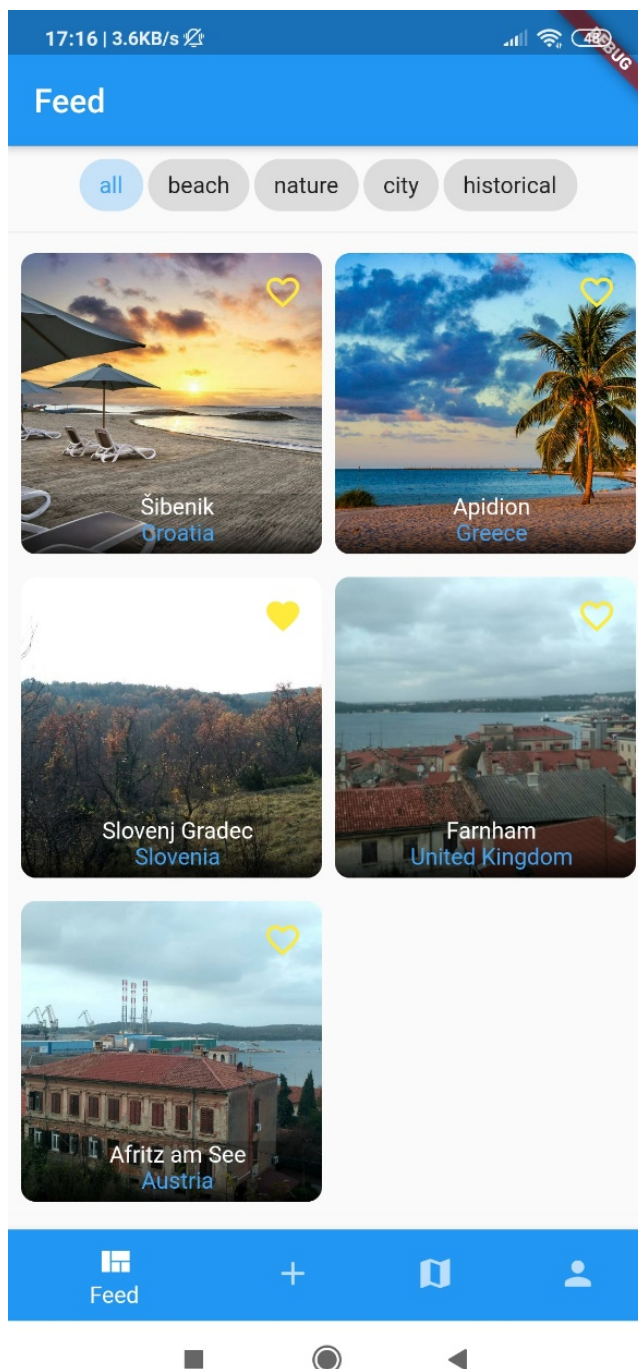
Kod prvog pokretanja aplikacije prikazuje se ekran za prijavu i registraciju. Prijava se obavlja u lijevoj a registracija u desnoj kartici (engl. tab). Za korisnike koje ne žele iz bilo kojeg razloga napraviti novi korisnički račun, prijava je moguća preko postojećeg Googleovog računa. U slučaju krive kombinacije korisničkog imena i lozinke prikazati će se poruka sa opisom greške. Aplikacija nudi i mogućnost ponovnog postavljanja zaboravljene lozinke („Forgot Password?” na donjoj slici).



Slika 4: Početni ekran aplikacije.

Nakon uspješne prijave prikazuje se „Feed” ekran sa galerijom fotografija raznih lokacija. Sve fotografije slikali su i podijelili korisnici aplikacije. Korisnik na ovom ekranu može filtrirati fotografije po kategorijama. Na primjer, mogu se prikazati samo fotografije plaža ili samo gradske fotografije. Odabrani filter će dobiti plavu boju koja ga ističe od ostalih neoznačenih (sivih) filtera. Korisnik može i lajkati bilo koju od prikazanih fotografija. Ikona punog srca znači da je korisnik lajkao fotografiju dok ikona srca bez ispune označava suprotno. Na donjoj slici vidimo i navigaciju na dnu ekrana koja je konzistentna kroz ostatak aplikacija. Konzistentnost je od velike važnosti kako bi se izbjegla dezorijentiranost korisnika. Korisnik zna u kojem je dijelu aplikacije zahvaljujući istaknutoj navigacijskoj ikoni

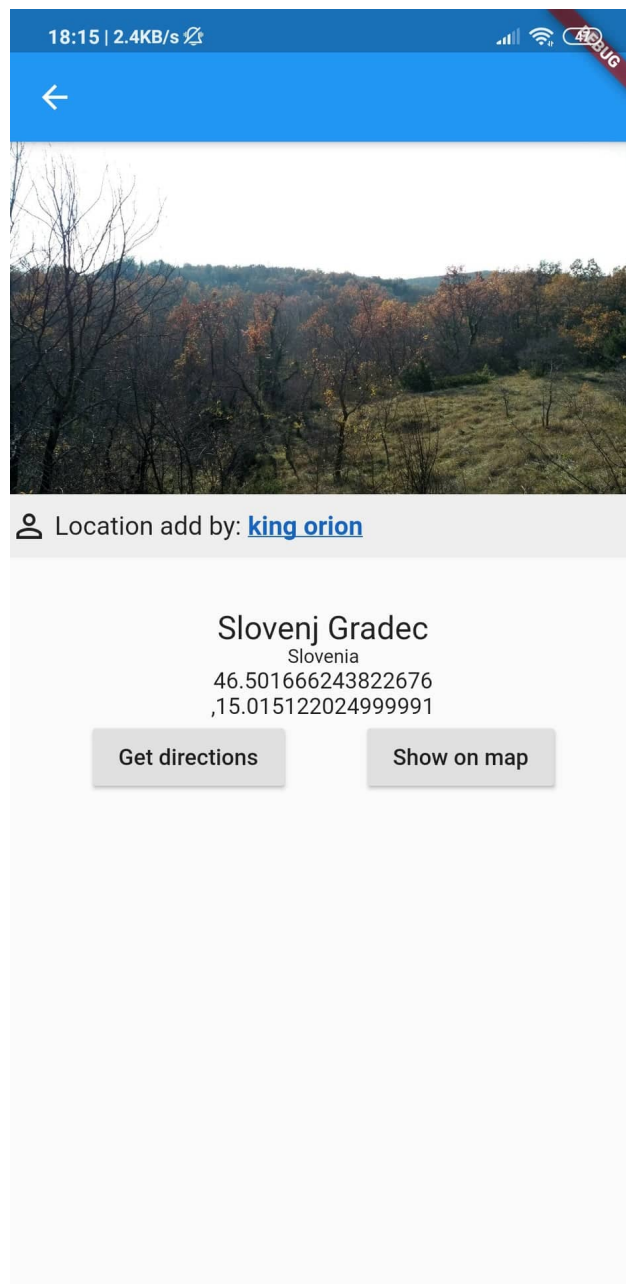
sa tekstualnim opisom (u donjem primjeru to je „Feed” ikona). Navigacijom se može u bilo kojem trenutku doći na bilo koji drugi bitni dio aplikacije.



Slika 5: Ekran sa galerijom.

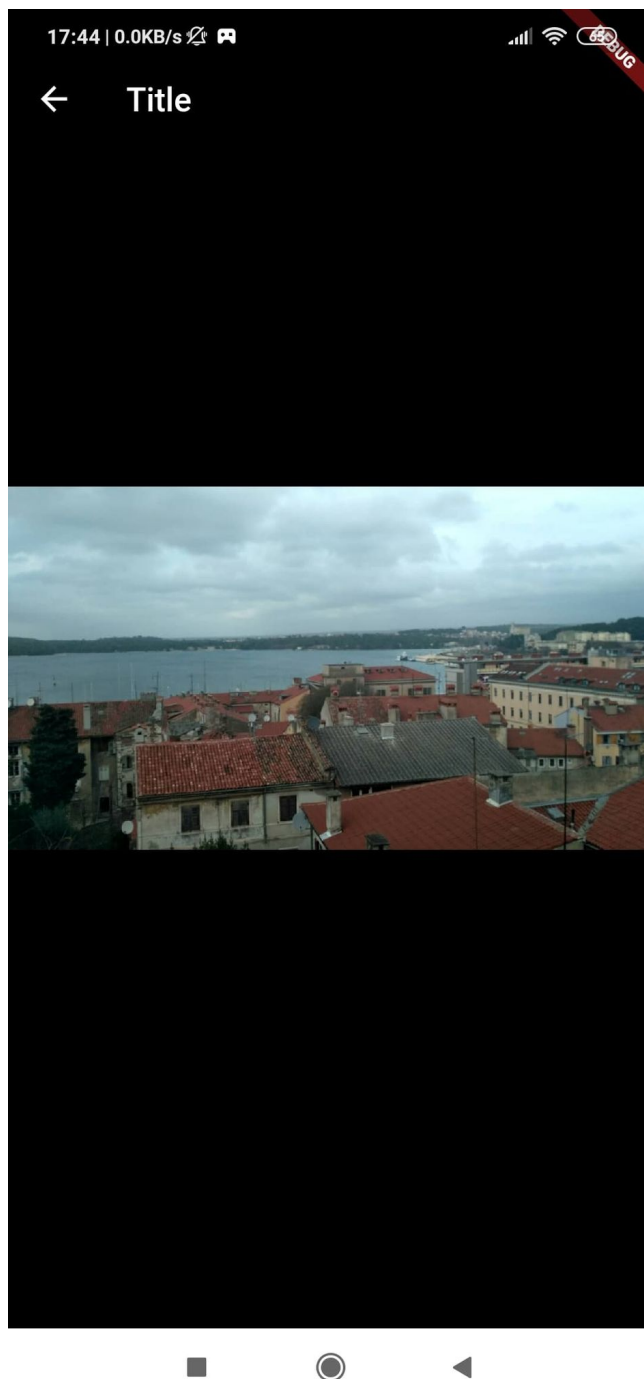
Korisnik može odabrati bilo koju fotografiju iz galerije i dobiti više informacija o prikazanoj lokaciji. Na donjoj slici vidimo koje se sve informacije nude ispod uvećane fotografije: ime korisnika koji je dodao fotografiju, grad i država, geografske koordinate te dva gumba. Prvi

gumb otvara Google Maps aplikaciju i prikazuje rutu od lokacije korisnika do lokacije na fotografiji. Drugu gumb pokazuje fotografiranu lokaciju na interaktivnoj mapi. Na vrhu ekrana nalazi se gumb za povratak na prijašnji ekran.



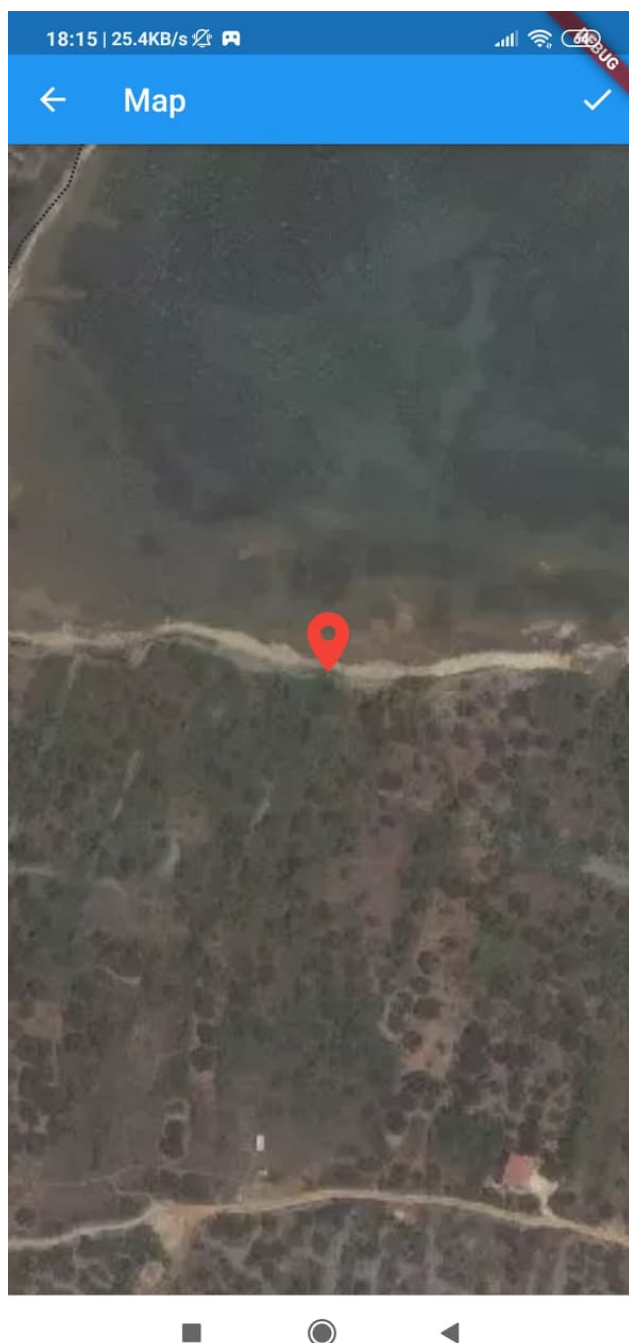
Slika 6: Ekran sa više informacija o lokaciji.

Odabirom fotografije na prethodnom ekranu otvara se uvećana verzija fotografije. U gornjem lijevom kutu nalazi se gumb za povratak.



Slika 7: Ekran sa uvećanom fotografijom.

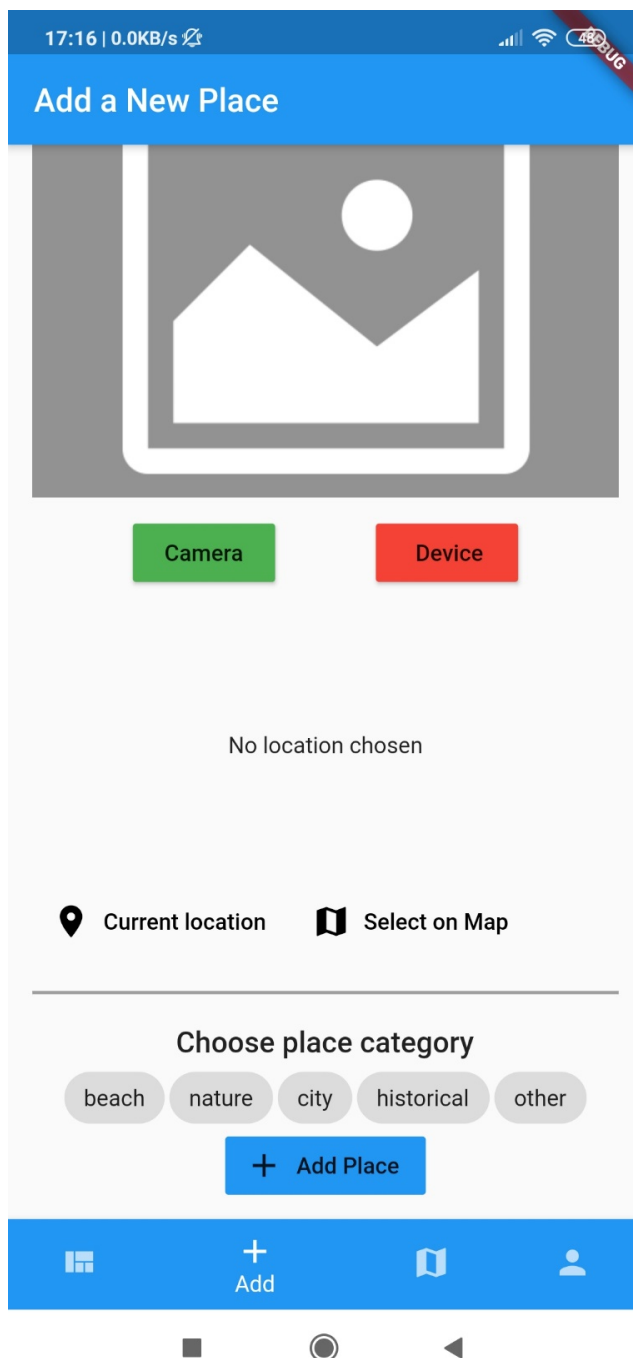
Slijedi i primjer ekrana koji se pokaže nakon što se odabere gumb „Show on Map”.



Slika 8: Ekran sa odabranom lokacijom na mapi.

Na sljedećoj slici možemo vidjeti ekran za dodavanje lokacije. Korisnik može direktno slikati lokaciju ili dodati neku od postojećih slika na uređaju. Za olakšavanje dodavanja same lokacije, korisnik može klikom na „Current location” odabrati svoju trenutnu GPS lokaciju. Moguće je odabrati i proizvoljnu lokaciju klikom na „Select on Map”. Ta opcija vodi

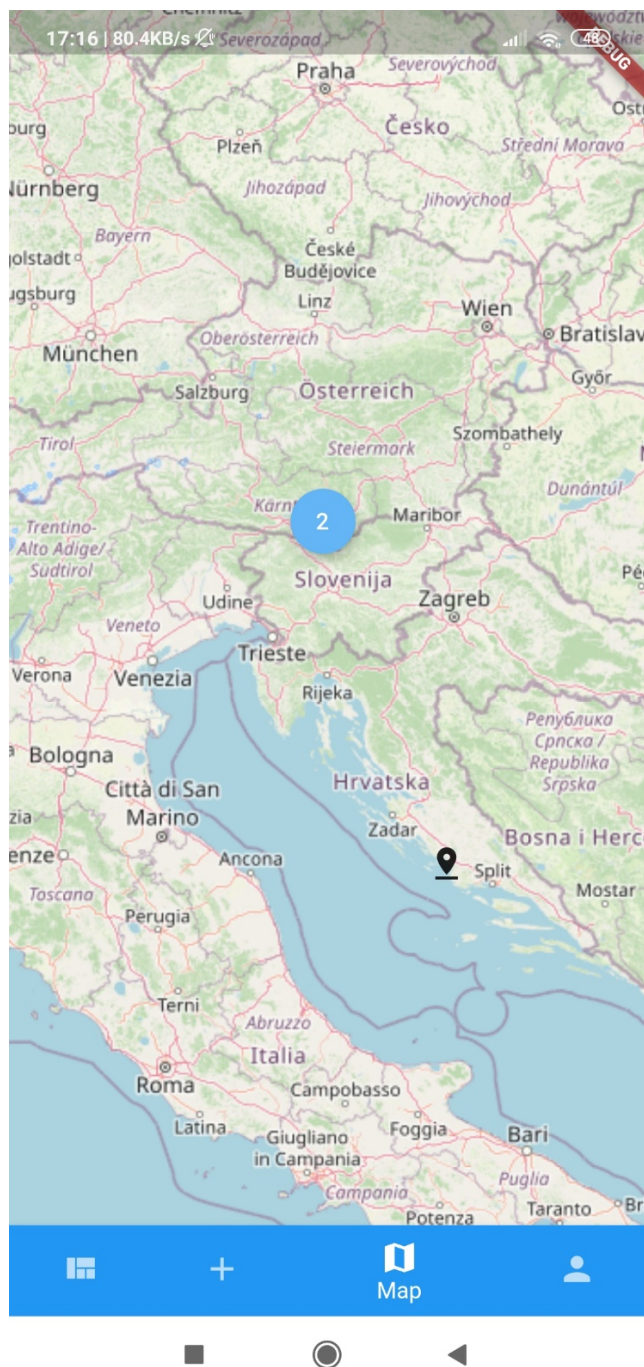
korisnika na interaktivnu mapu. Za svaku dodanu fotografiju potrebno je odabrati njenu kategoriju. Dostupne kategorije su: plaža, priroda, grad, povijesna lokacija i ostalo.



Slika 9: Ekran za dodavanje lokacije.

Treći gumb u glavnoj navigaciji vodi na ekran „Map”. On sadrži interaktivnu mapu sa svim dodanim lokacijama. Brojevi na mapi (kao npr. broj „2” na donjoj slici) označavaju skup lokacija koje su preblizu da bi se prikazale kao odvojene ikone. Lokacije će se pojaviti

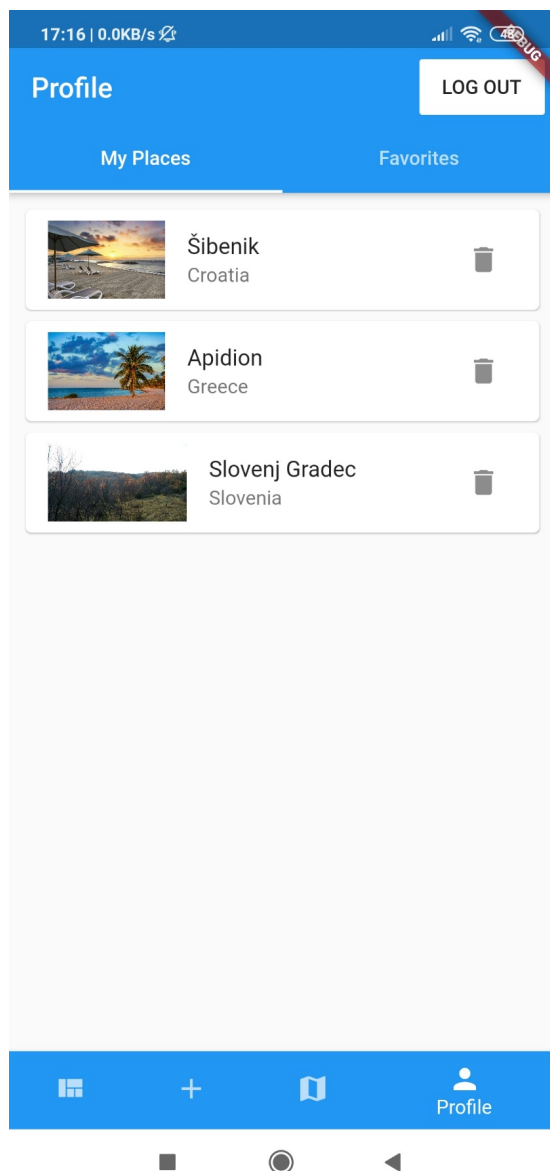
odvojene nakon što korisnik uveća mapu. Na donjoj mapi možemo vidjeti jednu dodanu lokaciju zapadno od Splita.



Slika 10: Ekran sa lokacijama na interaktivnoj mapi.

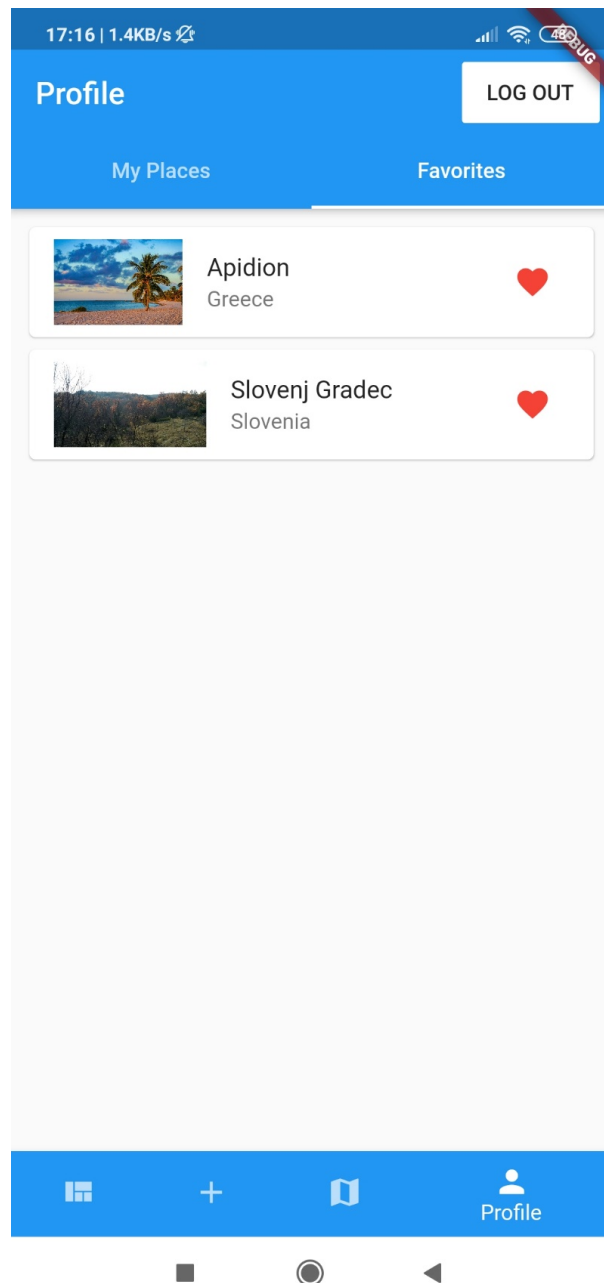
Klikom na četvrti gumb glavne navigacije korisnik može vidjeti svoj profil. U gornjem desnom kutu ekrana nalazi se gumb za odjavu koji vodi na početni ekran za prijavu tj.

registraciju. Ekran sa korisničkim profilom sadrži dvije kartice. Prva sadrži listu lokacija koje je korisnik dodao. Pojedine lokacije se mogu izbrisati odabirom ikone koša za smeće.



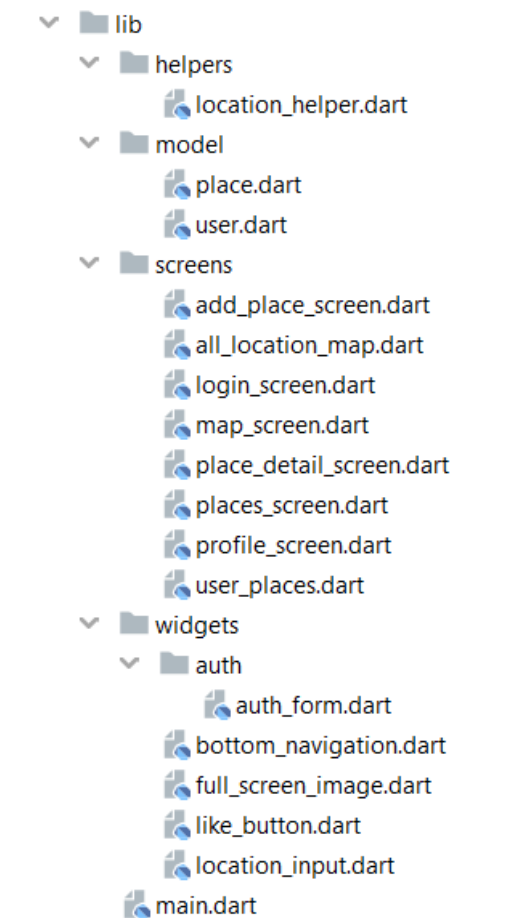
Slika 11: Prva kartica na ekranu sa korisničkim profilom.

Druga kartica na „Profile” ekranu sadrži mjesta koja je korisnik lajkao. Klikom na ikonu srca lokacija se može maknuti sa liste. Odabirom lokacije otvara se već opisani ekran sa više detalja.



Slika 12: Druga kartica na ekranu sa korisničkim profilom.

Arhitektura aplikacije



Slika 13: Sadržaj mape lib sa glavnim kodom aplikacije.

Unutar mape sa aplikacijom nalazi se mapa lib koja sadrži glavni kod.

Datoteka main.dart sadrži logiku za inicijalizaciju. U njoj je definirana nova MaterialApp aplikacija koja koristi Material Design dizajn. U pozivu konstruktora za MaterialApp definiran je naslov aplikacije, glavna paleta boja i početni ekran. U slučaju da korisnik nije prijavljen, početni ekran će biti ekran za prijavu. U suprotnom prikazuje se jedan od sljedećih ekrana: PlacesScreen (galerija), AddPlaceScreen (dodavanje fotografije tj. lokacije), AllLocationMap (mapa sa lokacijama) i ProfileScreen (korisnički profil). Koji ekran će se prikazati ovisi o odabranoj stavki navigacije. PlacesScreen je zadani ekran i on se pokazuje nakon pokretanja aplikacije.

Mapa helpers sadrži datoteku `location_helper.dart`. U njoj se nalazi klasa `LocationHelper` koja se koristi za dobivanje satelitske slike lokacija i za dobivanje naziva mjesta na temelju danih geografskih koordinata.

Unutar mape model nalaze se datoteke sa klasama specifičnim za aplikaciju. Tako `place.dart` sadrži npr. klasu `PlaceLocation` čiji se objekti, između ostalog, koriste za predstavljanje lokacije na ekranu za dodavanje lokacije (`AddPlaceScreen`). Datoteka `user.dart` u istoj mapi sadrži klasu `User` čiji objekti predstavljaju korisnika aplikacije. Klasa `User` ima tri atributa: korisničko ime, email adresa i lozinka.

Mapa `screens`, kao što samo ime kaže, sadrži datoteke sa klasama koje predstavljaju ekrane aplikacije. Ekрани su već opisani u prethodnom poglavlju tako da ih ovdje samo nabrajamo: `add_place_screen.dart` (ekran za dodavanje lokacije), `all_location_map.dart` (mapa sa svim lokacijama), `login_screen.dart` (ekran za prijavu tj. registraciju), `map_screen.dart` (ekran za postavljanje markera lokacije na mapi), `place_detail_screen.dart` (ekran sa detaljima lokacije), `places_screen.dart` (galerija), `profile_screen.dart` (korisnički profil) i `user_places.dart` (sve lokacije koje je dodao neki korisnik).

Pod mapom `widgets` nalaze se "ručno" definirani widgeti. Datoteka `full_screen_image.dart` sadrži klasu `FullScreenImage` koja se koristi za prikaz uvećane fotografije lokacije. `like_button.dart` sadrži klasu `LikeButton` koja predstavlja gumb za lajkanje. `location_input.dart` sadrži `LocationInput` klasu. Ona sadrži obrazac za unos detalja o lokaciji na ekranu za dodavanje nove lokacije. Unutar `widgets` mape nalazi se i `auth` mapa koja sadrži datoteku `auth_form.dart`. U njoj je definirana klasa `AuthForm` koja predstavlja obrazac za prijavu tj. registraciju. Obraz se koristi na ekranu `LoginScreen`.

Zaključak

U radu je dan opis Flutter softvera za izradu cross-platform mobilnih aplikacija. Navedene su njegove prednosti i mane. Glavne prednosti su performanse i to što ga razvija Google. Dana je i usporedba Fluttera sa tri popularne alternative za razvoj cross-platform mobilnih aplikacija. Flutter je realna opcija za razvoj mobilnih aplikacija a glavna konkurencija danas mu je React Native.

Druga polovica rada sadrži opis Flutter aplikacije za dijeljenje fotografija zanimljivih turističkih lokacija. Nakon opisa baze podataka, detaljno su navedeni svi ekrani aplikacije. Na kraju rada dan je opis arhitekture koda aplikacije.

Literatura

- [1] „Cross-platform software“, *Wikipedia*. kol. 06, 2020, Pristupljeno: kol. 16, 2020. [Na internetu]. Dostupno na: https://en.wikipedia.org/w/index.php?title=Cross-platform_software&oldid=971494855.
- [2] „FAQ“. <https://flutter.dev/docs/resources/faq> (pristupljeno kol. 15, 2020).
- [3] C. Software, „What is Flutter? Here is everything you should know“, *Medium*, kol. 26, 2019. <https://medium.com/@concisesoftware/what-is-flutter-here-is-everything-you-should-know-faed3836253f> (pristupljeno kol. 16, 2020).
- [4] „Showcase“. <https://flutter.dev/showcase> (pristupljeno kol. 16, 2020).
- [5] „Flutter (software)“, *Wikipedia*. kol. 14, 2020, Pristupljeno: kol. 16, 2020. [Na internetu]. Dostupno na: [https://en.wikipedia.org/w/index.php?title=Flutter_\(software\)&oldid=972866337](https://en.wikipedia.org/w/index.php?title=Flutter_(software)&oldid=972866337).
- [6] „Dart (programming language)“, *Wikipedia*. kol. 14, 2020, Pristupljeno: kol. 16, 2020. [Na internetu]. Dostupno na: [https://en.wikipedia.org/w/index.php?title=Dart_\(programming_language\)&oldid=972865334](https://en.wikipedia.org/w/index.php?title=Dart_(programming_language)&oldid=972865334).
- [7] K. Chinnathambi, „Understanding WebViews“, *kirupa.com*. <https://www.kirupa.com/apps/webview.htm> (pristupljeno kol. 27, 2020).
- [8] „Introduction to cross-platform app development - itCraft blog“, *itCraft | Mobile App Development Company*, svi. 21, 2020. <https://itcraftapps.com/blog/introduction-to-cross-platform-app-development/> (pristupljeno kol. 16, 2020).
- [9] „Stack Overflow Trends“. <https://insights.stackoverflow.com/trends?tags=flutter%2Cxamarin%2Cionic-framework%2Creact-native> (pristupljeno kol. 27, 2020).
- [10] „Google Trends“, *Google Trends*. https://trends.google.com/trends/explore?date=today%205-y&q=%2Fg%2F11f03_rzbg,react%20native,xamarin,%2Fg%2F1q6l_n0n0 (pristupljeno kol. 27, 2020).
- [11] „Stack Overflow Developer Survey 2020“, *Stack Overflow*. https://insights.stackoverflow.com/survey/2020/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2020 (pristupljeno kol. 27, 2020).
- [12] „Flutter vs Xamarin: The 2019 Developer’s Guide [Infographics Included]“, *Apiko | Learn*, lis. 25, 2019. <https://apiko.com/blog/flutter-vs-xamarin-the-complete-2019-developers-guide-infographics-included/> (pristupljeno kol. 18, 2020).
- [13] „React Native · A framework for building native apps using React“. <https://reactnative.dev/> (pristupljeno kol. 20, 2020).
- [14] A. Ravichandran, „React Native or Flutter — What Should I Pick To Build My Mobile App?“, *Medium*, lip. 21, 2019. <https://medium.com/@adhithiravi/react-native-vs-flutter-what-are-the-differences-b6dc892f0d34> (pristupljeno kol. 20, 2020).
- [15] „Flutter vs React Native – what to choose in 2020?“, *Droids On Roids*, pros. 11, 2019. <https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2020> (pristupljeno kol. 20, 2020).
- [16] R. Borah, „Ionic vs. Flutter: Which one Works for You?“, <https://www.clariontech.com/blog/ionic-vs.-flutter-which-one-works-for-you> (pristupljeno kol. 20, 2020).
- [17] Ionic, „Ionic Framework - Ionic Documentation“, *Ionic Docs*. <https://ionicframework.com/docs/undefined> (pristupljeno kol. 20, 2020).
- [18] M. Smallcombe, „SQL vs NoSQL: 5 Critical Differences“, *Xplenty*. <https://www.xplenty.com/blog/the-sql-vs-nosql-difference/> (pristupljeno kol. 23, 2020).

- [19] „DB-Engines Ranking“, *DB-Engines*. <https://db-engines.com/en/ranking> (pristupljeno kol. 23, 2020).
- [20] „Firestore: NoSQL document database“, *Google Cloud*. <https://cloud.google.com/firestore> (pristupljeno kol. 23, 2020).

Popis slika

Slika 1: Postotak mjesečnih pitanja na StackOverflowu po tehnologiji (vertikalna os) kroz vrijeme (horizontalna os).....	9
Slika 2: Relativna popularnost pojmova na Google tražilici (vertikalna os) kroz vrijeme (horizontalna os). Lijevo od glavnog grafa nalazi se prosječna popularnost.....	10
Slika 3: Rezultati StackOverflow ankete omiljenih tehnologija. Graf prikazuje postotak anketiranih koji su koristili određenu tehnologiju (vertikalna os) i htjeli bi je koristiti ponovno.....	11
Slika 4: Početni ekran aplikacije.....	17
Slika 5: Ekran sa galerijom.....	18
Slika 6: Ekran sa više informacija o lokaciji.....	19
Slika 7: Ekran sa uvećanom fotografijom.....	20
Slika 8: Ekran sa odabranom lokacijom na mapi.....	21
Slika 9: Ekran za dodavanje lokacije.....	22
Slika 10: Ekran sa lokacijama na interaktivnoj mapi.....	23
Slika 11: Prva kartica na ekranu sa korisničkim profilom.....	24
Slika 12: Druga kartica na ekranu sa korisničkim profilom.....	25
Slika 13: Sadržaj mape lib sa glavnim kodom aplikacije.....	26

Sažetak

Rad donosi opis Flutter softvera za razvoj cross-platform mobilnih aplikacija. Osim prednosti i mana, dana je i usporedba Fluttera sa ostalim sličnim tehnologijama. Rad sadrži i opis aplikacije za dijeljenje fotografija lokacija izrađene Flutterom.