

**MOBILNA APLIKACIJA ZA PRONALAZENJE MAJSTORA
MOBILE APPLICATION FOR FINDING THE HANDYMAN**Nataša Josić, *Fakultet tehničkih nauka, Novi Sad***Oblast – INŽENJERSTVO INFORMACIONIH
SISTEMA**

Kratak sadržaj – *Kroz ovaj rad biće opisan način na koji se može kreirati Android mobilna aplikacija koja služi za pretragu majstora kao i njihovo prijavljivanje na oglase. Prikazani su postupci razvijanja aplikacije. Detaljno će biti opisane korišćene tehnologije, kao i sama arhitektura aplikacije.*

Ključne reči: *Mobilne aplikacije, Flutter, Firebase, Android, Cubit*

Abstract – *This paper will describe the way in which an Android mobile application can be created. This application will be used to search for handymen and also their registration for projects. Application development procedures are presented. The technologies used will be described in detail, as well as the application architecture itself.*

Keywords: *Mobile applications, Flutter, Firebase, Android, Cubit*

1. UVOD

Potreba za mobilnim aplikacijama raste iz dana u dan, samim tim i potreba za softverskim rješenjima. Većina ljudi i dalje nalazi majstore pomoću preporuka, oglasa ili slučajnim izborom, iako živimo u 21. vijeku i pronalaznje majstora bi trebalo biti jednostavnije.

Pronalaznje majstora za bilo koju vrstu radova često može da bude mukotrpan proces. Nerijetko se dešava da potrošimo puno vremena na pronalaznje majstora, da bi se na kraju ispostavilo da smo angažovali nekog ko i nije kvalifikovan za radove ili popravke koje hoćemo da sprovedemo, tako da sve rezultuje velikim gubitkom vremena, živaca i novca.

Cilj ovog rada jeste prikaz razvijanja Android mobilne aplikacije za pretraživanje majstora.

Sa druge strane, majstori će imati pregled svih oglasa iz njihove kategorije i lokacije na kojoj se nalaze. Moći će da kontaktiraju korisnike koji su postavili taj oglas i tako lakše nađu posao.

U ovom radu ćemo riješiti taj problem mobilnom aplikacijom za pretragu majstora i opisati postupak izrade iste. Aplikacija će biti napisana u programskom jeziku Dart, korišćenjem Flutter razvojnog okvira, editor koda Visual Studio Code i pokretaćemo je u okviru Android operativnog sistema. Takođe će biti korišćen Firebase, Google-ova platforma za razvoj mobilnih i veb aplikacija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Srđan Sladojević.

2. KORIŠĆENE TEHNOLOGIJE

Tehnologije koje su korišćene za kreiranje aplikacije su:

- Flutter
- Dart
- Firebase

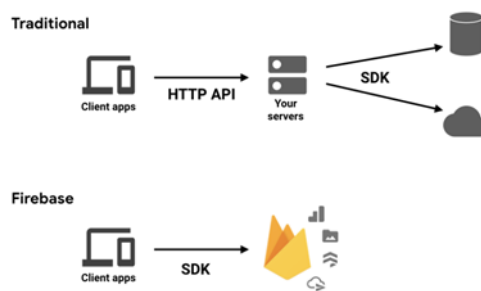
Za izradu ove aplikacije korišćena je arhitektura bez servera (*serverless architecture*). Ovaj termin se koristi da bi opisao aplikacije čiji se server nalazi na nekom od klad provajdera.

Aplikacija koristi i ostale pogodnosti koje provajder nudi, poput baze podataka, autentifikacije i druge.

Arhitektura aplikacije se može podijeliti na dva dijela:

- klijentske aplikacije koja komunicira sa Firebase-om pomoću SDK-a i
- Firebase-a.

Firebase proizvodi imaju pozadinske komponente koje u potpunosti održava i kojima upravlja Google. Klijentski SDK-ovi koje obezbjeđuje Firebase direktno komuniciraju sa ovim pozadinskim uslugama, bez potrebe za uspostavljanjem bilo kakvog međuservera između klijentske aplikacije i Firebase-a [1].



Slika 1. Prikaz arhitekture sa i bez servera

U nastavku je dat kratak opis korišćenih tehnologija.

2.1. Flutter

Flutter je Google-ov UI alat za pravljenje prelepih, izvorno kompajliranih aplikacija za mobilne uređaje, veb i desktop aplikacije iz jedne baze koda.

Flutter radi sa postojećim kodom, koriste ga programeri i organizacije širom sveta, besplatan je i otvorenog koda (eng: open source). Flutter je pristupačan programerima koji su upoznati sa objektno orijentisanim konceptima (klase, metode, promenljive, itd.) i konceptima imperativnog programiranja (petlje, uslovi, itd).

Flutter je dizajniran da podrži mobilne aplikacije koje rade na Android i iOS uređajima, kao i interaktivne aplikacije koje pokrećemo na veb stranicama ili na desktop aplikacijama.

Flutter se isporučuje sa skupom visokokvalitetnih *Material Design* i *Cupertino* (iOS stil) vidžeta, izgleda i tema. Naravno, ovi vidžeti su samo početna tačka. *Flutter* je dizajniran da olakša kreiranje sopstvenih vidžeta ili prilagođavanje postojećih vidžeta [2].

Centralna ideja iza Flutter-a je upotreba vidžeta. Kombinovanjem različitih vidžeta programeri mogu da naprave cio korisnički interfejs. Svaki od ovih vidžeta definiše strukturni element (kao što je dugme ili meni), stilski element (font ili šema boja), aspekt izgleda (kao što je padding) i mnoge druge. Naravno, programeri mogu kreirati i svoje vidžete.

Sami vidžeti nemaju promjenljivo stanje (sva njihova polja moraju biti konačna). *StatefulWidget* se koristi ukoliko želimo da mijenjamo stanje vidžeta [3].

Bilo da se implementiraju veb aplikacije ili mobilne aplikacije, upravljanje stanjem je ključ za upravljanje prikazima aplikacija. Upravljanje stanjem je strateški pristup za upravljanje svim interakcijama koje korisnik obavlja na aplikaciji i zatim odražava te promjene korisničkog interfejsa, ažuriranja baza podataka, serverskih zahteva itd. Za razliku od drugih okvira, Flutter pokušava da ponovo izgradi korisnički interfejs od nule, kako bi odražavao najnovije promjene u stanju.

2.2. Dart

Dart je otvorenog koda (eng. *open-source*), opšte namjene, objektno orijentisan programski jezik sa sintaksom u stilu C koji je razvio Google 2011. Svrha Dart programiranja je kreiranje frontend korisničkih interfejsa za veb i mobilne aplikacije. Dart je trenutno u aktivnom razvoju. Kompajliran je u izvorni mašinski kod za pravljenje mobilnih aplikacija, inspirisan drugim programskim jezicima kao što su Java, JavaScript, C#. Dart je kompajlirani jezik, tako da ne možete direktno izvršiti svoj kod; umjesto toga, kompajler ga analizira i prenosi u mašinski kod.

Podržava većinu uobičajenih koncepata programskih jezika kao što su klase, interfejsi, funkcije, za razliku od drugih programskih jezika. Dart jezik ne podržava nizove direktno [4].

2.3. Firebase

U početku je bila samo realtime baza, dok sada ima 18 servisa u ponudi. Cijela platforma je servis rješenja za mobilne i veb aplikacije koje nude usluge poput građenja, testiranja i upravljanja aplikacijama.

Postoji niz namjenskih API-ja za svaki od servisa koji Firebase nudi, tako da je jednostavno povezivanje aplikacije sa Firebase-om. Firebase pokriva dosta uobičajenih servisa koje bi inače developeri implementirali sami. Ti servisi uključuju autentifikaciju, bazu podataka, konfiguraciju, slanje poruka i mnoge druge.

Ovi servisi se nalaze na oblaku (eng. *cloud*) koje održava Google. Klijentski paketi za razvoj softvera koje obezbjeđuje Firebase direktno komuniciraju sa ovim pozadinskim uslugama, bez potrebe za uspostavljanjem bilo kakvog međuservisa između aplikacije i usluga Firebase-a. Ovo se

razlikuje od tradicionalnog pisanja aplikacije koje uključuje frontend i backend softvera. Uključivanjem Firebase-a, sa klijenta se poziva API tako da nije neophodno pisanje backend softvera.

Pristup Firebase servisima se odvija pomoću Firebase konzole. Neki od servisa koje Firebase nudi su:

Firebase Authentication vodi računa o tome da se korisnici prijave i identifikuju. Ovaj proizvod je neophodan za pravilno konfigurisanje aplikacije, posebno ako treba da se ograniči pristup podacima po korisniku (što će skoro svaka aplikacija željeti da uradi). Ono što je posebno u vezi sa Firebase autentifikacijom je to što olakšava bezbjedno prijavljivanje.

Takođe je moguće uvezivanje korisnikovog naloga sa drugim provajderima kao što su Google, Facebook i GitHub. Za aplikaciju „Velemajstor” je korišćen ovaj servis i omogućeno je logovanje uz pomoć email-a/lozinke, Google i Facebook naloga.

Firebase Realtime Database i **Cloud Firestore** pružaju usluge baze podataka. Obje baze su NoSQL i ažuriraju podatke u „realnom vremenu” kako se mijenjaju u bazi podataka. Koristi se klijentski SDK da bi se podesio „prisluškivač” (eng. listener) na lokaciji podataka koje koristi aplikacija. Za aplikaciju „Velemajstor” je korišćen Cloud Firestore.

Cloud Storage pruža usluge skladištenja fajlova. Klijentska aplikacija koristi Firebase SDK za skladištenje i preuzimanje datih fajlova.

Cloud Functions je još jedan servis Firebase-a pomoću kog se može pisati i slati kod na Firebase oblak. On reaguje na događaje koji dolaze iz ostalih Firebase servisa [5].

3. OPIS FUNKCIONALNOSTI SISTEMA

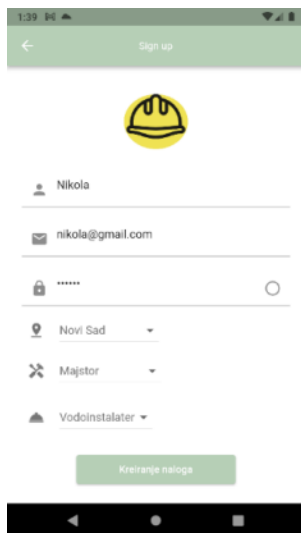
Korisnik ove aplikacije će imati mogućnost da nađe odgovarajućeg majstora i obrnuto, ukoliko je u pitanju majstor, biće u mogućnosti da se prijavi na oglase.

Prije svega, korisnik mora imati napravljen nalog na aplikaciji kako bi mogao da je koristi (slika 1).

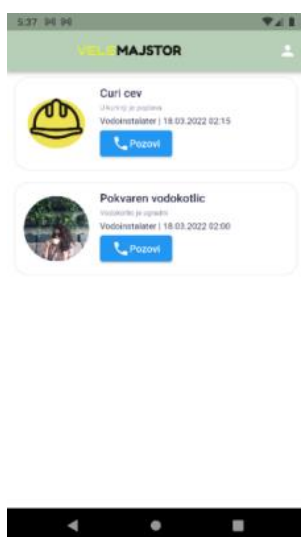
Za autentifikaciju je korišćen *Firebase Authentication* servis. S obzirom da od stanja autentifikacije zavisi rad cijele aplikacije, korišćen je obrazac *ubit* za upravljanje stanjem aplikacije.

ubit je minimalna verzija odn. podskup *Bloc* obrasca, koji pojednostavljuje način na koji upravljamo stanjem aplikacije. Da bi to uradio, on zamjenjuje upotrebu događaja (koji se koriste u *Bloc*-u) sa funkcijama koje ponovo grade korisnički interfejs, emitujući različita stanja u toku (eng. *stream*). Ukoliko uspješno napravimo nalog, emitovaćemo stanje *AuthSignUpSuccess*, a ukoliko dođe do greške emitovaćemo *AuthSignUpError*.

Na slici 2 prikazan je početni ekran koji se prikazuje nakon što se majstor uloguje. Na njemu su prikazani svi oglasi sa lokacije na kojoj se nalazi majstor. Takođe, ti oglasi su iz kategorije tog majstora. Na svakom oglasu se nalazi dugme „Pozovi” klikom na koji će se pozvati kreator oglasa.



Slika 2. Primjer kreiranja naloga



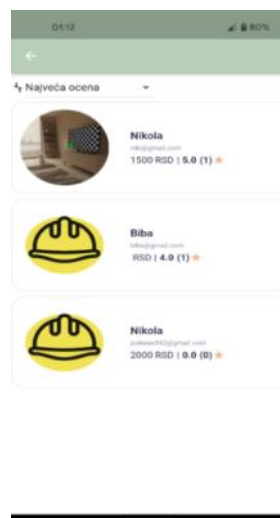
Slika 3. Oglasi

Na slici 3 prikazana je početna strana ukoliko se uloguje korisnik koji traži majstora. Prikazane su kategorije majstora kao i prosječna početna cijena za datu kategoriju. Omogućeno je i pretraživanje kategorija. Klikom na kategoriju dobijamo spisak dostupnih majstora iz te kategorije.



Slika 4. Prikaz kategorija majstora

Na slici 4 su prikazani majstori sortirani po najvećoj ocjeni. Takođe je moguće sortirati majstore po najmanjoj i najvećoj početnoj cijeni, kao i po popularnosti tj. po broju recenzija.



Slika 5. Majstori raspoređeni po najvećoj oceni

Kada se klikne na majstora, iskoči nam ekran sa svim detaljima datog majstora, kao što je prikazano na slici 5.



Slika 6. Prikaz izabranog majstora

4. ZAKLJUČAK

Ovim radom dat je prikaz upotrebe Android mobilne aplikacije za pronalaženje majstora. Početak rada se sastoji iz kratkog uvoda, nakon čega sledi prikaz tehnologija pomoću kojih je kreirana aplikacija. Glavni dio rada bio je vezan za implementaciju aplikacije.

Cilj ovog rada jeste prikaz razvijanja Android mobilne aplikacije za pretraživanje majstora. Iako dobro osmišljena, aplikacija ima prostora za nadogradnju. Dodavanjem chat-a i omogućavanjem komunikacije u okviru aplikacije bi se znatno poboljšalo korisničko iskustvo. Aplikacija bi mogla da se nadgradi tako što bi grupisala više kategorija u jednu i tako olakšala korisnicima organizovanje velikih radova. Na primjer, pod kategorijom selidba bi se mogle dodati podkategorije usluga prevoza, postavljanje TV-a na zid, sastavljanje namještaja, krećenje itd.

Još jedna od nadogradnji jeste i prilagođavanje implementacije i za veb i iOS aplikacije, tako da bi više korisnika moglo da je koristi.

5. LITERATURA

- [1] Luksa, M. (2018). Kubernetes in Action. Manning Publications.
- [2] Mario Casciaro, L. M. (2020). Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques, 3rd Edition. Packt Publishing.
- [3] Quevedo, W. (2018). Practical NATS: From Beginner to Pro. Apress.
- [4] Ronnie Mitra, I. N. (2020). Microservices Up & Running. O'Reilly.
- [5] Schenker, G. N. (2020). Learn Docker - Fundamentals of Docker 19.x. Packt Publishing.
- [6] What is MongoDB? Introduction, Architecture, Features & Example. (n.d.). (Guru99) Preuzeto sa: <https://www.guru99.com/what-is-mongodb.html>

Kratka biografija:



Nataša Josić, rođena je u Tuzli 1992. godine. 2011. godine upisuje Fakultet tehničkih nauka u Novom Sadu na smjeru Geodezija i geomatika koji završava 2016. godine. 2018. godine upisuje master studije na smjeru Inženjerstvo informacionih sistema koji završava 2022. godine.