Hibridna mobilna aplikacija za pohranjivanje lovačkih trofeja

Valentić, Antonio

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:609851

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

HIBRIDNA MOBILNA APLIKACIJA ZA POHRANJIVANJE LOVAČKIH TROFEJA

Završni rad

Antonio Valentić

Osijek, 2020.

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

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

Osijek, 31.08.2020.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Antonio Valentić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4002, 26.09.2019.
OIB studenta:	50567117943
Mentor:	Doc.dr.sc. Josip Balen
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Hibridna mobilna aplikacija za pohranjivanje lovačkih trofeja
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena 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 mentora:	31.08.2020.
Datum potvrde ocjene Odbora:	09.09.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 11.09.2020.

Ime i prezime studenta:	Antonio Valentić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4002, 26.09.2019.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Hibridna mobilna aplikacija za pohranjivanje lovačkih trofeja**

izrađen pod vodstvom mentora Doc.dr.sc. Josip Balen

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 2

2. PREGLED KORIŠTENIH TEHNOLOGIJA 3

2.1. Android operacijski sustav 3

2.2. iOS operacijski sustav 4

2.3. Hibridne mobilne aplikacije 4

2.4. Flutter platforma 5

2.5. Programski jezik Dart 7

2.6. Program za pisanje koda Visual Studio Code 8

2.7. Servis za aplikacije Firebase 9

3. RAZVOJ APLIKACIJE 11

3.1. Prijava korisnika 11

3.2. Registracija korisnika 14

3.3. Zasloni s trofejima 15

3.4. Zaslon s detaljima trofeja 18

3.5. Logika dodavanja trofeja 19

3.6. Korisnički profil 20

3.7. Dio aplikacije s informacijama 22

4. ZAKLJUČAK 23

SAŽETAK 24

ABSTRACT 25

ŽIVOTOPIS 26

LITERATURA 27

1. UVOD

Tema ovog završnog rada je izrada hibridne mobilne aplikacije koja će omogućiti lovcima pohranu trofeja sa pripadajućim informacijama o tom trofeju. Lovci trenutno svoje odstrijeljene trofeje fotografiraju i dijele na društvenim mrežama ili čuvaju u galerijama na svojim telefonima. Takav način dijeljenja trofeja, odnosno spremanja trofeja za lovce često nije praktičan zbog toga što društvene mreže nisu namijenjene tomu, a u galerijama se slike vrlo često znaju zagubiti ili se mogu zauvijek izgubiti ukoliko se fotografije nemaju sigurnosnu kopiju na nekom oblaku. Također, lovci na trenutačnim društvenim mrežama nemaju predviđena mjesta unos informacija o trofeju kao što su oružje s kojim je trofej odstrijeljen, kalibar oružja, optika, vrsta zrna te druge slične informacije koje su lovačkom svijetu podosta bitne.

Jednostavnim grafičkim sučeljem i pregršt funkcionalnosti prilagođenih baš prema lovcima, ova aplikacija za cilj ima riješiti takve probleme, te ujediniti lovce na jednu platformu za dijeljenje trofeja. Za početak aplikacija će omogućiti lovcima unos svakog odstrijeljenog trofeja uz pripadajuće fotografije, vrstu oružja s kojim je trofej odstrijeljen, datum odstrjela, lokaciju odstrjela, optiku na oružju, informacijama o zrnu te bilješkama. Nakon što lovac unese trofej u aplikaciju, sve informacije će se pohraniti u bazu podataka i povezati sa računom u aplikaciji tako da se trofeji nikada neće moći izgubiti. Uz to aplikacija će omogućivati i vrlo jednostavno dijeljenje trofeja kroz aplikaciju, a unesene trofeje će moći pogledati i korisnici aplikacije i oni koji ju nemaju kroz internetski preglednik. Unutar aplikacije će postojati i dio gdje će korisnici aplikacije koje se odluče platiti pretplatu moći pregledavati sve javne trofeje od drugih korisnika aplikacije. Aplikacija cilja na svjetsko tržište, pa zbog toga mora biti podržana na većini mobilnih uređaja koji se danas koriste u svijetu. Zbog toga je za izradu aplikacije korištena platforma za izradu aplikacija Flutter koju je razvila tvrtka Google i koja omogućuje hibridni razvoj mobilnih aplikacija.

Prvi dio završnog rada sadrži kratki opis dvaju platformi za koje se aplikacija razvija, te opis nekoliko tehnologija za izradu hibridnih mobilnih aplikacija. Nakon toga slijedi detaljni opis platforme za izradu hibridnih aplikacija Flutter, opis programskog jezika Dart, uobičajena razvojna okruženja i testiranja aplikacija. Na kraju će biti opis nastanka aplikacije, snimke zaslona te kod aplikacije s detaljnim opisom.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je opisati nekoliko trenutno najvažnijih platformi za izradu hibridnih mobilnih aplikacija, detaljno opisati platformu Flutter, te prednosti i nedostatke iste. U praktičnom dijelu rada potrebno je razviti Android i iOS mobilnu aplikaciju koju će koristiti lovci za spremanje i upravljanje podacima o svojim lovačkim trofejima. Mobilna aplikacija treba biti sinkronizirana s bazom podataka na internetu, te omogućiti registraciju korisnika i dijeljenje sadržaja.

2. PREGLED KORIŠTENIH TEHNOLOGIJA

Mobilni operacijski sustavi su operacijski sustavi koji danas pokreću pametne telefone, tablete, poneka prijenosna računala i slične uređaje. Ovi operacijski sustavi kombiniraju značajke operacijskih sustava za osobna računala i značajke koje su korisne za prijenosne uređaje, a to su danas pristup internetu i ostalim telekomunikacijskim uslugama. Dva mobilna operacijska sustava koji se danas najviše koriste su Android i iOS. Trenutno na svijetu ima 3.5 milijardi korisnika pametnih telefona od kojih 72.26% pokreće Android operacijski sustav, a 27.03% iOS [1][2]. Tržište razvoja mobilnih aplikacija je trenutno jako veliko i tvrtke koje se bave razvojem mobilnih aplikacija su trenutno jedne od najbogatijih tvrtki na svijetu.

2.1. Android operacijski sustav

Android je operacijski sustav otvorenog koda s jezgrom Linuxa. „Android Inc. osnovali su Andy Rubin i Rick Miner 2003. godine kako bi razvijali programe za pametne uređaje“ [3]. Dvije godine poslije, tvrtku Android Inc. je kupio tehnološki div Google. Nakon toga, Google je počeo otkupljivati i različite tvrtke koje posluju u domeni mobilne industrije. Kada su prikupili dostatan broj, osnovali su OHA (engl. Open Handset Alliance) kojemu je cilj bio stvoriti javni standard za mobilne uređaje. Dana 5. studenog 2007. godine, OHA i Google su najavili prvu potpuno otvorenu i sveobuhvatnu platformu za mobilne uređaje naziva Android [4]. Deset mjeseci nakon najave, prva verzija operacijskog sustava je puštena u javnost [5]. HTC Dream (poznat i kao T-Mobile G1) je bio prvi mobilni telefon koji je pokretao ovaj operacijski sustav [6].

Arhitektura Android operacijskog sustava podijeljena je u nekoliko dijelova i slojeva, svaki od tih slojeva ima različite karakteristike i zadaće. Granice između slojeva nisu jasno definirane, pojedini slojevi se mogu i preklapati s drugima. „Android operacijski sustav je baziran na Linux Kernel 2.6 jezgri napisanom u C ili C++ jeziku“. Iako je operacijski sustav razvijen u C/C++ razvojnom okruženju, aplikacije se uz ova dva jezika mogu razvijati i pomoću Java i Kotlin programskih jezika. Sama Linux jezgra ovom operacijskom sustavu omogućuje pokretanje na vrlo velikom rasponu uređaja. Tako danas Android operacijski sustav možemo vidjeti na pametnim telefonima, tabletima, pametnim satovima, pametnim televizorima, u automobilima i drugim pametnim uređajima. Razlozi zbog kojih se ovaj operacijski sustav koristi su njegova otvorenost, prenosivost, sigurnost i jedinstvene značajke [7].

2.2. iOS operacijski sustav

iPhone Operacijski Sustav (iPhone Operating System) ili skraćeno iOS je operacijski sustav koji je razvila tvrtka Apple Inc. koji pokreće većinu njihovih uređaja kao što su iPhone, iPad i iPod. Ovaj operacijski sustav predstavljen je iste godine kao i njegov glavni konkurent, 2007. na Macworld konferenciji gdje je istovremeno predstavljen prvi uređaj koji će ovaj novi operacijski sustav pokretati, iPhone. I iOS i iPhone su u javnost pušteni u lipnju 2007. godine. iOS je prvotno zamišljen kao operacijski sustav koji će pokretati samo Apple iPhone uređaje, ali napretkom tehnologije i samom promjenom politike tvrtke, iOS je prenamijenjen kako bi se mogao koristiti i na ostalim Apple uređajima [8].

Prva verzija ovog operacijskog sustava nije imala podršku za izradu nativnih aplikacija. Razvojni programeri su morali razvijati web aplikacije koje bi se ponašale slično kao i nativne aplikacije koje je razvio sam Apple. Od tog trenutka je zapravo krenuo razvoj platformi za razvoj hibridnih mobilnih aplikacija. Sa kasnijim verzijama iOS-a, točnije verzijom 2.0, Apple je omogućio razvojnim programerima razvoj nativnih iOS aplikacija. Verzija 2.0 iOS-a je u javnost puštena 2008. godine i sa njom su dakle pušteni i alati za izradu aplikacija. Na početku su se iOS aplikacije razvijale s Objective-C programskim jezik. Kasnije je Apple razvio svoj jezik po uzoru na Objective-C i nazvao ga Swift. Danas se većina iOS aplikacija razvija u Swift programskom jeziku zbog njegove jednostavnosti i samih značajki ovog jezika [9].

2.3. Hibridne mobilne aplikacije

Hibridne mobilne aplikacije su mobilne aplikacije koje nisu razvijene u nativnom okruženju za izradu mobilnih aplikacija na specifičnoj platformi, a to su kao što je rečeno, Java ili Kotlin za Android operacijski sustav, te Objective-C ili Swift za iOS. Razvoj hibridnih mobilnih aplikacija započeo je izlaskom prve verzije iOS-a i tim putem su se započele razvijati hibridne mobilne aplikacije, kao kombinacija pristupa nativnih i web aplikacija. „Kod ovakvog pristupa dio aplikacije se razvija kao i u slučaju web aplikacije – u HTML5, CSS i JavaScript-u, dok se pristup API-ju operacijskog sustava vrši pomoću dodatnih nativnih dijelova, koji su prilagođeni da se mogu koristiti pomoću web tehnologija“ [10]. Nativni dijelovi su najčešće napisani od strane neke platforme za razvoj hibridnih mobilnih aplikacija. Dvije trenutno najpoznatije takve platforme su React Native i Flutter. Kod ove dvije platforme postoje vrlo bitne razlike između načina izvođenja mobilnih aplikacija koje će kasnije biti objašnjene.

React Native je platforma za izradu hibridnih mobilnih aplikacija koju je razvila tvrtka Facebook. Glavni programski jezik na kojemu se zasniva čitav rad ove platforme je JavaScript kojeg platforma prevodi te korisniku dostavlja sučelje koje nalikuje nativnom sučelju mobilne aplikacije [11]. Ova platforma omogućava izvođenje JavaScript koda na mobilnim uređajima preko takozvanog mosta (engl. Bridge) koji povezuje pametni telefon i JavaScript kod i omogućava njihovu međusobnu komunikaciju [12]. Taj most predstavlja srž ove platforme, bez njega se ne bi mogla izvoditi niti jedna hibridna mobilna aplikacija razvijena na ovoj platformi, te je on ujedno i najveći problem platforme jer ograničava samu komunikaciju između mobilne aplikacije i pametnog telefona i usporava izvođenje aplikacije na pametnom telefonu.

S druge strane imamo drugu platformu za izradu mobilnih aplikacija, Flutter. Flutter je razvila tvrtka Google kao njihov odgovor na razvoj hibridnih mobilnih aplikacija. Ova platforma rješava probleme i donosi mnogo novih značajki u svijet razvoja hibridnih mobilnih aplikacije koje će biti kasnije detaljno opisane.

2.4. Flutter platforma

Kao osnova za razvoj ove aplikacije izabrana je platforma za izradu hibridnih mobilnih aplikacija Flutter. Kao što je već spomenuto, Flutter je razvila tvrtka Google. Ova platforma je predstavljena 2015. godine na Dart developer samitu. Na toj konferenciji Google je najavio Flutter kao platformu koja će omogućiti izvođenje mobilnih aplikacija na Android operacijskom sustavu u 120 sličica po sekundi bez korištenja Java programskog jezika [13]. Ali javnost ovu platformu nije mogla koristiti sve do svibnja 2017. godine kada je izašla prva alfa verzija platforme [14]. Tada je Google odlučio da će ova platforma biti otvorenog koda tako da bilo tko može pomoći pri razvoju platforme. Tako se platforma se razvijala kroz godine da bi napokon u prosincu 2018. godine izašla prva stabilna verzija [15]. Od tada, popularnost ove platforme je svakim danom sve veća zbog njezinih odlika te performansi koje nudi.

Glavna prednost i razlika između Fluttera i ostalih platformi za razvoj hibridnih mobilnih aplikacija je ta da Flutter nema ranije spomenuti most kao što je slučaj kod React Native platforme. Kod ove platforme, kod se piše u programskom jeziku Dart, a taj se kod na mobilnom telefonu prikazuje pomoću Flutter pokretača (engl. Flutter Engine). Flutter pokretač je razvijen u C++ programskom jeziku i koristi se za prikazivanje 2D aplikacija. Nadalje, taj pokretač ima direktnu komunikaciju s uređajem na kojem se izvodi, zbog toga se ne gubi na brzini i performansama Flutter aplikacija. Uz to, pokretač se brine o svim različitostima operacijskog sustava na kojemu

se pokreće i to je razlog zbog kojeg aplikacije razvijene s Flutter platformom izgledaju i ponašaju se gotovo identično kao i nativne aplikacije [16].

Glavni elementi svih Flutter aplikacija su widgeti. Svaki widget je deklaracija jednog dijela korisničkog sučelja. Za razliku od drugih platformi koje imaju različite poglede, kontrolere za poglede, izglede i druga svojstva, Flutter ima samo widgete. S widgetima se mogu napraviti različite stvari, kao što su dodavanje gumba ili menija, promjena fonta ili boje slova, dodavanje ispunjenja i tako dalje. Widgeti se kroz aplikaciju slažu u strukturu stabla, svaki novi widget koji dodamo u aplikaciju će se nalaziti na dnu stabla i bude preuzeo sva svojstva widgeta koji se nalaze iznad njega [16]. S programskim kodom 3.1 dan je primjer jednog widgeta koji se u Flutter platformi koristi za prikaz teksta na ekranu. Taj widget će na ekranu pokazati tekst „Primjer“, veličine slova 18, crvene boje.

```
Text(
  "Primjer",
  style: TextStyle(
    fontSize: 18,
    color: Colors.red,
  ),
);
```

Programski kod 3.1: Flutter widget za prikaz teksta na ekranu

Zbog svoje jednostavnosti, brzine razvoja mobilnih aplikacija, načina izvođenja aplikacija i mnogih drugih prednosti, ova platforma za razvoj hibridnih mobilnih aplikacija je sve češći izbor kako pojedinaca tako i tvrtki. Flutter kao platforma se razvija i dalje, Google-ovi inženjeri i ostali razvojni programeri koji žele pomoći pri razvoju ove platforme svakodnevno ispravljaju greške, dodaju nove widgete ili jednostavno pišu nove nativne dijelove za platformu kako bi Flutter-ov pokretač mogao što bolje komunicirati i iskoristiti uređaj na kojemu se izvodi. Trenutno zadnja verzija ove platforme je 1.14.6 koja je izašla u drugom mjesecu 2020. godine. U trenutku pisanja ovog rada, Flutter podržava izradu mobilnih aplikacija za Android i iOS operacijskih sustava, web aplikacija, te aplikacija za osobna računala s Microsoft Windows, MacOS i Linux operacijskim sustavima [16]. Na programskom kodu 3.2 prikazana je osnovna aplikacija razvijena s Flutter platformom.

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Hello World App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('Hello World App'),
        ),
        body: Center(
          child: Text('Hello World App'),
        ),
      ),
    );
  }
}
```

Programski kod 3.2: Osnovna aplikacija razvijena pomoću Flutter platforme

2.5. Programski jezik Dart

Dart je programski jezik optimiziran kako bi se aplikacije razvijene pomoću njega mogle izvoditi na širokoj paleti operacijskih sustava. Ovaj jezik razvila je tvrtka Google i koristi se za izradu mobilnih aplikacija, aplikacija za osobna računala, web aplikacija ili za kod koji će se izvršavati na poslužiteljima. Dart je prvi put predstavljen javnosti na GOTO konferenciji u Danskoj 2011. godine [17]. Kako Google većinu svojih aplikacija ima primarno kao web aplikacije, željeli su razviti programski jezik koji će im pomoći da se te web aplikacije izvode još bolje i izgledaju ljepše. Tako je nastao Dart i to je bio prva svrha ovog programskog jezika, izrada web aplikacija [18]. Prva stabilna verzija ovog programskog jezika izašla je dvije godine nakon prvog predstavljanja, u studenom 2013. godine [19]. Iako Dart nikada nije u potpunosti zaživio kao jezik za izradu web aplikacija, pa je Google odlučio prenamijeniti u druge svrhe.

Tako je Dart postao jedna od glavnih prednosti Flutter platforme kao programski jezik u kojemu se razvijaju aplikacije. Ovaj objektno orijentirani jezik ima sve odlike modernog programskog jezika koji olakšava posao razvojnom inženjeru, tako da se lako privikne na njega zbog njegove sintakse koja je vrlo slična programskom jeziku C, brine se o „skupljanju smeća“ (engl. Garbage collecting), podržava sučelja, apstraktne klase i mnogo više. Godine 2015. Dart je dobio 1.9 nadogradnju koja je u potpunosti izbacila sve stare ideje ovog programskog jezika te je

Dart dobio mogućnost prevođenja koda u JavaScript. I kao takav se je počeo koristiti za Flutter platformu [19].

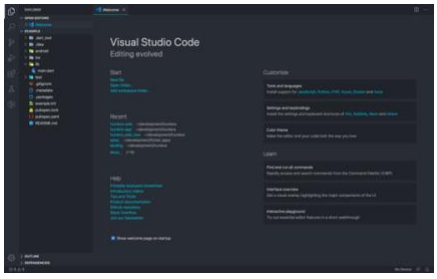
Ovaj programski jezik je i dalje razvija, 2018. izašla je 2.0 verzija koja je donijela podršku za zvuk, a najnovija verzija je 2.7. Ta verzija donijela je poneke nove značajke i te mogućnost da se Dart kod može prevesti u nativan kod za operacijski sustav za koji se aplikacija [20]. Primjer jednostavne aplikacije koja ispisuje „Hello world“ tekst u konzolu napisan pomoću Dart programskog jezika može se vidjeti na isječku programskog koda 3.3.

```
void main() {
  print('Hello, World!');
}
```

Programski kod 3.3: Program koji ispisuje tekst "Hello world" u konzolu napisan pomoću Dart-a

2.6. Program za pisanje koda Visual Studio Code

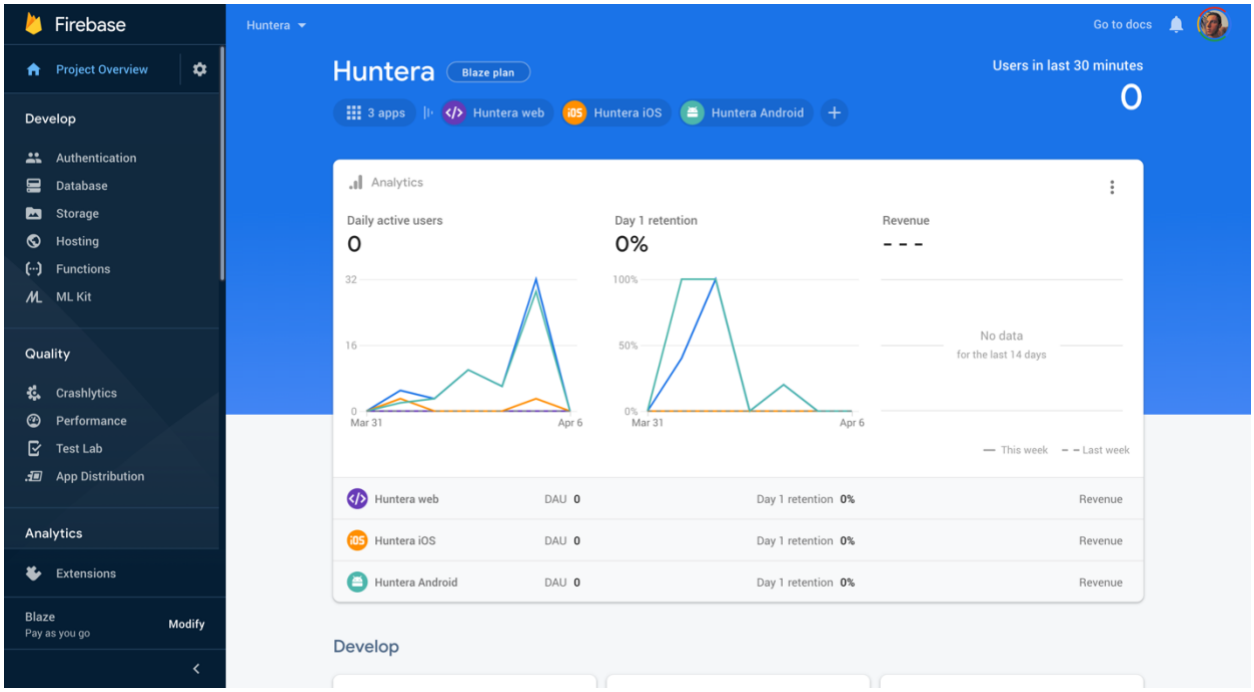
Visual Studio Code prikazan na slici 3.1 je besplatni uređivač teksta otvorenog koda koji je razvila tvrtka Microsoft za Windows, Linux i MacOS operacijske sustave. „Ima integriranu podršku za ispravljanje grešaka u kodu, alate za verzioniranje koda (Git control), označavanje sintakse, inteligentno nadopunjavanje te refaktoriranje koda“. Uz to, ovaj uređivač teksta ima mogućnost dodavanja dodatnih paketa koji mu daju podršku za nove vrste datoteka, promjenu izgleda programa, mogućnost formatiranja datoteka prema određenom formatu i slično. U njemu možemo pisati bilo koji programski jezik dokle god za taj programski jezik postoji podrška u obliku paketa. Visual Studio Code razvijen je pomoću platforme Electron koji se uz izradu aplikacija za osobna računala koristi i za pokretanje Node.js aplikacija na istima. U 2018. godini provedena je anketa gdje su razvojni inženjeri glasali koji uređivač teksta ili program za programiranje koriste, 35% ispitanih je odgovorilo da koristi Visual Studio Code što je ovaj uređivač teksta svrstalo u najkorišteniji program za programiranje [11].



Slika 3.1: Početna stranica Visual Studio Code-a

2.7. Servis za aplikacije Firebase

„Firebase je web i mobilna platforma s alatima i infrastrukturom osmišljenim kako bi pomogla programerima izgradnju visokokvalitetnih aplikacija“. Sastoji se od prijeko potrebnih alata za izradu web i mobilnih aplikacija koje bi inače programer morao razvijati sam na svojim poslužiteljima. Firebase (slika 3.2) je na početku pružao uslugu baze podataka u realnom vremenu. Ta baza podataka je omogućivala programerima implementaciju baze podataka u njihove aplikacije koju će zatim moći koristiti više korisnika u isto vrijeme. Kroz vrijeme, Firebase je polako dodavao nove alate i tako dobio pažnju Google-a koji je zatim 2014. godine i kupio tvrtku.



Slika 3.2: Početna stranica Firebase konzole

Kako je Google preuzeo vlasništvo nad tvrtkom i imenom Firebase, na razvoju platforme počeli su raditi Google-ovi inženjeri i ova platforma je postajala sve popularnija među programerima [21]. Za izradu ove aplikacije korišteni su slijedeći alati od Firebase-a:

- Auth – alat koji pruža autentifikaciju korisnika. Korisnički podaci će biti osigurani i spremljeni na Firebase-ove poslužitelje. Ovaj alat također nudi autentifikaciju kroz druge servise kao što su Google, Facebook, Twitter i slični.

- Cloud Functions – omogućava programiranje okidača koji će izvršiti određenu funkciju nakon što se ispuni neki uvjet. Taj uvjet može biti novi unos u bazu podataka, izmjena u bazi podataka, posjeta web adresi…
- Cloud Firestore – baza podataka koja omogućava pristup podacima pametnim telefonima, web aplikacijama i poslužiteljima. Također nudi podršku ako klijent nema pristup internetu.
- Cloud Storage – služi za pohranu podataka (slika, zvuka, teksta ili bilo kojih drugih datoteka).
- Hosting – poslužitelj koji se može konfigurirati kako god korisnik želi i služi za podizanje web stranice ili aplikacije.
- Realtime Database – kao što je već ranije rečeno, ovo je prvi alat koji je Firebase nudio svojim korisnicima, a omogućava pristup podacima više korisnika od jednom.
- Crashlytics – alat koji kada se integrira u aplikaciju svako rušenje aplikacije šalje na Firebase konzolu s logovima koji su doveli do rušenja aplikacije.
- Analytics – kao što samo ime ovog alata kaže, služi za praćenje analitike aplikacije. Prikuplja informacije korisnika aplikacije kao što su lokacija, verzija aplikacije, operacijski sustav, dobna skupina, spol itd. [22]

3. RAZVOJ APLIKACIJE

Aplikacija se sastoji od četiri glavna dijela, to su: dio s korisnikovim trofejima, dio s informacijama o korisničkom računu, dio gdje se mogu pogledati svi javnim trofeji od drugih korisnika te dio s informacijama o aplikaciji kao što su politika privatnosti, uvjeti korištenja i slično. Uz to postoji i dio za autentifikaciju korisnika, za prijavu postojećih korisnika i registraciju novih. Kada korisnik otvori aplikaciju prvi puta nakon instalacije, otvorit će se zaslon prikazan na slici 4.1 gdje korisnik vidi logotip i ime aplikacije, gumbove za prijavu i registraciju, gumbovi za prijavu pomoću nekih od servisa, u ovom slučaju Google i Facebook te gumb koji vodi do stranice na internetu gdje se nalaze uvjeti korištenja.

Slika 4.1: Početni zaslon aplikacije nakon instalacije

3.1. Prijava korisnika

Ukoliko korisnik pritisne tipku „PRIJAVA“, otvorit će mu se zaslon prikazan na slici 4.2 s dva polja za unos. Prvo polje traži od korisnika da unese svoju email adresu s kojom se je registrirao u aplikaciju. Drugo polje zahtjeva od korisnika da unese lozinku koju je koristio prilikom

registracije. Gumb „PRIJAVA“ na ovom zaslonu ima sivu boju zbog toga što njega nije moguće pritisnuti dok se ne ispune potrebni uvjeti. Uvjeti da gumb postane aktivan su da je unesen ispravan format email adrese i da je unesena lozinka koja ima 6 ili više znakova.

Slika 4.2: Zaslon za prijavu korisnika u aplikaciju

Ukoliko korisnik ne unese ispravnu email adresu ili prekratku lozinku, aplikacija će ispod polja ispisati poruku crvenim slovima kako nisu ispunjeni uvjeti koji bi omogućili prijavu korisnika. Nakon što je korisnik unesao podatke koji ispunjavaju uvjete aplikacije, gumb „PRIJAVA“ će poprimiti zelenu boju kako bi se korisniku dalo do znanja da je gumb sada aktivan. Klikom na gumb poziva se metoda za prijavu korisnika u klasi HunteraApi koja sadrži sve metode za komunikaciju s Firebase servisom prikazana na isječku programskog koda 4.1. Ta metoda će kao rezultat vratiti podatke prijavljenog korisnika ili predefiniranu konstantu tipa FirebaseError. Ukoliko je metoda vratila konstantu tipa FirebaseError, korisniku će na zaslon biti ispisana poruka o grešci koja se dogodila. To može biti poruka o nepostojećoj email adresi u bazi podataka, netočnoj lozinki koja se veže uz tu email adresu ili nedostatak povezanosti s internetom.

```
Future<dynamic> login(String email, String password) async {
  var user, firebaseError;
  try {
    user = (await _auth
            .signInWithEmailAndPassword(
              email: email,
              password: password,
            )
            .timeout(Duration(seconds: 5)))
        .user;
  } catch (e) {
    print(e);
    if (e.toString().contains("USER_NOT_FOUND"))
      firebaseError = FirebaseError.USER_NOT_FOUND;
    else if (e.toString().contains("WRONG_PASSWORD"))
      firebaseError = FirebaseError.WRONG_PASSWORD;
    else
      firebaseError = FirebaseError.NO_INTERNET;
  }

  return firebaseError ?? user;
}
```

Programski kod 4.1: Metoda koja služi za prijavu korisnika u aplikaciju

Nakon uspješne prijave, korisniku se prikazuje zaslon s njegovim trofejima. U isto vrijeme asinkronom metodom se u memoriju uređaja zapisuju podaci o prijavljenom korisniku kako bi od idućeg pokretanja aplikacije korisnik bio automatski prijavljen u aplikaciju. Tako će korisnik ostati prijavljen u aplikaciju sve dok se ne odluči sam odjaviti ili se izbrišu podaci aplikacije koji su spremljeni na pametni telefon. Podaci koji se spreme na uređaj nakon prijave su na iOS uređajima zaštićeni u takozvanom KeyChainu, a na Android operacijskom sustavu su kriptirani AES enkripcijom kako ne bi došlo do zlouporabe podataka.

Također, korisnik na zaslonu za prijavu ima gumb „ZABORAVLJENA LOZINKA?“ koji omogućava postavljanje nove lozinke putem email adrese koja je korištena prilikom registracije korisničkog računa. Korisnika se klikom na gumb zahtjeva da unese email adresu svog korisničkog računa. Nakon unosa na unesenu email adresu dobiva poruku s URL-om za postavljanje nove lozinke.

3.2. Registracija korisnika

Postupak registracije korisnika se odvija u nekoliko koraka. Razlog tomu je kako ne bi jedan zaslon bio prepun informacija za korisnika od jednom. Na ovaj način korisnik dobiva dojam kako je registracija jedan kraći postupak. Koraci registracije korisnika su:

- Unos email adrese, lozinke i potvrda lozinke.
- Potvrda email adrese.
- Unos punog imena i prezimena te datuma rođenja.
- Dodavanje slike profila ukoliko korisnik to želi.
- Povezivanje korisnikovog oružja s Huntera računom ako korisnik želi.

Nakon što korisnik unese svoju email adresu i dva puta unese lozinku, metoda za registraciju u HunteraApi klasi će napraviti novi korisnički račun s unesenim podacima na Firebase servisu. Nakon što je račun uspješno napravljen, korisniku će se animirati prijelaz do drugog zaslona laganim pomakom s desna na lijevo. Na drugom zaslonu vidljivom na slici 4.3 od korisnika će se tražiti da potvrdi svoju email adresu. Ovaj korak je nužan za obaviti jer u protivnom aplikacija neće pustiti korisnika u daljnji proces registracije. Potvrda email adrese je nužna kako bi se spriječila zlouporaba izrade korisničkih računa koji bi se koristili protivno uvjetima korištenja ove aplikacije. Čak iako korisnik obriše aplikaciju sa svog pametnog telefona, ponovno ju instalira i proba se prijaviti s istim korisničkim računom, aplikacija će od njega tražiti potvrdu email adrese.

Nakon što je email adresa potvrđena, istom animacijom će korisnik prijeći na idući zaslon koji će on njega tražiti unos punoga imena i prezimena te datuma rođenja. Korisnik ove podatke također mora ispuniti prije nastavka procesa registracije. Ovi podaci su kasnije potrebni u aplikaciji u slučaju ako će se korisnikovi trofeji prikazivati drugim korisnicima, kako bi drugi korisnici znali tko je odstrijelio trofej.

Od tog trenutka, korisnik daljnje informacije nije nužan dostaviti. Pod te ostale informacije spadaju slika profila koja se također prikazuje ostalim korisnicima aplikacije u slučaju da korisnik ima javnih trofeja na svom profilu te podaci o oružju. Podaci o oružju sadrže kategoriju oružja, proizvođača i model oružja, kalibar i optiku ukoliko je ista montirana na oružje. Korisnik ove informacije može dodati na svoj profil u bilo kojem trenutku kroz sučelje aplikacije. Dok slika profila nije potrebna niti u jednom trenutku tijekom korištenja aplikacije i služi samo kao vizualna reprezentacija vlasnika trofeja ostalim korisnicima, podaci o oružju su bitni jer bez povezanog oružja na svom profilu, korisnik neće biti u mogućnosti dodati novi trofej u aplikaciju.

Nakon što je korisnik unesao sve potrebne informacije, iste se pomoću metode pohranjuju u bazu podataka na Firebase servisu u tablicu *user_data*. Kada se podaci uspješno pohrane u bazu podataka, korisnik će biti prebačen na zaslon s trofejima u aplikaciji. Kako je to novo kreirani račun, niti jedan trofej neće biti vidljiv, pa će korisnika dočekati tekst koji govori kako trenutno nema niti jedan uneseni trofej i gumbom za dodavanje trofeja.

Slika 4.3: Zaslon koji traži potvrdu email adrese prije nastavka

3.3. Zasloni s trofejima

Kada je korisnik ulogiran u aplikaciju, prvi zaslon koji će vidjeti je zaslon s korisnikovim trofejima. Prilikom svakog ulaska u aplikaciju aplikacija će pozvati metodu (programski kod 4.2) za dohvaćanje svih trofeja za prijavljenog korisnika. Ukoliko metoda iz nekog razloga ne izvrši zahtjev na siguran način, to znači da pametni telefon nema aktivnu vezu s internetom. Ako se to dogodi, Firebase Firestore baza podataka će vratiti rezultate koji su spremljeni u pričuvnu memoriju. U slučaju da nikakvi podaci nisu spremljeni u pričuvnoj memoriji, korisniku će se na zaslon ispisati prikladna poruka kako bi se korisniku dalo do znanja zbog čega se trofeji ne

učitavaju. Također Firebase Firestore baza podataka će i ako postoji aktivna veza s internetom, potražiti podatke u pričuvnoj memoriji kako bi korisniku podaci bili što brže servirani.

```
Future<List<Trophy>> getUserTrophyList() async {
  FirebaseUser user = await _auth.currentUser();
  List<Trophy> trophies = [];

  try {
    await _firestore
        .collection('trophies')
        .orderBy("date", descending: true)
        .where("userId", isEqualTo: user.uid)
        .getDocuments()
        .then((QuerySnapshot snapshot) {
      snapshot.documents.forEach((DocumentSnapshot documentSnapshot) {
        String id = documentSnapshot.documentID;
        trophies.add(Trophy.fromJson(documentSnapshot.data, id));
      });
    });
  } catch (e) {
    print(e);
  }

  return trophies;
}
```

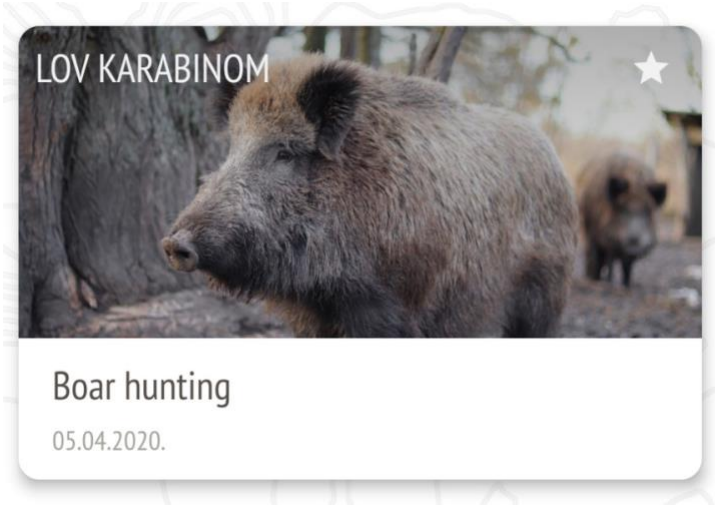
Programski kod 4.2: Metoda za dohvaćanje trofeja iz baze podataka

Kao što je vidljivo iz programskog koda 4.2, metoda za dohvaćanje trofeja pretražuje trofeje u bazi podataka koji imaju parametar userId jednak trenutno prijavljenom korisniku u aplikaciji. Nakon što ih pronađe, prolazi kroz svaki i od JSON rezultata proizvodi objekte klase Trophy (programski kod 4.3) te ih sprema u niz Trophy objekata. Kada je metoda prošla kroz sve rezultate baze podataka i napravila objekte, niz tih objekata se vraća kako bi aplikacija mogla prikazati obrađene podatke korisniku.

Nakon što se niz objekata Trophy vrati na glavni zaslon za prikaz trofeja, svaki od objekata se koristi za izradu materijalno dizajniranih kartica prikazanih na slici 4.4 te korisnik napokon može vidjeti svoje trofeje i interaktirati s njima. Svaka kartica trofeja ima istaknutu sliku, kategoriju lova (ovisi o kategoriji oružja s kojom je trofej odstrijeljen), naslov, datum te ikonu favorita. Ikona favorita je vizualna reprezentacija korisniku koja mu govori da li je spremio određeni trofej u svoje favorite ili ne. Klikom na tu ikonu, korisnik dodaje ili briše trofej iz favorita. Ukoliko korisnik klikne bilo gdje drugdje na karticu trofeja, otvorit će se zaslon sa svim detaljima i svim slikama trofeja.

```
class Trophy {
  String id;
  final String userId;
  final String username;
  final String userImage;
  bool private;
  final String title;
  List<String> imagesUrl;
  final String category;
  DateTime date;
  bool favorite;
  final String location;
  final String animal;
  final String animalTranslated;
  final Weapon weapon;
  final String bullet;
  final String notes;
```

Programski kod 4.3: Klasa Trophy i njezini parametri

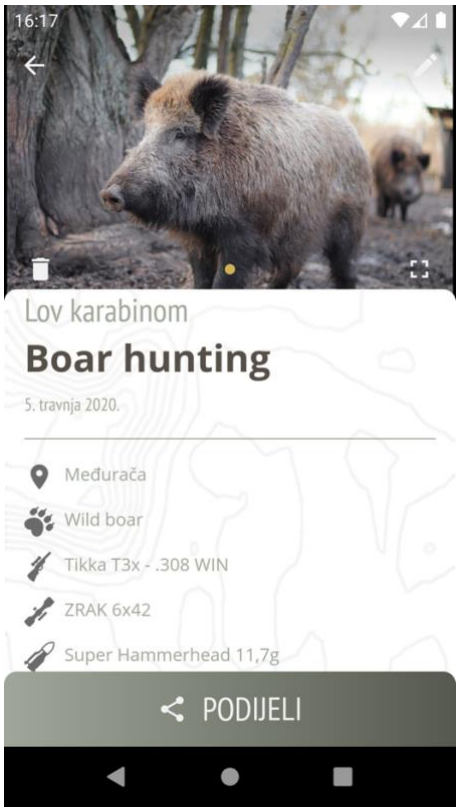


Slika 4.4: Kartica trofeja

Osim glavnog zaslona gdje korisnik može pregledavati svoje trofeje, u aplikaciji i postoji takozvani „ISTRAŽI“ zaslon. Taj zaslon je dostupan samo onim korisnicima koji plaćaju mjesečnu pretplatu za pristup takvom sadržaju. Zaslon „ISTRAŽI“ pretplaćenim korisnicima omogućava da pregledavaju sve javne trofeje drugih korisnika ove aplikacije. Korisnik se može pretplatiti za pristup ovom sadržaju kroz sučelje aplikacije, na zaslonu s informacijama profila koji će biti opisan kasnije.

3.4. Zaslon s detaljima trofeja

Ovaj zaslon (slika 4.5) je jedan od najbitnijih zaslona u aplikaciji. To je zaslon na kojemu bi korisnici trebali provoditi dosta vremena pregledavajući fotografije trofeja i detalje s kojima je trofej odstrijeljen. Kao i cijela aplikacija, i ovaj zaslon je materijalno i minimalistički dizajniran. Sadrži samo najosnovnije potrebne informacije o svakom trofeju. Prikazuje sve fotografije koje je korisnik postavio za taj trofej te informacije o oružju, lokaciji, datumu i slične. Zaslon za razliku od ostalih ima podršku i za horizontalni i vertikalni položaj pametnog telefona. Ukoliko se pametni telefon nalazi u horizontalnoj orijentaciji, detalji trofeja će se sakriti i samo će fotografije ostati vidljive. Na dnu zaslona je vidi gumb „PODIJELI“, on služi za jednostavno dijeljenje trofeja. Klikom na taj gumb, otvara se izbornik za odabir aplikacije s kojom se želi podijeliti trofej. Aplikacija je generirala sadržaj koji će se poslati odabranom aplikacijom kao i URL za pregled tog trofeja putem internetskog preglednika ukoliko primatelj nema instaliranu aplikaciju na svom pametnom telefonu. Uz to, na ovom zaslonu su i neki funkcijski gumbi koji služe za brisanje odnosno uređivanje trofeja.



Slika 4.5: Zaslon s detaljima trofeja

3.5. Logika dodavanja trofeja

Jedna od neizostavnih značajki aplikacije je dodavanje trofeja u aplikaciju. Za tu funkcionalnost je dizajniran potpuno novi zaslon koji ima samo jedan zadatak. Prikupiti sve informacije o trofeju i predati ih metodi koja će iste pohraniti u bazu podataka. Tako ovaj zaslon ima devet polja za unos podataka kako bi trofej imao što više informacija. Od tih devet polja za unos podataka, sedam njih je obavezno za popuniti. Ukoliko jedno od tih sedam polja nije ispunjeno, gumb za dodavanje trofeja neće biti aktivan i bude imao sivu boju. U trenutku kada korisnik ispuni sve preduvjete za dodavanje trofeja u aplikaciju, gumb će postati aktivan i poprimit će zelenu boju. U trenutku kada korisnik pritisne gumb za dodavanje trofeja, poziva se metoda za slanje unesenih informacija bazi podataka (programski kod 4.4).

```
Future<dynamic> addTrophy(Trophy trophy) async {
  try {
    final inserted = await _firestore
        .collection('trophies')
        .add(trophy.toJson())
        .timeout(Duration(seconds: 5));
    return inserted.documentID;
  } catch (e) {
    print(e);
    return FirebaseError.NO_INTERNET;
  }
}
```

Programski kod 4.4: Metoda za dodavanje trofeja

Metoda kao parametar prima objekt tipa Trophy, pretvara objekt u JSON tip podatka i sprema ga u tablicu *trophies*. Prije toga, druga metoda je korištena za slanje svih odabranih fotografija na Firebase Storage alat. Ta metoda, ukoliko je prijenos fotografija obavljen uspješno, vraća niz String tipova podataka koji predstavljaju URL putanju do fotografije na Firebase Storage poslužitelju. Taj niz URL-ova se dodaje u objekt trofeja i na kraju predaje metodi za dodavanje trofeja. Metoda će vratiti ID novo unesenog trofeja ukoliko je sve prošlo uredu ili grešku tipa FirebaseError ako pametni telefon nije spojen na Internet.

Prilikom dodavanja trofeja moraju se prenositi fotografije na poslužitelje, ovaj proces zbog toga može potrajati. Iz tog razloga, čim aplikacija krene s obradom unesenih podataka, šalje se obavijest na pametne telefone kako bi korisnik znao da se trofej trenutno šalje na poslužitelje i da

se može dalje nastaviti služiti aplikacijom. Jednom kada se trofej prenese i spremi u bazu podataka, obavijest će se sama maknuti s korisnikovog pametnog telefona.

3.6. Korisnički profil

Iznimno je važno pružiti korisniku mogućnost izmjene korisničkih podataka unutar same aplikacije. Zbog toga je napravljen „PROFIL“ zaslon koji je uvijek lako dostupan korisnicima. Na tom zaslonu korisnici prvenstveno vide svoju sliku profila, ime i prezime te broj do sada unesenih trofeja. Nadalje korisnik ima nekoliko mogućnosti što dalje napraviti sa svojim profilom.

Prva opcija je opcija „HUNTERA PREMIUM“. To je zaslon na kojemu korisnik može provjeriti trenutno stanje svoje pretplate u aplikaciji ili ako korisnik nema aktivnu pretplatu, izabrati duljinu pretplate i kupiti pretplatnički paket. Kupnjom pretplatničkog paketa, korisnik istog trenutka kada Google ili Apple (ovisno o operacijskom sustavu na kojemu se aplikacija u tom trenutku izvodi) potvrde uspješnu transakciju dobiva pristup punoj verziji aplikacije.

Druga mogućnost je pregled i uređivanje oružja povezanog sa korisnikovim računom. Na ovom zaslonu korisnik može pregledati oružje koje je povezao sa svojim računom, izmijeniti poneke detalje ili greške, te brisati postojeće oružje i dodavati novo. Kod dodavanja oružja postoje četiri polja za unos, polje za kategoriju oružja, proizvođača i model, kalibar oružja i optiku. Sva polja su obavezna za ispuniti osim optike, zato jer postoje lovci koji love mehaničkim nišanom. U slučaju da optika za neko od oružja nije unesena i korisnik unese novi trofej odstrijeljen tim oružjem, onda će se u podacima trofeja pod optikom pisati mehanički nišan.

Promjena lozinke za pristup korisničkom računu također mora biti moguća u bilo kojem trenutku. Zato je ova mogućnost postavljena kao treća opcija na ovom zaslonu. Postoji nekoliko uvjeta kako bi korisnik bio u mogućnosti promijeniti svoju trenutnu lozinku. Prvi je taj da je korisnik u aplikaciju registriran putem email adrese, jer u protivnom korisnik neće niti imati postavljenu lozinku. Nadalje, korisnik mora znati svoju trenutačnu lozinku jer će aplikacija od korisnika zahtijevati unos trenutne lozinke. Ako se unese netočna trenutna lozinka, aplikacija će o toj grešci obavijestiti korisnika. Osim polja za unos trenutne lozinke postoje još dva polja. Jedno polje je za unos nove lozinke, a drugo za potvrdu nove lozinke. Nova lozinka također mora zadovoljiti uvjet da ima minimalno 6 znakova, a potvrda nove lozinke znači da korisnik u drugo polje mora unesti istu lozinku kao i u prethodnom polju. Sve dok se svi uvjeti ne ispune, gumb za promjenu lozinke neće postati aktivan. Klikom na gumb poziva se metoda za promjenu lozinke prikazana na programskom kodu 4.5.

```
Future<dynamic> changePassword(
    String currentPassword, String newPassword) async {
  final user = await _auth.currentUser();
  try {
    await _auth.signInWithEmailAndPassword(
        email: user.email, password: currentPassword);

    await user.updatePassword(newPassword);
  } catch (e) {
    print(e);
    if (e.toString().contains("ERROR_WRONG_PASSWORD"))
      return FirebaseError.WRONG_PASSWORD;
    else
      return FirebaseError.NO_INTERNET;
  }

  return true;
}
```

Programski kod 4.5: Metoda za promjenu lozinke

Tu spada promjena slike profila, imena i prezimena, datuma rođenja i email adrese. Korisnik u bilo kojem trenutku može promijeniti sve osim email adrese. Email adresu može promijeniti samo ukoliko nema povezane neke od društvenih profila (Google i Facebook). Također, ako se korisnik duže vrijeme nije ponovno prijavljivao u aplikaciju i želi promijeniti email adresu, aplikacija će takvom korisniku prije promjene prikazati dijaloški okvir prikazan na slici 4.6 gdje će se od korisnika tražiti unos trenutne lozinke. Samo ukoliko korisnik unese točnu trenutnu lozinku, njegova email adresa će biti promijenjena i bit će poslana poruka za potvrdu email adrese. Korisnik neće moći dodavati nove trofeje u aplikaciju sve dok njegova email adresa ne bude potvrđena. Sve ostale informacije korisnik može promijeniti bez ikakvih dodatnih provjera. Uz sve to, kao dodatna mogućnost na ovom zaslonu je dodana mogućnost povezivanja Google i Facebook računa s postojećim korisničkim računom.

Slika 4.6: Dijaloški okvir za autentifikaciju korisnika

Od svih tih mogućnosti, ostala je još jedna podosta važna mogućnost ovog ekrana, a to je funkcija odjave. Pritiskom na gumb „ODJAVA“ aplikacija će obrisati sve podatke o trenutno prijavljenom korisniku u aplikaciji i korisniku prikazati zaslon s gumbovima za prijavu i registraciju.

3.7. Dio aplikacije s informacijama

Najjednostavniji zaslon u cijeloj aplikaciji je zaslon s informacijama. Ovaj zaslon ima dva gumba koji vode do web adrese s politikom privatnosti aplikacije odnosno uvjeta korištenja. Treća mogućnost je gumb za kontakt. Pomoću tog gumba korisnik može pronaći sve informacije vezane za kontakt razvojnog programera ove aplikacije kako bi ga mogao kontaktirati u vezi bilo čega, grešaka u aplikaciji, zahtjevom za dodavanjem novih mogućnosti i slično. Na kraju ekrana nalazi se logotip razvojnog programera. Klik na logotip korisnika odvodi na web stranicu razvojnog programera.

4. ZAKLJUČAK

Na svjetskom tržištu ima sve više pametnih telefona kao i korisnika takvih uređaja. Velika većina ljudi provodi barem dva sata dnevno na svojim pametnim telefonima. Lovci nisu izuzetak, sve je više mladih lovaca koji su itekako u trendu s novim tehnologijama i svakodnevno koriste svoje pametne telefone. Do sada lovci nisu imali nikakvu sličnu aplikaciju koja bi im služila kao pohrana, sinkronizacija i platforma za dijeljenje ulovljenih trofeja. Čak i danas kada je gotovo sve pohranjeno u „oblaku“ postoji velika mogućnost da se poneke slike zauvijek izgube, na ovaj način čim se trofej objavi, pohranjen je zauvijek u bazi podataka i lovac će ga uvijek moći pogledati i podijeliti s drugima. Tako je nastala ideja za ovu aplikaciju.

Ova aplikacija je podosta kompleksna zbog mnogobrojnih zaslona koji moraju biti sinkronizirani jedan s drugim. To je bio najveći izazov prilikom razvoja ove aplikacije. Flutter platforma je relativno nova i još ne postoji najbolji način za sinkronizaciju podataka između više zaslona. U ovom slučaju je taj problem riješen s *dependency injection* oblikovnim obrascem. Nadalje pošto aplikacija cilja na globalno tržište, jako je važno da je ista lokalizirana na nekoliko svjetskih jezika. To je drugi nedostatak platforme koja je odabrana za razvoj ove aplikacije. Ova značajka na ovoj platformi također još nije do kraja razvijena iako Google ubrzano radi na tome kako bi se i lokalizacija aplikacija mogla što lakše implementirati.

Iako ima svoje nedostatke, ova platforma ima puno više prednosti. Razvijena je kompleksna aplikacija za Android i iOS operacijske sustave u relativno kratkom vremenu. Većina značajki u aplikaciju se jako lagano implementira. Kao što je dohvaćanje lokacije, pristup galeriji fotografija, pristup kameri i slično. Sve je to otprije implementirano od strane zajednice koja razvija ovu platformu kao i sami Google. Nedavno je Google izbacio dodatak platformi za animiranje prijelaza između zaslona koji je korišten i u ovoj aplikaciji i koji je iznenađujuće lako implementirati.

Zbog svega toga ova platforma vrlo vjerojatno ima blistavu budućnost u razvoju mobilnih i ostalih aplikacija. Pogotovo zbog toga što jedna velika tvrtka kao što je Google razvija i neprekidno ulaže u nju. Vjerujem kako će se platforma u budućnosti postati još stabilnija, podržavati mogućnost razvoja za još više operacijskih sustava i da će postati još popularnija među zajednicom razvojnih programera.

SAŽETAK

Cilj ovog završnog rada je razvoj hibridne mobilne aplikacije za Android i iOS operacijske sustave pomoću Flutter platforme koja će omogućiti lovcima pohranu, sinkronizaciju i dijeljenje lovačkih trofeja. Lovci se mogu registrirati i prijaviti u aplikaciju, a svi njihovi podaci i trofeji će biti pohranjeni u bazu podataka. Trofeji se unutar aplikacije mogu podijeliti samo jednim klikom, a korisnici koji žele pogledati informacije o trofeju ne moraju imati instaliranu aplikaciju na svom pametnom telefonu. Aplikacija ima potencijala za postizanjem komercijalnog uspjeha na globalnom tržištu. Iz tog razloga, bilo je potrebno razviti aplikaciju koja će podržavati operacijske sustave koji se danas najviše koriste na tržištu pametnih telefona. Zbog toga je za razvoj odabrana platforma Flutter. Tako su pomoću Flutter platforme izrađene Android i iOS aplikacije čiji proces izrade je opisan u ovom radu.

Ključne riječi: Android aplikacija, iOS aplikacija, Flutter, lovački trofeji

ABSTRACT

Hybrid mobile application for storing hunting trophies

The main goal of this project was the development of a hybrid mobile application for Android and iOS operating systems using the Flutter framework, which will allow hunters to store, synchronize, and share their hunting trophies. The hunters can log in and register, and their personal information and trophies will be saved in a database. The trophies can be shared with just one click from the application, and users who want to see information about specific trophy don't have to have an app installed on their smartphone. The application has the potential to reach global commercial success. Because of that, the app was needed to be developed for operating systems, which are the most used in today's smartphone market. A Flutter framework has been chosen to achieve this. Both applications for Android and iOS have been developed with Flutter, for which the process of development is described at this work.

Keywords: Android application, iOS application, Flutter, hunting trophies

ŽIVOTOPIS

Antonio Valentić rođen je 20.11.1997. godine u Bjelovaru, Hrvatska. Nakon završene osnovne škole, 2012. godine upisuje Tehničku školu u Bjelovaru, smjer tehničar za računalstvo. Tijekom druge godine srednje škole se upoznaje s programiranjem i od onda se bavi time. Godine 2016. upisuje prvu godinu sveučilišnog preddiplomskog studija računarstva na Fakultetu Elektrotehnike, Računarstva i Informacijskih Tehnologija Osijek. U razdoblju od 2012. godine do 2017. radi kao slobodni profesionalac nakon čega prelazi raditi za novu Bjelovarsku tvrtku kao Android razvojni programer. Tijekom rada u navedenoj tvrtki, započinje s edukacijom za Flutter platformu. Nakon završene edukacije bavi se izradom hibridnih aplikacija za Android i iOS operacijske sustave koristeći Flutter platformu. U studenom 2019. godine, dobiva posao Flutter razvojnog programera u Belgijskoj tvrtki u kojoj radi do trenutka pisanja ovog rada. Od ostalih znanja posjeduje osnove razvoja web aplikacija, određeno znanje engleskog jezika i vozačku kategoriju B kategorije.

Antonio Valentić

LITERATURA

[1] A. Turner, How many smartphones are in the world?: Number of mobile phone & smartphone users [online], BankMyCell, 2020., dostupno na: https://www.bankmycell.com/blog/how-many-phones-are-in-the-world [07. travnja 2020.]

[2] Nepoznat autor, Mobile Operating System Market Share Worldwide [online], StatCounter, 2020., dostupno na: https://gs.statcounter.com/os-market-share/mobile/worldwide [07.04.2020.]

[3] B. Kokot, Razvoj Android aplikacije, Fakultet ekonomije i turizma „Dr. Mijo Mirković“, Pula, 2015.

[4] A. Rubin, Where's my Gphone? [online], Official Google Blog, 2007., dostupno na: https://googleblog.blogspot.com/2007/11/wheres-my-gphone.html [07. travnja 2020.]

[5] D. Morrill, Announcing the Android 1.0 SDK, release 1 [online], Android Developers Blog, 2008., dostupno na: https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html [07. travnja 2020.]

[6] R. Gao, Android and its first purchasable product, the T-Mobile G1, celebrate their 8th birthdays today [online], Android Police, 2016., dostupno na: https://www.androidpolice.com/2016/09/23/android-first-purchasable-product-t-mobile-g1-celebrate-8th-birthdays-today/, [07. travnja 2020.]

[7] M. Marinović, Razvoj Android aplikacije „Kviz“, Pomorski fakultet u Rijeci, Rijeka, 2014.

[8] D. Radman, Usporedba vodećih operacijskih sustava za pametne telefone, Ekonomski fakultet Split, Split, 2018.

[9] B. Ševo, Razvoj iOS aplikacija, Prirodoslovno – matematički fakultet u Zagrebu, Zagreb, 2014.

[10] I. Dujlović, Z. Đurić, Razvoj hibridnih mobilnih aplikacija pomoću PhoneGap platforme, Elektrotehnički fakultet Banja Luka, Banja Luka, 2013.

[11] D. Buha, Mobilna aplikacija za šah na React-Native platformi, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Osijek, 2018.

[12] B. Evkovski, React Native: What it is and how it works: How does it work? [online], Medium, 2017., dostupno na: https://medium.com/we-talk-it/react-native-what-it-is-and-how-it-works-e2182d008f5e [08. travanja 2020.]

[13] R. Amadeo, Google's Dart language on Android aims for Java-free, 120 FPS apps [online], Ars Technica, 2015., dostupno na: https://arstechnica.com/gadgets/2015/05/googles-dart-language-on-android-aims-for-java-free-120-fps-apps/ [08. travanja 2020.]

[14] C. Bracken, v0.0.6: Rev alpha branch version to 0.0.6, flutter 0.0.26 (#10010) [online], Github, 2017., dostupno na: https://github.com/flutter/flutter/releases/tag/v0.0.6 [09. travanja 2020.]
[15] K. Titong, Speed Up Native Development As Google Flutter Comes Out Of Beta [online], Appetiser Apps, 2018., dostupno na: https://appetiser.com.au/blog/speed-up-native-development-as-google-flutter-comes-out-of-beta/ [09. travanja 2020.]
[16] T. Sneath, Technical overview – Flutter [online], Flutter, 2020., dostupno na: https://flutter.dev/docs/resources/technical-overview [09. travanja 2020.]
[17] Nepoznat autor, Presentation: „Opening Keynote: Dart, a new programming language for structured web programming“ [online], GOTO Conference, 2011., dostupno na: http://gotocon.com/aarhus-2011/presentation/Opening%20Keynote:%20Dart,%20a%20new%20programming%20language%20for%20structured%20web%20programming [10. travanja 2020.]
[18] K. Walrath, What is Dart?: Inside a new language for building structured web apps [online], O'Reilly Radar, 2012., dostupno na: http://radar.oreilly.com/2012/03/what-is-dart.html [10. travanja 2020.]
[19] L. Bak, Dart 1.0: A stable SDK for structured web apps [online], Dart News & Updates, 2013., dostupno na: https://news.dartlang.org/, [10. travanja 2020.]
[20] A. T. Sandholm, Annoucing Dart 2: Optimized for Client-Side Development: Dart's Core Tenets [online], Medium, 2018., dostupno na: https://medium.com/dartlang/announcing-dart-2-80ba01f43b6 [10. travanja 2020.]
[21] V. Katić, Android klijentska aplikacija za centralizirano naručivanje hrane u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Osijek, 2016.
[22] Nepoznat autor, Documentation | Firebase [online], Firebase, 2020., dostupno na: https://firebase.google.com/docs [10. travanja 2020.]