Mobilna Android aplikacija za pomoć pri biranju usluga manikiranja

Miličević, Melani

Undergraduate thesis / Završni rad

2023

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:200:617018

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij

MOBILNA ANDROID APLIKACIJA ZA POMOĆ PRI BIRANJU USLUGA MANIKIRANJA

Završni rad

Melani Miličević

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

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

Osijek, 13.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Melani Miličević
Studij, smjer:	Računalno inženjerstvo
Mat. br. Pristupnika, godina upisa:	R 4404, 12.10.2021.
OIB Pristupnika:	73263904022
Mentor:	prof. dr. sc. Goran Martinović
Sumentor:	mag. ing. comp. Dino Kurtagić
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna Android aplikacija za pomoć pri biranju usluga manikiranja
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rad:	U teorijskom dijelu završnog rada treba proučiti i opisati postupak, načela, izazove i postojeća programska rješenja za preporučivanje usluga manikiranja. Na temelju osobnog profila korisnice, oblika, dužine i boje noktiju, doba godine, kao i prijašnjih izbora noktiju i ponašanja korisnika, treba definirati funkcionalne i nefunkcionalne zahtjeve, model, arhitekturu i dizajn mobilne aplikacije s bazom podataka, kao i odgovarajući postupak biranja kombinacija noktiju višekriterijskim odlučivanjem. Mobilna aplikacija za preporučivanje
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	13.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

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

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 26.09.2023.

Ime i prezime studenta:	Melani Miličević
Studij:	Računalno inženjerstvo
Mat. br. studenta, godina upisa:	R 4404, 12.10.2021.
Turnitin podudaranje [%]:	12

Ovom izjavom izjavljujem da je rad pod nazivom**: Mobilna Android aplikacija za pomoć pri biranju usluga manikiranja**

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

i sumentora mag. ing. comp. Dino Kurtagić

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

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. POMOĆ PRI BIRANJU USLUGA MANIKIRANJA I PRIKAZ STANJA U PODRUČJU 3

2.1. Manikura 3

2.1.1. Moguće posljedice manikiranja 3

2.2. Tehnike izrade noktiju 3

2.3. Oblik i dužina 3

2.4. Tehnike ekspresije 4

2.5. Postojeće aplikacije u području i prikaz stanja u području 4

2.5.1. YouCam Nails – Manicure Salon 5

2.5.2. Lak za nokte 5

2.5.3. Styles4Nailz – Nail Designs 6

2.6. Idejno rješenje vlastite mobilne aplikacije za pomoć pri biranju usluga manikiranja 6

3. MODEL I ARHITEKTURA MOBILNE APLIKACIJE 7

3.1. Funkcionalni zahtjevi za mobilnu aplikaciju 7

3.1.1. Početni zaslon aplikacije 7

3.1.2. Prijava korisnika 7

3.1.3. Registriranje korisnika 7

3.1.4. Izrada profila korisnika 7

3.1.5. Postupak višekriterijskog odlučivanja 8

3.1.6. Postupak predlaganja novih dizajna 8

3.1.7. Skočni izbornik 9

3.1.8. Uređivanje galerije 9

3.2. Nefunkcionalni zahtjevi za mobilnu aplikaciju 9

3.3. Građa mobilne aplikacije 9

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE 11

4.1. Korišteni alati i tehnologije 11

4.1.1. Operacijski sustav *Android* 11

4.1.2. Figma 11

4.1.3. Android Studio 12

4.1.4. Razvojna platforma *Flutter* 12

4.1.5. Dart 12
4.1.6. Firebase 12
4.2. Prikaz programskog rješenja po komponentama **13**
4.2.1. Pokretanje aplikacije 13
4.2.2. Registracija i prijava korisnika 14
4.2.3. Postavljanje i prikaz korisničkog profila 16
4.2.4. Generiranje novih dizajna 19
4.2.5. Skočni izbornik 22
4.2.6. Uređivanje galerije 24
5. NAČINI KORIŠTENJA I ISPITIVANJA MOBILNE APLIKACIJE S ANALIZOM REZULTATA **25**
5.1. Korištenje mobilne aplikacije **25**
5.2. Prikaz rada mobilne aplikacije **25**
5.2.1. Sučelje početnog zaslona 25
5.2.2. Prikaz zaslona za registraciju i prijavu 26
5.2.3. Profil korisnika i generiranje novih dizajna 27
5.2.4. Prikaz skočnog izbornika 28
5.2.5. Prikaz zaslona za uređivanje profila 29
5.3. Ispitivanje rada mobilne aplikacije **30**
5.3.1. Prvi ispitni slučaj 30
5.3.2. Drugi ispitni slučaj 31
6. ZAKLJUČAK **32**
LITERATURA ..... Error! Bookmark not defined.
POPIS SLIKA **35**
POPIS TABLICA **37**
SAŽETAK **38**
ABSTRACT **39**
PRILOZI **40**

1. UVOD

Manikura noktiju uključuje turpijanje i oblikovanje rubova noktiju, potiskivanje i rezanje neživog tkiva zanoktica i noktiju, tretmane raznim tekućinama, masaže ruku i nanošenje laka za nokte. Redovna manikura sprječava nastanak infekcija (npr. gljivice), potiče zdravu cirkulaciju i može unaprijediti i mentalno zdravlje. Nastanak manikure ne može se pripisati jednoj kulturi. Prisutna je bila u vremenu starog Egipta, Indije i Babilona datirajući i do 5. stoljeća prije Krista. Kroz vrijeme, različiti dizajni manikure su bili popularni i danas je moguće vidjeti od najobičnijih do iznimno složenih dizajna. Najčešće, aplikacije nude mogućnost uređivanja noktiju ili samo pregled zbirke slika i videozapisa.

Izrađena mobilna aplikacija prati navike korisnika i nudi nove dizajne potaknute dosadašnjim izborima. Kod izrade manikure moguće je birati tehniku, dužinu i oblik. Za uspješnost rada aplikacije potrebna je zbirka različitih dizajna i određena tehnika i kriteriji odlučivanja. Korisnik registracijom, odnosno prijavom u aplikaciju ima mogućnost spremanja dizajna, koji mu se sviđaju, u galeriju. Na osnovnu spremljenih dizajna i preferencija postavljenih tijekom registracije, korisniku se nude novi dizajni manikure. Za izradu mobilne aplikacije korištena je razvojna platforma Flutter, programski jezik Dart i programsko okruženje Andorid Studio. Za rad s bazama podataka i autentifikaciju korisnika korištene su usluge Google Firestore-a. Kroz rad opisat će se nekoliko kriterija bitnih za manikuru, programi i jezici potrebni za razvoj aplikacije, objasniti građa aplikacije i prikazati način rada aplikacije.

Drugo poglavlje opisuje čimbenike manikiranja kao što su oblik, dužina, tehnike i dizajn te neka programska rješenja sa sličnim funkcionalnostima. U trećem poglavlju prikazuje se model i arhitektura mobilne aplikacije s opisima glavnih funkcionalnosti i parametrima koji se koriste za rad aplikacije. U četvrtom poglavlju navedene su korištene tehnologije i jezici za razvoj mobilne aplikacije. U petom poglavlju, prikazan je rad aplikacije i njeno testiranje kroz nekoliko slučajeva.

1.1. Zadatak završnog rada

U teorijskom dijelu završnog rada treba proučiti i opisati postupak, načela, izazove i postojeća programska rješenja za preporučivanje usluga manikiranja. Na temelju osobnog profila korisnice, oblika, dužine i boje noktiju, doba godine, kao i prijašnjih izbora noktiju i ponašanja korisnika, treba definirati funkcionalne i nefunkcionalne zahtjeve, model, arhitekturu i dizajn mobilne aplikacije s bazom podataka, kao i odgovarajući postupak biranja kombinacija noktiju višekriterijskim odlučivanjem. Mobilna aplikacija za preporučivanje usluga manikiranja treba

omogućiti registriranje i prijavu, unos i pohranu profila korisnika i osnovnih oblika, dužina, boja i izvedbi noktiju, stvaranje i prikaz mogućih usluga manikiranja prema definiranim pravilima uz odgovarajući postupak višekriterijskog odlučivanja i njihov pojednostavljeni prikaz na obliku prstiju korisnika. Mobilnu Android aplikaciju potrebno je programski ostvariti koristeći aktualne programske jezike, tehnologije i razvojne okvire, kao što su Kotlin i Flutter te je ispitati za odgovarajuće ulazne podatke i slučajeve korištenja

2. POMOĆ PRI BIRANJU USLUGA MANIKIRANJA I PRIKAZ STANJA U PODRUČJU

2.1. Manikura

Manikura se smatra tretmanom ruku i noktiju. Ključni koraci u manikuri su rezanje noktiju, oblikovanje turpijom, uklanjanje zanoktica i lakiranje. Lakiranje noktiju, takozvani *nail art,* jedan je od najpoznatijih načina samoizražavanja i postoji još od staroegipatskog razdoblja gdje su žene nokte bojale kanom, a plemići su koristili crnu boju. U moderno vrijeme, umjetnost noktiju postala je jedna od najvećih industrija koja ima kreativan način bojanja, ukrašavanja i poboljšavanja noktiju [1].

2.1.1. Moguće posljedice manikiranja

Mlade žene koje odlaze na manikuru su samouvjerenije i atraktivnije u društvenom okruženju. Glavna bit manikure je održavanja te je priprema i proces manikiranja strog. Bez obzira na to, manikura ima i svoje negativne strane. Prema istraživanju [2], neki od materijala korišteni u procesu sadržavaju toksične tvari kao što su ftalati i metilni alkohol. Takve tvari mogu biti štetne ljudskom tijelu ako im se često izlaže. Iako je većinom slučaj za kozmetičarke koje rade manikuru, moguća je pojava kontaktnog alergijskog dermatitisa zbog akrilata koji su u sastavu nekih materijala. U članku [3], opisane su neke značajke, dijagnostičke procedure, tretmani i preventivne metode.

2.2. Tehnike izrade noktiju

Pri odabiru tehnike izrade manikure u obzir uzimaju se različiti čimbenici kao što su prigoda, cijena i stil života. Neke od najpoznatijih tehnika manikure prema [2] su:

1. Osnovna
2. *Shellac*
3. Gel
4. Akril

2.3. Oblik i dužina

Kod manikiranja odabir dužine i oblika je iznimno važan i svaki oblik ne pristaje uz svaku dužinu. Dužina nokta može biti:

- Kratka

- Srednja
- Duga
- Ekstra duga

Nekoliko je različitih oblika nokta prema [3]:

- Okrugli
- Ovalni
- Kockasti
- Badem
- Kovčeg/Balerina

2.4. Tehnike ekspresije

Prema [6], tehnike ekspresije su:

- ručno oslikavanje (engl. *Hand Painting*) – izravno crtanje na obojanim noktima,
- drago kamenje (engl. *Rhine Stone*) – odnosi se na umjetne dijamante, često u kombinaciji sa ravnim tehnikama kao što je ručno oslikavanje i zračni kist,
- 3D umjetnost (engl. *3D Art*) – stvaranje 3D umjetnosti s akril prahovima i tekućinama,
- preslikavanje (engl. *Decal*) – već oslikane naljepnice koje se koriste na obojenim noktima,
- mramor (engl. *Marbling*) – crtanje stvarnih oblika s nekoliko boja sa zasebnim kistovima kako bi imali apstraktni, „mramorni“ izgled,
- zračni kist (engl. *Airbrush*) – prskanje boje pomoću kompresora i zračnog pištolja kako bi se stvorile ilustracije i detaljni prikazi.

Svaka tehnika može se koristiti zasebno ili u kombinaciji s drugim.

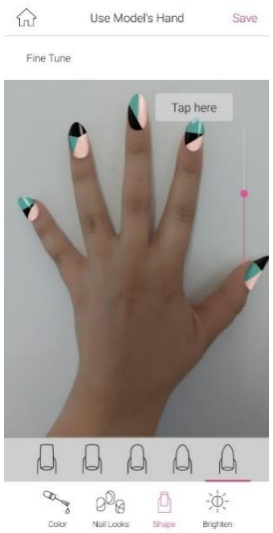
2.5. Postojeće aplikacije u području i prikaz stanja u području

Većina aplikacija u području manikure nudi mogućnost uređivanja noktiju sa nekim već poznatim dizajnima. Mogućnosti aplikacija najčešće su pravljenje i spremanje vlastitih dizajna te video uputstva kako postići dizajn. Nekolicina je razvijenih aplikacija u stilu mobilnih igara koje korisniku daju „iskustvo“ radnika u salonu za manikuru.

2.5.1. YouCam Nails – Manicure Salon

YouCam Nails aplikacija je razvijena od strane *Perfect Corp.*, tvrtke koja koristi umjetnu inteligenciju kako bi ponudila rješenja za razne potrebe industrije ljepote [6]. Aplikacija nudi preko 65 boja laka, preko 25 različitih uzoraka i razne naljepnice za potrebe pravljenja dizajna. Pravljenja vlastitog dizajna vidi se na slici 2.1. Moguće je i pomoću kamere isprobati dizajne na svojim noktima kao na slici 2.2. Sadrži i video upute kako realizirati razne ideje.

Slika 2.2 Pravljenje vlastitog dizajna



Slika 2.1 Isprobavanje dizajna pomoću kamere

2.5.2. Lak za nokte

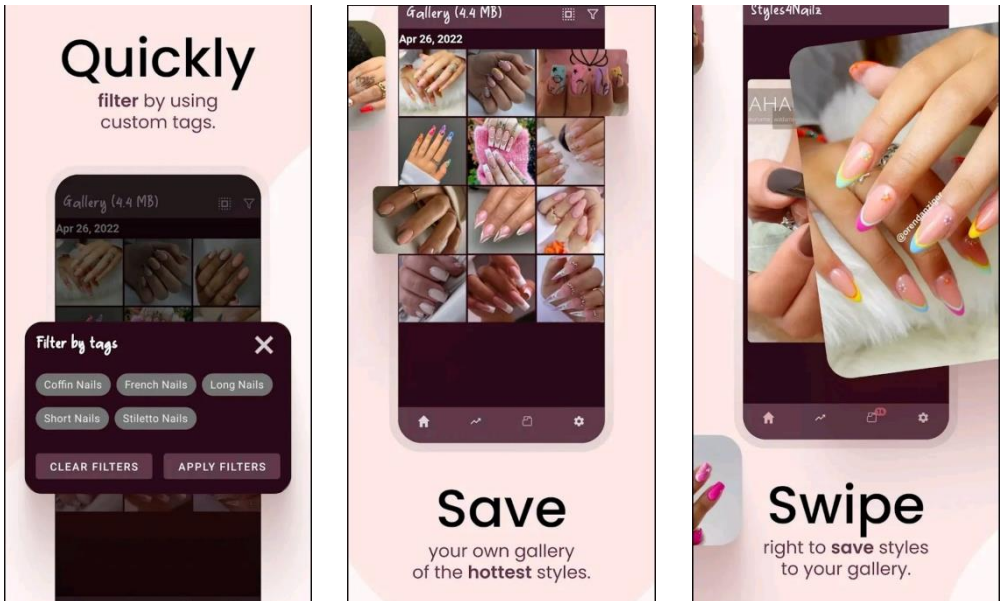
Aplikacija Lak za nokte (engl. *Nail Art Design Ideas*) sadrži veliku kolekciju dizajna(videozapise i slike). Dizajni se mogu preuzeti i omogućen je filtrirani pregled dizajna po boji. Slika 2.3 prikazuje filtar za odabir boja, a na slici 2.4 vidi se kolekcija dizajna.

Slika 2.3 Prikaz filtera aplikacije

Slika 2.4 Kolekcija dizajna aplikacije

2.5.3. Styles4Nailz – Nail Designs

Styles4Nailz aplikacija pomaže korisniku da istraži raznolike dizajne pomoću poteza lijevo-desno. Ako se korisniku sviđa dizajn potrebno je povući prstom u desno i dizajn se sprema u galeriju, su suprotnom potrebno je povući lijevo. Aplikacija ima i mogućnost filtriranja dizajna pomoću prilagođenih oznaka. Sučelje aplikacije nalazi se na slici 2.5.



Slika 2.5 Sučelje aplikacije Styles4Nailz

2.6. Idejno rješenje vlastite mobilne aplikacije za pomoć pri biranju usluga manikiranja

Navedene postojeće aplikacije i njihove komponente inspirirale su aplikaciju za pomoć pri biranju usluga manikiranja. Prva opisana aplikacija daje korisniku personalizaciju dužine i oblika nokta. U drugoj aplikaciji korisnik pretražuje bazu slika s mogućnosti filtriranja po bojama. Treća aplikacija također omogućuje filtrirano pretraživanje pomoću ključnih riječi koje nisu ograničene samo na boju. Osim pretraživanja, korisnik ima mogućnost spremanja fotografija u galeriju. Slično prethodno opisanim aplikacijama aplikacija za pomoć pri biranju usluga manikiranja mora moći izgraditi profil korisnika s nekim početnim filterima, tj. preferencijama pomoću kojih će se korisniku nuditi različiti dizajni i omogućiti njihovo spremanje u galeriju. Kako bi korisnik mogao postaviti profil potrebna je registracija, tj. prijava korisnika. Za razliku od spomenutih postojećih aplikacija, ova aplikacija predstavljala bi dizajne jedan po jedan i umjesto da korisnik ručno pretražuje po filterima, svako dodavanje u galeriju proširuje preferencije korisnika pa tako i ponudu novih dizajna bez da je korisnik toga svjestan. Više o modelu i načinu rada aplikacije detaljnije je opisano u trećem poglavlju.

3. MODEL I ARHITEKTURA MOBILNE APLIKACIJE

3.1. Funkcionalni zahtjevi za mobilnu aplikaciju

Mobilna aplikacija za pomoć pri biranju usluga manikiranja na početku ima opciju registracije i prijave korisnika (ako već ima korisnički račun). Pri registraciji, korisnik unosi e-mail adresu, željenu lozinku i osobne podatke. Zatim bira svoje preferencije za dužinu, oblik nokta i tehniku manikure. Na profilu korisnika se nalazi galerija spremljenih dizajna i navigacijska traka koja ima gumb za prebacivanje sa profila na zaslon za predlaganje novog dizajna.

3.1.1. Početni zaslon aplikacije

Kako će izgledati zaslon kada se pokrene aplikacija ovisi o tome je li korisnik prijavljen ili ne. Ako korisnik nije prijavljen prikazan će biti početni zaslon s dva gumba, jedan za prijavu korisnika, drugi za registraciju. U slučaju da je korisnik prijavljen otvoren treba biti profil korisnika.

3.1.2. Prijava korisnika

Pritiskom gumba za prijavu na početnom zaslonu, prikazuju se dva polja i od korisnika se traži unos e-mail adrese i lozinke. Nakon autentifikacije podataka, otvara se profil korisnika. Aplikacija prati stanje prijave i odjave korisnika tako da se korisnik ne mora prijavljivati svakim ponovnim otvaranjem aplikacije osim ako se prija izlaska nije odjavio.

3.1.3. Registriranje korisnika

Ako korisnik odabere registraciju, korisniku se otvara zaslon sa tri polja i jednim gumbom. U polja je potrebno upisati osobno ime, e-mail adresu i lozinku. Nakon toga, pritiskom na gumb podaci se spremaju i otvara se zaslon za odabir preferencija.

3.1.4. Izrada profila korisnika

Podaci traženi tijekom registracije nisu od velike važnosti kao oni prikupljeni tijekom izrade profila. U izradi profila od korisnika se traži da odabere preferiranu dužinu, oblik nokta i tehniku manikure. Ti podaci pomažu pri generiranju novih dizajna za korisnika.

Tablica 3.1 Moguće vrijednosti dužine, oblika i tehnike nokta

DUŽINA NOKTIJU	OBLIK NOKTIJU	TEHNIKA MANIKURE
Kratki Srednji Dugi Ekstra Dugi	Okrugli Ovalni Kockasti Badem Kovčeg/Balerina	Obična *Shellac* Gel Akril

3.1.5. Postupak višekriterijskog odlučivanja

Bitan dio procesa višekriterijskog odlučivanja je strukturiranje problema odluke u formalni i upravljivi format [8]. Kod odabira preferencija nokta problem se pojavljuje kod odabira pravilnog oblika noktiju. Osim duljine nokta, utjecaj u odluku oblika ima i prirodni oblik nokta i prstiju klijenta. Prema [9], napravljen je vodič (Tablica 3.2) za svaki oblik nokta.

Tablica 3.2 Preporučeni oblici nokta prema obliku ruke i prstiju

Oblik nokta	**Prirodni oblik nokta i prstiju**
Ovalni i okrugli	Male ruke, mali prsti
Kockasti	Dugački prsti, uska ploča nokta
Badem	Kratki, široki prsti
Kovčeg/Balerina	Tanki prsti, čvrsta ploča nokta

Što se tiče duljine nokta, u tablici 3.3 se nalazi najbolje kombinacije oblika za svaku duljinu nokta [10].

Tablica 3.3 Preporučeni oblici nokta prema duljini nokta

Duljina nokta	**Oblik nokta**
Kratki	Kockasti, ovalni
Srednji	Bilo koji
Dugi	Bilo koji
Ekstra dugi	Kovčeg/balerina

Korisnik može odabrati preferencije koje se ne podudaraju, npr. kratki, kovčeg nokti, za koje nema primjer *nail arta*. U tom slučaju gleda se treći kriterij odlučivanja, tehnika. Tehnika je moguća u kombinaciji sa bilo kojim tipom duljine i oblika pa će tako biti predložene kombinacije oblik-tehnika i duljina-tehnika. Na primjer, odabrane preferencije su kratki, kovčeg, gel nokti, prijedlog *nail arta* biti će kratki, gel nokti i/ili kovčeg, gel nokti.

3.1.6. Postupak predlaganja novih dizajna

Na zaslonu za predlaganje dizajna nalazi se fotografija s dizajnom i dva gumba. Jedan gumb je za preskakanje, tj. traženje novih dizajna, a ako se korisniku sviđa predloženi dizajn pritiskom na drugi gumb, aplikacija ga sprema u galeriju. Osim što sprema fotografije dizajne, u korisničku bazu spremaju se i, kao preferencije, ključne riječi koje svaka fotografija sadrži. Ključne riječi dodatno opisuju dizajn i tako svakim spremanjem proširuje se ponuda novih dizajna koji bi se

mogli sviđati korisniku. Za predlaganje novih dizajna potrebna je i baza s fotografijama različitih *nail art* dizajna s ključnim riječima koje ih opisuju. Za potrebe ove aplikacije korištene su fotografije preuzete s *Instagram* profila „DORIS NAILS“ (@dorisnailartist) [7]. Sve fotografije spremljene su u *Firebase Storage*.

3.1.7. Skočni izbornik

Na zaslonu profila u gornjem desnom kutu nalaze se tri točkice, to je skočni izbornik. U njemu su tri opcije: „*Edit profile“* (Uredi profil), „*Clear gallery“* (Očisti galeriju), „*Sign out“* (Odjava). ,,Uredi profil" otvara zaslon gdje korisnik može promijeniti neke postavke profila kao što su slika profila, ime i lozinka. U slučaju promjene lozinke od korisnika se prvo traži da potvrdi trenutnu lozinku. Tipka *„Sign out“* odjavljuje korisnika i otvara početni zaslon gdje se ponovno može prijaviti ili registrirati novi račun. *„Clear gallery“* briše sve spremljene dizajne i vraća preferencije na početne, odabrane prilikom registracije.

3.1.8. Uređivanje galerije

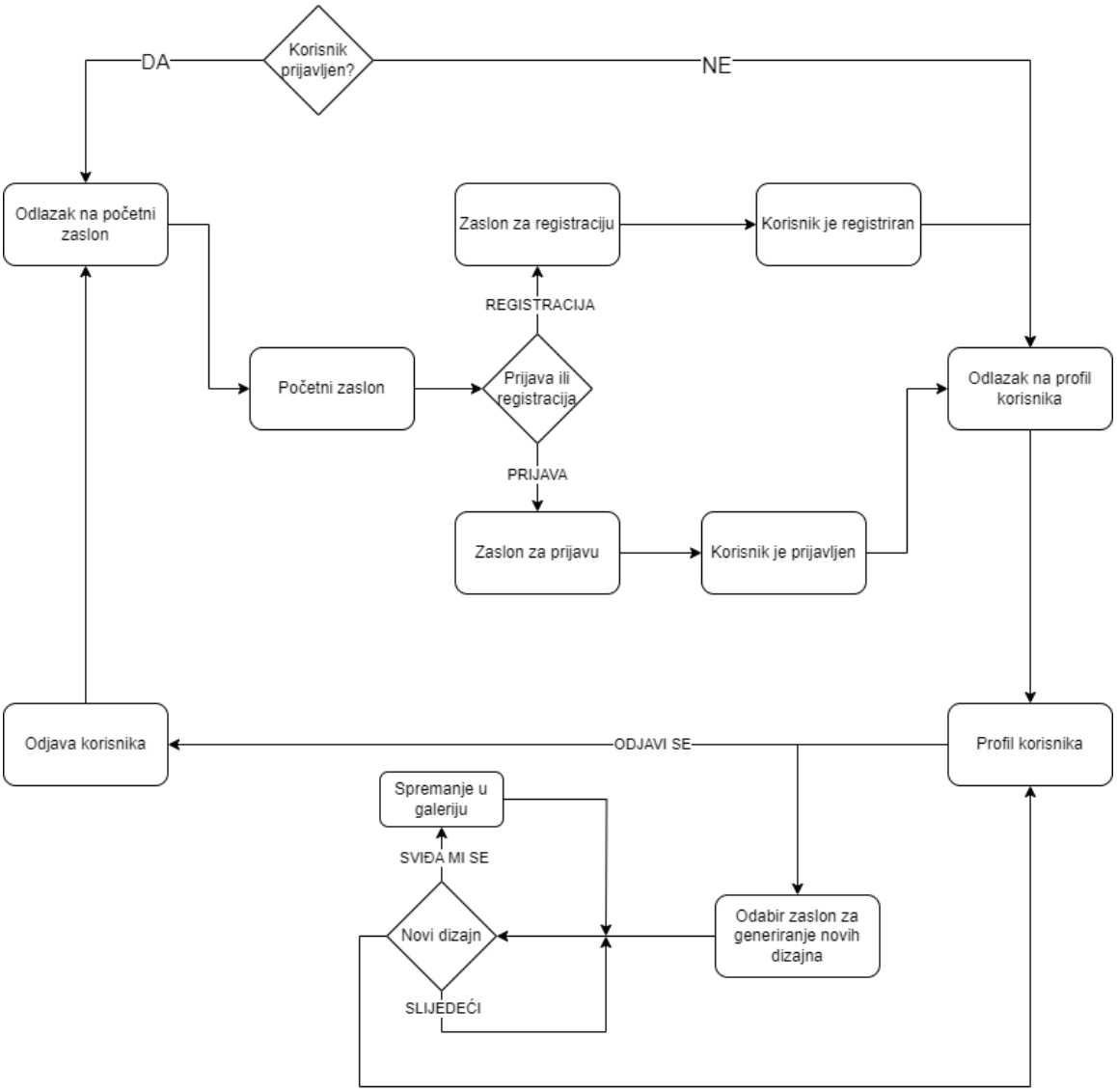
Korisnik ima mogućnost odabirom na fotografiju otvoriti je u većem prikazu i da izbriše ju iz galerije.

3.2. Nefunkcionalni zahtjevi za mobilnu aplikaciju

Nefunkcionalni zahtjevi za mobilnu aplikaciju skup su specifikacija koji opisuju mogućnosti i ograničenje i pokušaje poboljšavanje funkcionalnosti rada sustava. To su zahtjevi kao što su brzina, sigurnost, pouzdanost, integritet podataka i dr. [8].

3.3. Građa mobilne aplikacije

Dio korisničke strane aplikacije su početni zaslon, zaslon za prijavu, zaslon za registraciju, zaslon profila korisnika, zaslon za generiranje novih dizajna. Postoje dvije baze podataka: za pohranu korisničkih podataka i za pohranu mogućih opcija dizajna. Za rad sa bazama korišten je Firebase. Dijagramom na slici 3.1 prikazana je arhitektura i tok aplikacije. U funkcionalnim zahtjevima mobilne aplikacije detaljnije je opisan tijek rada aplikacije.



Slika 3.1 Dijagram toka aplikacije

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE

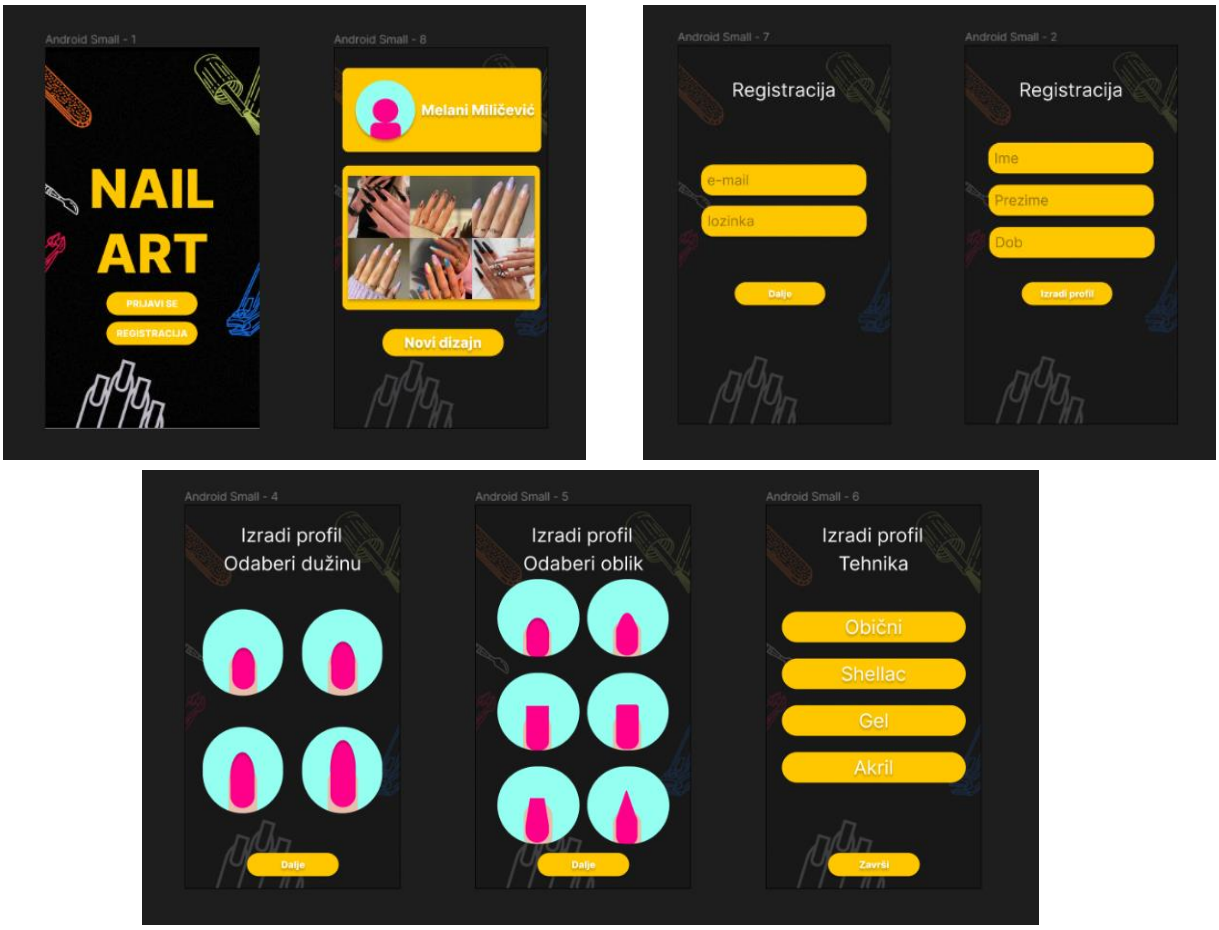
4.1. Korišteni alati i tehnologije

4.1.1. Operacijski sustav *Android*

Android [9] je operacijski sustav razvijen od strane *Googlea*. Primarno je korišten na pametnim telefonima i tabletima. *Android* sustav je baziran na *Linux* jezgri i podržava veliki broj aplikacija. Otvoren je sustav, što znači da je besplatan i može ga bilo tko koristiti.

4.1.2. Figma

Figma je aplikacija za dizajn sučelja koja radi u pregledniku. Daje sve alate potrebne za dizajn, uključujući vektorske alate koji su sposobni za izradu potpunih ilustracija, kao i mogućnosti izrade prototipa i generiranje programskog koda [10].



Slika 4.1 Dizajn sučelja aplikacija pomoću aplikacije *Figma*

4.1.3. Android Studio

Android Studio [11] službeno je integrirano razvojno okruženje (engl. *Integrated Development Environment*) za razvoj *Android* aplikacija. *Android studio* nudi niz značajki kao što su: fleksibilan sustav izrade temeljen na *Gradleu*, brz emulator, jedinstveno sučelje u kojem je moguće razvijati sve *Android* uređaje, *GitHub* integracija, alati i okviri za testiranje i dr.

4.1.4. Razvojna platforma *Flutter*

Flutter [12] je platforma za razvoj aplikacija kreirana od strane *Googlea*. Dozvoljava programerima da razvijaju Internet, računalne i među-platformske aplikacije koje rade na *Andoridu* i iOS uređajima. Koristi programski jezik *Dart*, što čini programiranje bržim i lakšim od nekih tradicionalnih metoda.

4.1.5. Dart

Dart [13] je objektno-orijentirani programski jezik. Koristi se za razvijanje *web*, mobilnih i *desktop* aplikacija. Može se koristiti za kreiranje samostalnih programa ili za razvoj aplikacija koristeći razvojne okoline kao što je *Flutter*.

4.1.6. Firebase

Google Firebase platforma je koja pruža razne alate i usluge koji pomažu u razvoju aplikacija. Nudi gotovu pozadinu za aplikacije temeljenu na oblaku. Radi kao posrednik između aplikacije i usluga temeljenih na oblaku (engl. *cloud*) koje su potrebne za učinkovito funkcioniranje, upravljajući pohranom podataka, autentifikacijom korisnika, analitikom i dr. [14].

4.2. Prikaz programskog rješenja po komponentama

4.2.1. Pokretanje aplikacije

Na slici 4.2 može se vidjeti konstruktor *AuthProvider* u kojem se nalazi *updateShouldNotfiy* koji prati stanje autentifikacije i obavještava *AuthProvider* ako dođe do promjene (prijava/odjava).

```
class AuthProvider extends InheritedWidget {
  final AuthService auth;

  AuthProvider({Key? key, required Widget child, required this.auth}) : super(key: key, child: child);

  @override
  bool updateShouldNotify(InheritedWidget oldWidget) {
    return true;
  }

  static AuthProvider of(BuildContext context) =>
      context.dependOnInheritedWidgetOfExactType<AuthProvider>()!;

}
```

Slika 4.2 Konstuktor *AuthProvider*

U *HomeController* (Slika 4.3) poziva se *AuthProvider* i ako postoji prijavljen korisnik poziva se zaslon koji prikazuje profil korisnika (*ProfilePage*), u suprotnom početni zaslon (*MyHomePage*).

```
class HomeController extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    final AuthService auth = AuthProvider.of(context).auth;
    return StreamBuilder<String?>(
        stream: auth.onAuthStateChanged,
        builder: (context, AsyncSnapshot<String?> snapshot) {
          if (snapshot.connectionState == ConnectionState.active) {
            final bool signedIn = snapshot.hasData;
            if (signedIn) {
              final uid=snapshot.data!;
              return ProfilePage(uid: uid);
            } else {
              return MyHomePage();
            }
          }
          return CircularProgressIndicator();
        });  // StreamBuilder
  }
}
```

Slika 4.3 *HomeController widget*

4.2.2. Registracija i prijava korisnika

Nakon što korisnik unese tražene informacije (puno ime korisnika, e-mail i lozinku) pritiskom na tipku *„Next“* na zaslonu poziva se metoda *signUp* (Slika 4.4) u kojoj se poziva *signUpWithEmailAndPassword*. (Slika 4.5). Metoda *signUpWithEmailAndPassword* pomoću *Firebase Authentication* stvara novog korisnika, a metoda *saveUserInfo* (Slika 4.6) sprema ime i e-mail u *Realtime database*. Ako je autentifikacija uspješna, otvara se zaslon *profileSetUp*.

```
void signUp()async{
  final auth = AuthProvider.of(context).auth;
  try{
    _userCredential = await auth.signUpWithEmailAndPassword(
        _emailController.text,
        _passwordController.text,
        _fullNameController.text);

    Navigator.of(context).pushReplacement(MaterialPageRoute(
      builder: (context) => ProfileSetUp(userCredential: _userCredential!),
    ),  // MaterialPageRoute
    );
  }catch(e){
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
        ├── content: Text('Something went wrong, try again!')));  // SnackBar
  }
}
```

Slika 4.4 Metoda *signUp*

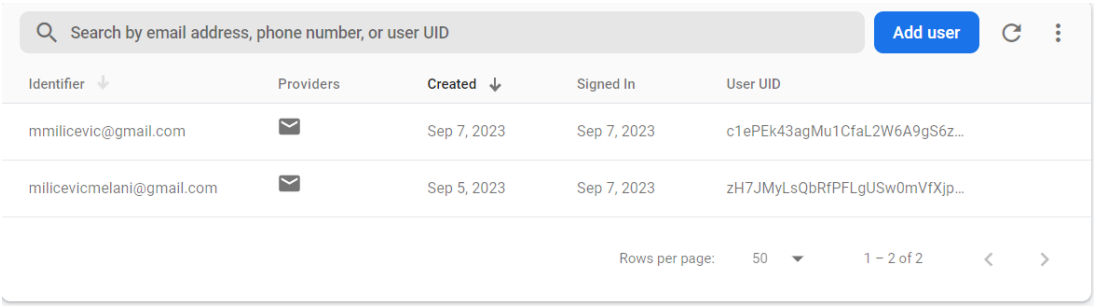
```
class AuthService{
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Stream<String?> get onAuthStateChanged => _auth.authStateChanges().map(
        (User? user) => user?.uid,
  );

  Future<UserCredential> signUpWithEmailAndPassword(String email, String password, String name) async {
    UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
    DataService().saveUserInfo(userCredential, email, name);

    return userCredential;
  }
```

Slika 4.5 Klasa *AuthService* i metoda *signUpWithEmailAndPassword*

```
Future<void> saveUserInfo(UserCredential userCredential, String email,
    String name) async {
  await _databaseReference.child('users').child(userCredential.user!.uid).set(
      {
        'fullName': name,
        'email': email,
      });
}
```

Slika 4.6 Metoda *saveUserInfo* klase *DataService*

Na slici 4.7 prikazan je izgled spremljenog korisnika u *Firebase Authenticationu*. Svakom novom korisniku dodijeljen je jedinstveni identifikator (*User UID*).



Slika 4.7 Spremljeni korisnici u *Firebase Authenticationu*

Kod prijave korisnika poziva se metoda *signInWithEmailAndPassword* (Slika 4.8) koja pomoću *Firebase Authenticationa* provjerava uneseni *e-mail* i lozinku. Ako je prijava uspješna otvara se profil korisnika. Prijava će biti neuspješna ako korisnik unese netočne podatke i u tom slučaju ponovno će morati unijeti podatke i neće moći ući u profil dok ne unese točne (Slika 4.9).

```
Future<UserCredential> signInWithEmailAndPassword(String email, String password) async {
  UserCredential userCredential = await _auth.signInWithEmailAndPassword(
    email: email,
    password: password,
  );
  return userCredential;

}
```

Slika 4.8 Metoda *signInWithEmailAndPassword*

```
void signIn()async{
  final auth = AuthProvider.of(context).auth;
  UserCredential userCredential;
  String uid ;
  try{
     userCredential=await auth.signInWithEmailAndPassword(
        _emailController.text, _passwordController.text);
    uid = userCredential.user!.uid;
    Navigator.of(context).pushReplacement(MaterialPageRoute(
      builder: (context) => ProfilePage(uid: uid),
    ),  // MaterialPageRoute
    );

  }catch(e){
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
        ├─ content: Text('Something went wrong, try again!')));  // SnackBar
  }
}
```

Slika 4.9 Metoda *signIn*

4.2.3. Postavljanje i prikaz korisničkog profila

Nakon što se korisnik uspješno registrira, odabire početne postavke manikure. Na slici 4.10 i u tablici 1. vide se opcije za odabir dužine (*length*), oblika (*shape*) i tehnike (*technique*) manikure.

```
enum NailArtBasics {length,shape,technique}
Map<NailArtBasics, List<String>> basicsLists = {
  NailArtBasics.length: ['Short', 'Medium', 'Long','Extra Long'],
  NailArtBasics.shape: ['Round', 'Square', 'Oval','Coffin','Almond'],
  NailArtBasics.technique: ['Basic', 'Shellac', 'Gel','Acrylic'],
};
```

Slika 4.10 Moguće opcije početnih postavki manikure

Pritiskom na gumb poziva se metoda *setUp* (Slika 4.11) gdje *savePreferences* (Slika 4.12) sprema odabir u *Realtime database*, nakon čega se otvara profil korisnika.

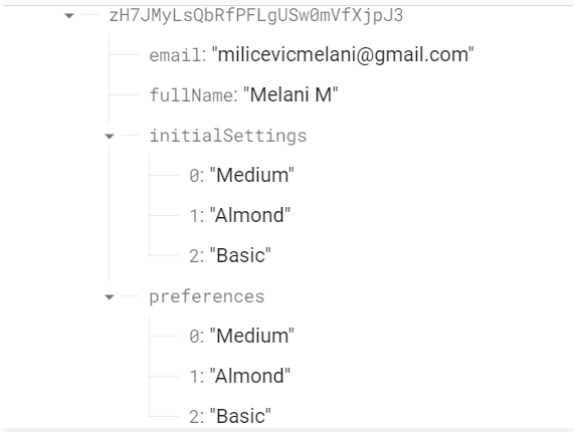
```
void setUp()async{
  String uid ;
  DataService().savePreferences(widget.userCredential, selectedLength!, selectedShape!, selectedTechnique!);
  uid =widget.userCredential.user!.uid;
  Navigator.of(context).pushReplacement(MaterialPageRoute(
      builder: (context) => ProfilePage(uid: uid)));  // MaterialPageRoute

}
```

Slika 4.11 Metoda *setUp*

```
Future<void> savePreferences(UserCredential userCredential, String length,
    String shape, String technique) async {
  List<String> preferences = [length,shape,technique];
  await _databaseReference.child('users')
      .child(userCredential.user!.uid)
      .update(
      {
        'initialSettings':preferences,
        'preferences': preferences,
      });
}
```

Slika 4.12 Metoda *savePreferences*

Na slici 4.13 vidljivi su spremljeni podaci u *Realtime database* nakon registracije korisnika.S obzirom na to da se preferencije korisnika dodavanjem novih dizajna mijenjaju, početne preferencije spremaju se dva puta kako bi se u slučaju zahtjeva za resetiranje, postavke mogle vratiti.



Slika 4.13 Spremljeni podaci u *Realtime database*

Kako bi se prikazao profil korisnika potrebni su podaci spremljeni u bazi. Za to je potrebna klasa *UserInformation* (Slika 4.14) i metoda *getUserPreferences*. *Klasa UserInformation* kao atribute

ima korisničke informacije koje se nalaze u bazi. To su: ime (*fullName*), e-mail adresa (*email*), listu preferencija (*preferences*), listu početnih postavki preferencija (*preferences*) , listu spremljenih fotografija u galeriji (*imageUrls*) i profilnu sliku korisnika (*profilePicture*). U listu spremljenih fotografija nalaze se poveznice na fotografije spremljenih u Firebase Storage.

```
class UserInformation{
  final String? fullName;
  final String? email;
  final List? preferences;
  final List? initialSettings;
  final List? imageUrls;
  final String? profilePicture;

  UserInformation(
      {this.fullName,
        this.email,
        this.preferences,
        this.initialSettings,
        this.imageUrls,
        this.profilePicture}
      );
}
```

Slika 4.14 Klasa *UserInformation*

Metoda *getUserPreferences* vraća objekt klase *UserInformation* sa svim spremljenim vrijednostima korisnika koji je prijavljen (Slika 4.15), ako neke informacije nema taj atribut će biti bez vrijednosti (null).

```
Future<UserInformation> getUserPreferences(String uid) async {
  DataSnapshot snapshot = await _databaseReference.child('users')
      .child(uid)
      .get();

  if (snapshot.value != null) {
    Map<String, dynamic> userData = Map<String, dynamic>.from(
        snapshot.value as Map);
    return UserInformation(
        fullName: userData['fullName'] as String?,
        email: userData['email'] as String?,
        preferences: userData['preferences'] as List?,
        initialSettings: userData['initialSettings'] as List?,
        imageUrls: userData['imageUrls'] as List?,
        profilePicture: userData['profilePicture'] as String?
    );
  }
  else {
    throw Exception('User data not found');
  }
}
```

Slika 4.15 Metoda *getUserPreferences*

Na dnu zaslona profila nalazi se navigacijska traka s dvije opcije. Odabirom prve opcije prikazuje se profil sa slikom korisnika, ime korisnika i galerija spremljenih dizajna. U slučaju da korisnik nije postavio svoju sliku na mjestu se nalazi avatar. Odabirom druge opcije s navigacijske trake otvara se zaslon na kojem se prikazuju novi prijedlozi dizajna.

4.2.4. Generiranje novih dizajna

Kada se otvori zaslon za generiranje novih dizajna poziva se metoda *loadImages* koja poziva metodu *fetchImages* klase *DataService* koja uzima sve dizajne iz *Realtime database* i sprema u listu *images*. Osim toga sprema sve preferencije korisnika u listu ključnih riječi (engl. *keywords*).

```
Future<void> loadImages() async {
  List<nailArtImage> newDesigns = await DataService().fetchImages();
  userInformation = await DataService().getUserPreferences(widget.uid!);
  setState(() {
    images = newDesigns;
    isInGallery = List.filled(newDesigns.length, false);
    keywords=List.from(userInformation!.preferences!);
  });
}
```

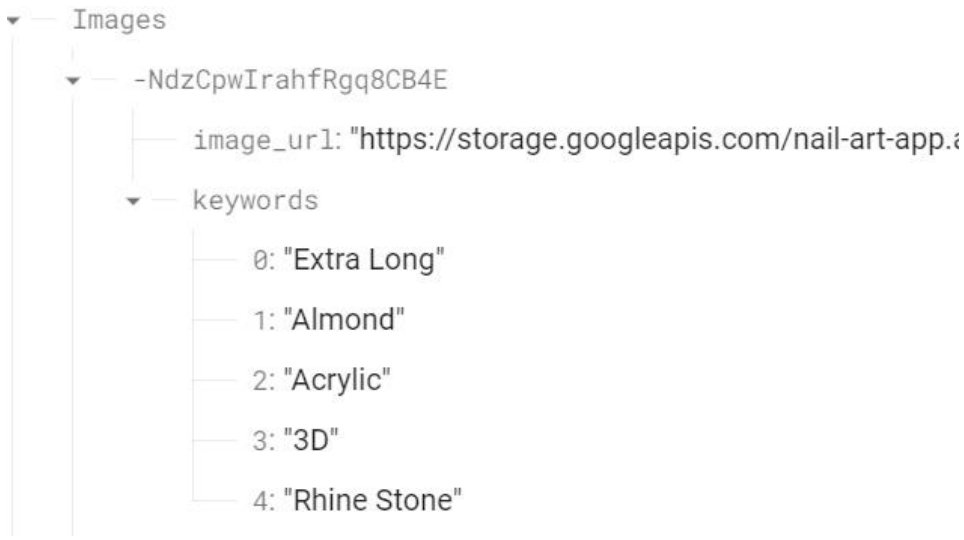
Slika 4.16 Metoda *loadImages*

Metoda *fetchImages* traži referencu na lokaciju baze i na dijete *Images* u bazi. Nakon toga prolazi kroz prikupljene podatke i svaki sprema kao par (*imageUrl*, *keywords*) gdje je *imageUrl* poveznicu na fotografiju i *keywords* lista ključnih riječi koji opisuju fotografiju.

```
Future<List<nailArtImage>> fetchImages() async {
  DatabaseReference ref = FirebaseDatabase.instance.ref().child('Images');
  DataSnapshot dataSnapshot = await ref.get();
  List<nailArtImage>imageList=[];
  if (dataSnapshot.value != null) {
    dynamic imageData = dataSnapshot.value;
      imageData.forEach((key,value) {
        List<dynamic> myList = value['keywords'] as List<dynamic>;
        imageList.add(nailArtImage(image_url: value['image_url'], keywords: myList));
      });

  }
  return imageList;
}
}
```

Slika 4.17 Metoda *fetchImages*

Podaci u *Realtime Database* spremljeni su u skup *Images*. Svaki podatak u *Images* ima poveznicu na fotografiju spremljenu u *Firebase Storage* (*image_storage*) i listu njenih ključnih riječi (*keywords*) (Slika 4.18).



Slika 4.18 Primjer spremljenog podatka *Images* u *Realtime Database*

Slika 4.17 prikazuje kako metoda *fetchImages* vraća Listu tipa *nailArtImage*. Klasa *nailArtImage* za atribute ima *url* (poveznicu) fotografije i listu ključnih riječi koji opisuju dizajn (dužina, oblik, tehnika, boja manikure, itd.). Na slici 4.19 prikazuje klasu *nailArtImage.*

```
class nailArtImage{
  final String image_url;
  final List<dynamic> keywords;

  nailArtImage({
    required this.image_url,
    required this.keywords
  });
}
```

Slika 4.19 Klasa *nailArtImage*

Nakon što se učitaju fotografije, poziva se metoda *checkGallery* (slika 4.20) koja za svaki dizajn u bazi provjerava nalazi li se već u galeriji korisnika ako ne, provjerava odgovara li dizajn korisniku tako što provjeri jesu li barem dvije ključne riječi dizajna iste preferencijama korisnika (Slika 4.21). Ako su zadovoljeni svi uvjeti dizajn se prikaže. Korisnik može preskočiti i tražiti novi dizajn ili označiti da mu se sviđa.

```
Future<void> checkGallery() async {
  for (int i = 0; i < images.length; i++) {
    if (userInformation!.imageUrls !=null && userInformation!.imageUrls!.contains(images[i].image_url)) {
      setState(() {
        isInGallery[i] = true;
      });
    }
  }
}
```

Slika 4.20 Metoda *checkGallery*

```
bool checkKeywords(List<dynamic> first,List<dynamic> second)
{
  Set<String> set1 = Set.from(first);
  Set<String> set2 = Set.from(second);

  Set<String> intersection = set1.intersection(set2);

  if (intersection.length >= 2) {
    return false;
  } else {
    return true;
  }
}
```

Slika 4.21 Metoda *checkKeywords*

Na slici 4.22 prikazana je metoda *skip* koja provjerava ima li novih dizajna. Ako nema, prikaže se poruka *No more designs* (Nema više dizajna). Ako dizajna ima, provjerava se nalazi li se dizajn u galeriji korisnika i odgovara li preferencijama korisnika kako bi se znalo hoće li se dizajn prikazati korisniku ili preskočiti.

```
_skip(){
  if (images.isEmpty) {
    return Text('No more designs');
  }
  while(isInGallery[index] || checkKeywords(images[index].keywords, userInformation!.preferences!)){
    _next();
  }

  return Container(
      height:350,
      width:250,
      child: Image.network(images[index].image_url,height: 300,width: 200,fit:BoxFit.fill,));  // Container
}
```

Slika 4.22 Metoda *skip*

U slučaju da se korisniku sviđa dizajn poziva se metoda *favorite* (slika 4.23) koja, osim što sprema fotografije u galeriju, sprema i ključne riječi kao dodatne preferencije korisnika metodom *saveKeywords*. (slika 4.24)

```
void _favorite() async {
  if (images.isEmpty) {
    _skip();
  }
  if (!isInGallery[index]) {
    await DataService().saveToGallery(widget.uid, images[index].image_url);
    _saveKeywords();
    setState(() {
      isInGallery[index] = true;
    });
    Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (context)=>ProfilePage(uid: widget.uid!)));
  }
}
```

Slika 4.23 Metoda *favorite*

```
_saveKeywords() async{
  for(String keyword in images[index].keywords){
    if(!keywords!.contains(keyword)){
      keywords!.add(keyword);
    }
  DataService().saveKeywords(widget.uid!, keywords!);
}
}
```

Slika 4.24 Metoda *saveKeywords*

4.2.5. Skočni izbornik

U skočnom izborniku nalazi se gumb za uređivanje profila, gumb za brisanje svih fotografija iz galerije i gumb za odjavu korisnika.

4.2.5.1. Edit Profile

Odabirom *Edit Profile* otvara se zaslon za uređivanje profila. Ako korisnik želi promijeniti sliku profila pritiskom na malu ikonu fotoaparata u kutu avatara poziva se metoda *selectImage* (slika 4.25) koja otvara osobnu galeriju korisnika kako bi odabrao novu sliku profila. Za to je korišten *Flutter* dodatak *ImagePicker*. U polje ispod avatara unosi se novo ime, a pritiskom na gumb poziva se metoda *save* koja sprema promjene i vraća se na zaslon profila. Na slici 4.26 prikazana je metoda *save* koja nakon što spremi promjene vrati na zaslon profila.

```
class _EditProfileState extends State<EditProfile> {
  Uint8List? _image;
  final TextEditingController _fullNameController = TextEditingController();
  String passswordChange="";
  bool show=false;

  void selectImage() async {
    Uint8List img = await pickImage(ImageSource.gallery);
    setState(() {
      _image = img;
    });
  }

  pickImage(ImageSource source) async{
    final ImagePicker _imagePicker = ImagePicker();
    XFile? _file = await _imagePicker.pickImage(source: source);
    if(_file != null){
      return await _file.readAsBytes();
    }
    print('No Images Selected');
  }
```

Slika 4.25 Metode *selectImage* i *pickImage*

```
void _save() async{
  await DataService().editData(widget.uid,_fullNameController.text, _image);
  Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (context)=>ProfilePage(uid: widget.uid)));
}
```

Slika 4.26 Metoda *save*

Na slici 4.27 uočava se metoda *newPassword* koja se pozove ako korisnik pritisne „*Change Password*“. Ona otvara skočni prozor i od korisnika se traži da unese trenutnu lozinku i novu. Stara lozinka je potrebna kao dodatna razina sigurnosti.

```
Future<void> _newPassword() async {
  bool success=false;
  show=true;
  final TextEditingController _oldpasswordController = TextEditingController();
  final TextEditingController _newpasswordController = TextEditingController();
```

```
SimpleDialogOption(
  onPressed: () async {
    success=await AuthService().changePassword(_oldpasswordController.text, _newpasswordController.text);
    if (success) {
      passswordChange="Password changes successfully";
      Navigator.of(context).pop();
    } else {
      passswordChange="Something went wrong. Try again";
      Navigator.of(context).pop();
    }
  },
```

Slika 4.27 Metoda *newPassword*

4.2.5.2. Clear Gallery i Sign Out

Ako korisnik odabere *Clear Gallery* pozvat će se metoda *clear* (Slika 4.28) koja briše sve spremljene dizajne iz galerije i vraća preferencije na početne. Kao što je prethodno navedeno, početne preferencije su spremljene kao zasebni atribut pa je vraćanje jednostavno.

```
_clear() async{
  await DataService().clearGallery(widget.uid);
  Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (context) => ProfilePage(uid: widget.uid)));
}
```

Slika 4.28 Metoda *clear*

U slučaju da se korisnik želi odjaviti to može učiniti pritiskom *na Sign out* koja poziva metodu za odjavu korisnika i odlazi na početni zaslon aplikacije. Slika 4.29 prikazuje metodu *signOut* koja odjavi korisnika i vrati na početni zaslon gdje korisnik ponovno može prijaviti se ili registrirati novi račun.

```
_signOut()async{
  AuthService auth = AuthProvider.of(context).auth;
  await auth.signOut();
  Navigator.of(context).pushReplacementNamed('/home');
}
```

Slika 4.29 Metoda *signOut*

4.2.6. Uređivanje galerije

Pritiskom na bilo koju fotografiju u galeriji u skočnom prozoru prikazuje se uvećana slika dizajna i u kutu se nalazi gumb *Delete* koji može obrisati sliku iz galerije metodom *delete*. (Slika 4.30)

```
_delete(String image){
  DataService().deleteFromGallery(widget.uid, image);
}
```

Slika 4.30 Metoda *delete*

5. NAČINI KORIŠTENJA I ISPITIVANJA MOBILNE APLIKACIJE S ANALIZOM REZULTATA

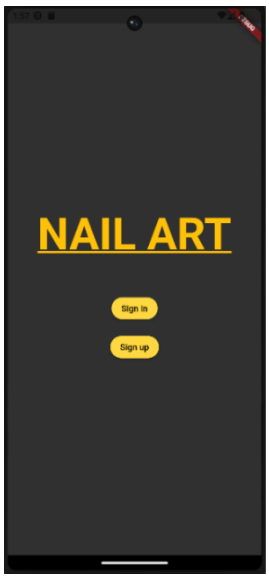
5.1. Korištenje mobilne aplikacije

Kada korisnik prvi puta pokrene aplikaciju otvara se početni zaslon koji korisniku nudi mogućnost prijave u već postojeći profil i registraciju novog korisnika. Odabirom na registraciju od korisnika se traži da unese svoje ime, e-mail adresu i lozinku. U nastavku korisnik može postaviti preferencije dužine i oblika nokta i tehnike manikure. Svi podaci spremaju se i bazu i otvara se profil korisnika. Na stranici profila korisnika u gornjem kutu se nalazi skočni izbornik koji nudi korisniku da uredi profil, izbriše sve dizajne iz galerije ili se odjavi. Kod uređivanja profila korisnik može promijeniti sliku profila, ime i lozinku. U slučaju da korisnik želi promijeniti lozinku potrebno je unijeti i trenutnu lozinku za potvrdu. Pronalazak novih dizajna moguć je pritiskom na desni gumb navigacijske trake koja se nalazi na dnu zaslona. Ako se korisniku ne sviđa dizajn može ga preskočiti ili ako mu se sviđa može ga spremiti u galeriju. Galerija se nalazi na profilu korisnika i svaku sliku u galeriji moguće je uvećati i izbrisati iz galerijom dodirom na nju. Sljedećim ulaskom u aplikaciju ako se korisnik nije odjavio, nije potrebna prijava korisnika i profil se onda prikazuje.

5.2. Prikaz rada mobilne aplikacije

5.2.1. Sučelje početnog zaslona

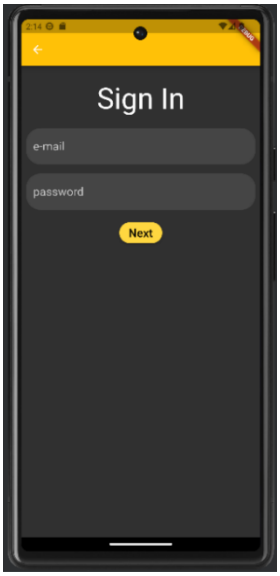
Na slici 5.1 prikazano je sučelje početnog zaslona s gumbima za prijavu i registraciju.



Slika 5.1 Sučelje početnog zaslona

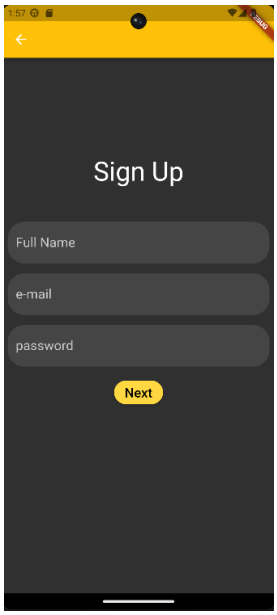
5.2.2. Prikaz zaslona za registraciju i prijavu

Ako se korisnik želi prijaviti u profil potrebno je unijeti točan e-mail i lozinku kao što se vidi na slici 5.2.

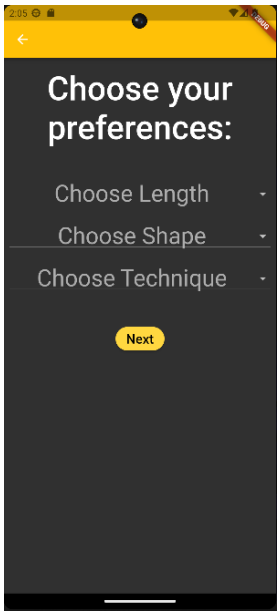


Slika 5.2 Sučelje prijave

Kod registracije od korisnika prvo se traži da unese ime, e-mail i željenu lozinku (slika 5.3), a zatim da odabere neke od ponuđenih postavki dužine i oblika nokta i tehnike manikure (slika 5.4).



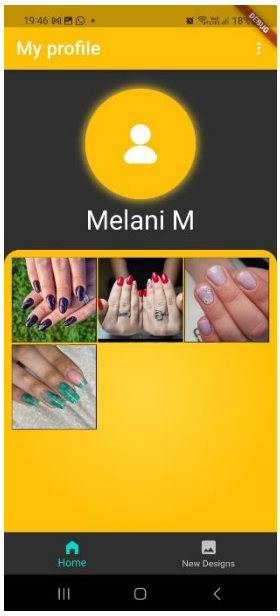
Slika 5.3 Sučelje registracije



Slika 5.4 Sučelje za odabir preferencija

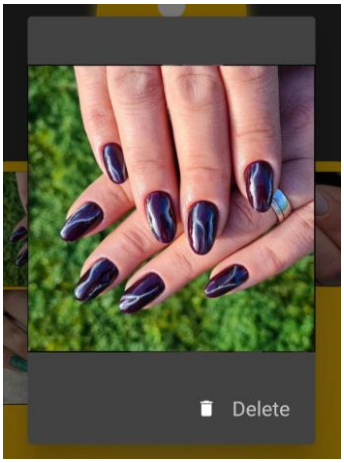
5.2.3. Profil korisnika i generiranje novih dizajna

Slika 5.5 prikazuje profil korisnika. Prvo na profilu je fotografija profila, ako ju korisnik nije postavio prikazat će se generički avatar. Ispod fotografije piše ime korisnika, a ispod toga galerija spremljenih dizajna. Galerija je raspoređena tako da u svaki red stanu po tri kvadratne slike dizajna iste veličine. Ako u galeriji nema fotografija onda se prikaže tekst „*No saved designs*“. Na dnu zaslona je navigacijska traka i dva gumba. Prvi je ikona *Home* koji prikazuje profil, a drugi je ikona *Image* kojim se otvara zaslon za predlaganje novih dizajna.



Slika 5.5 Sučelje profila korisnika

Korisnik može uvećati slike tako da stisne na nju i otvorit će se skočni prozor s odabranom slikom, na dnu prozora nalazi se i gumb koji može ukloniti dizajn iz galerije (Slika 5.6).



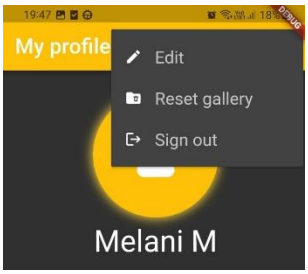
Slika 5.6 Prikaz skočnog izbornika

Na slici 5.7 prikazano je sučelje kada korisnik odabere drugu opciju navigacijske trake. Na zaslonu je prikazan novi dizajn odabran pomoću ključnih riječi i preferencija korisnika. Pritiskom na lijevi gumb ispod slike predlaže se novi dizajn. Ako korisnik želi spremiti predloženi dizajn to može pritiskom na desni gumb.

Slika 5.7 Sučelje za prikaz prijedloga novih dizajna

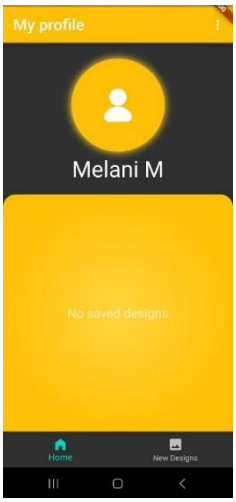
5.2.4. Prikaz skočnog izbornika

U desnom gornjem kutu kod prikaza profila i prijedloga novih dizajna nalaze se tri točkice. Pritiskom na njih otvara se skočni prozor prikazan na slici (Slika 5.8).



Slika 5.8 Skočni izbornik

Ako korisnik izabere *Reset gallery*, sve fotografije iz galerije će se ukloniti i preferencije korisnika vratit će se na one postavljene tijekom registracije (Slika 5.9).

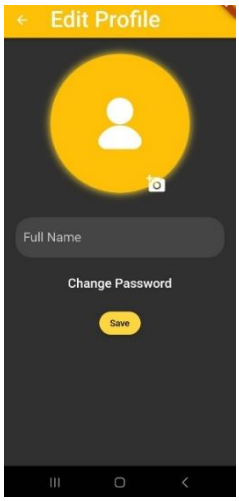


Slika 5.9 Prikaz prazne galerije

Ako se korisnik želi odjaviti i vratiti na početni zaslon to može pritiskom na zadnju opciju izbornika *Sign out*.

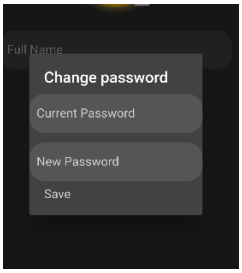
5.2.5. Prikaz zaslona za uređivanje profila

Prva opcija skočnog izbornika *Edit profile* otvara sučelje prikazano na slici 5.10. Tu korisnik može promijeniti postavke profila kao što je slika profila, ime i lozinka.



Slika 5.10 Sučelje zaslona za uređivanje profila

Tipka *Change Password* daje opciju mijenjanja lozinke, ali prvo je potrebno upisati trenutnu lozinku kao dodatan korak sigurnosti (Slika 5.11).

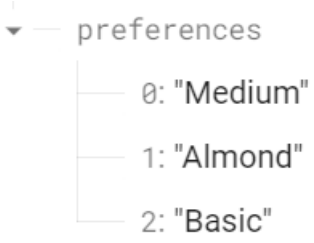


Slika 5.11 Promjena lozinke

5.3. Ispitivanje rada mobilne aplikacije

5.3.1. Prvi ispitni slučaj

U prvom ispitnom slučaju za početne preferencije odabrano je: *Medium* (srednja dužina nokta), *Almond* (oblik badema) i *Basic* (obični lak) kao što se vidi i na slici 5.12. Nakon što je u galeriju dodan dizajn na slici 5.14, preferencije su: *Medium*, *Almond*, *Basic*, gel, *purple* (ljubičasta) i *marble* (mramor) (Slika 5.13). Aplikacija je uspješno predložila dizajn koji ima barem dvije sličnosti sa preferencijama korisnika (*Medium* i *Almond*) i dodala gel, *purple* i *marble* kao nove.



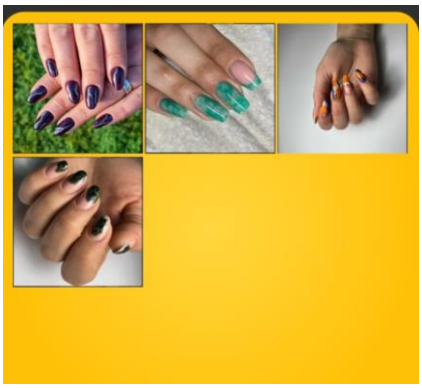
Slika 5.12 Prikaz početnih preferencija u Realtime Database

Slika 5.13 Prikaz preferencija nakon dodavanja dizajna u galeriju

Slika 5.14 Predloženi dizajn aplikacije

5.3.2. Drugi ispitni slučaj

U drugom slučaju promatra se što se događa kada se doda nekoliko novih dizajna kao na slici 5.15. Kao što se vidi na slici 5.16 lista ključnih riječi se uvelike proširila, a može se primijetiti i raznolikost u dizajnima.



Slika 5.15 Prikaz dodanih dizajna

Slika 5.16 Lista ključnih riječi nakon dodavanja više dizajna

6. ZAKLJUČAK

Popularnost manikiranja je svakim danom sve veća. Neki od glavnih karakteristika dizajna manikure su dužina i oblik nokta, tehnike izrade i tehnike ekspresije. Cilj ove mobilne aplikacije je pomoći korisniku pri odabiru usluga manikiranja s unaprijed definiranim preferencijama. Aplikacija pojednostavljuje odabir tako što od korisnika traži da unese tijekom registracije preferiranu dužinu, oblik nokta i tehniku izrade na osnovu kojih može predlagati nove dizajne. Svakim spremanjem novog dizajna proširuju se preferencije korisnika i veća je raznolikost predloženih dizajna. Prijavom korisnika, on ima uvid u spremljene dizajne i mogućnost zahtijevanja novih.

Mobilna Android aplikacija razvijena je u programskom okruženju Android Studio, korištenjem razvojne platforme Flutter i programskog jezika Dart. Za autentifikaciju korisnika i spremanje podataka u bazu korištene su usluge platforme Google Firestore. Aplikacija je ispitana kroz par slučajeva i utvrđena je ispravnost rada. Rezultati pokazuju da aplikacija ispravno sprema nove preferencije nakon dodavanja novih dizajna i uspješno pokazuje nove dizajne korisniku. U budućnosti, moguće je proširenje aplikacije uvođenjem pružatelja usluge manikure kao korisnika.

LITERATURA

[1] E. S.T., »Nail Art Design: History and Future,« Indonesia Design:, 27. Rujan 2019. Dostupno na: https://indonesiadesign.com/story/nail-art-design-history-and-future. [Pristupano 25. lipanj 2023.].

[2] H.-F. Hung, »Understanding the Features of Ingredients of Manicures,« u *2017 International Conference on Applied System Innovation (ICASI)*, Sapporo, 2017.

[3] M. Gatica-Ortega, M. Pastor-Nieto i J. Silvestre-Salvador, »Allergis Contact Dermatits Caused by Acrylates in Long-Lasting Nail Polish,« *Actas Dermo-Sifiliográficas*, svez. 109, br. 6, pp. 508-514, 2018.

[4] B. Lapolla, »From Price to Quality to Longevity: Here's Your Official Guide to Every Type of Manicure,« PureWow, 24. Listopad 2022. Dostupno na: https://www.purewow.com/beauty/manicure-guide. [Pristupano 25. lipanj 2023.].

[5] R. Norris, »All of the Different Nail Shapes, Explained—From Ballerina to Squoval,« Byrdie, 12. Siječanj 2023. Dostupno na: https://www.byrdie.com/different-nail-shapes-7093193. [Pristupano 25. lipanj 2023.].

[6] K. Jeongmee i J. Seungeun, »A Study on Expression Techniques of Nail Art : Focused on Nail Holic in 2012,« *Journal of Fashion Business*, pp. 101-115, Prosinac 2014..

[7] »About Us,« Perfect Corp, 2023. Dostupno na: https://www.perfectcorp.com/business/info/aboutus. [Pristupano 25. lipanj 2023.].

[8] Umm-e-Habiba i S. Asghar, »A Survey on Multi-Criteria Decision Making,« u *2009 International Conference on Emerging Technologies*, 200p.

[9] J. Smock, »How to Pick the Best Nail Shape for Your Hands,« Real Simple, 12. travnja 2022. Dostupno na: https://www.realsimple.com/beauty-fashion/skincare/hands-feet/nail-care/nail-shapes. [Pristupano 11. rujna 2023.].

[10] Expert, »Nail Length Chart: How to Choose Ideal Length and Shape,« Nail Designs Journal, 23. svibnja 2023. Dostupno na: https://naildesignsjournal.com/nail-length-chart/. [Pristupano 23. rujna 2023.].

[11] D. Šokčević, »@dorisnailartist,« Dostupno na: https://www.instagram.com/dorisnailartist/. [Pristupano 10. rujan 2023.].

[12] »Non-functional Requirements: Examples, Types, How to Approach,« altexsoft, 26. Srpanj 2022. Dostupno na: https://www.altexsoft.com/blog/non-functional-requirements/. [Pristupano 26. lipanj 2023.].

[13] E. Mixon, »Android OS,« TechTarget, Veljača 2023. Dostupno na: https://www.techtarget.com/searchmobilecomputing/definition/Android-OS. [Pristupano 26. lipanj 2023.].

[14] K. Bracey, »What is Figma?,« envatotuts+, 16. Rujan 2022. Dostupno na: https://webdesign.tutsplus.com/articles/what-is-figma--cms-32272. [Pristupano 26. lipanj 2023.].

[15] »developers,« Android, 6. lipanj 2023. Dostupno na: https://developer.android.com/studio/intro. [Pristupano 26 lipanj. 2023.].

[16] »What is Flutter and Its Advantages?,« Adservio, 20. srpanj 2022. Dostupno na: https://www.adservio.fr/post/what-is-flutter-and-what-are-its-advantages. [Pristupano 26. lipanj 2023.].

[17] A. Badkar, »What is Dart Programming? Everything You Need to Get Started!,« simplilearn, 14 srpanj 2023. Dostupno na: https://www.simplilearn.com/what-is-dart-programming-article. [Pristupano 7 rujan 2023].

[18] H. Özşahan, »What is Firebase? Learn The Basics From A to Z,« resmo, 18. Svibanj 2023. Dostupno na: https://www.resmo.com/blog/what-is-firebase. [Pristupano 26. lipanj 2023.].

POPIS SLIKA

Slika 2.2 Isprobavanje dizajna pomoću kamere 5
Slika 2.1 Pravljenje vlastitog dizajna 5
Slika 2.3 Prikaz filtera aplikacije 5
Slika 2.4 Kolekcija dizajna aplikacije 5
Slika 2.5 Sučelje aplikacije Styles4Nailz 6
Slika 3.1 Dijagram toka aplikacije 10
Slika 4.1 Dizajn sučelja aplikacija pomoću aplikacije *Figma* 11
Slika 4.2 Konstuktor *AuthProvider* 13
Slika 4.3 *HomeController widget* 13
Slika 4.4 Metoda *signUp* 14
Slika 4.5 Klasa *AuthService* i metoda *signUpWithEmailAndPassword* 14
Slika 4.6 Metoda *saveUserInfo* klase *DataService* 15
Slika 4.7 Spremljeni korisnici u *Firebase Authenticationu* 15
Slika 4.8 Metoda *signInWithEmailAndPassword* 15
Slika 4.9 Metoda *signIn* 16
Slika 4.10 Moguće opcije početnih postavki manikure 16
Slika 4.11 Metoda *setUp* 17
Slika 4.12 Metoda *savePreferences* 17
Slika 4.13 Spremljeni podaci u *Realtime database* 17
Slika 4.14 Klasa *UserInformation* 18
Slika 4.15 Metoda *getUserPreferences* 18
Slika 4.16 Metoda *loadImages* 19
Slika 4.17 Metoda *fetchImages* 19
Slika 4.18 Primjer spremljenog podatka *Images* u *Realtime Database* 20
Slika 4.19 Klasa *nailArtImage* 20
Slika 4.20 Metoda *checkGallery* 21
Slika 4.21 Metoda *checkKeywords* 21
Slika 4.22 Metoda *skip* 21
Slika 4.23 Metoda *favorite* 22
Slika 4.24 Metoda *saveKeywords* 22
Slika 4.25 Metode *selectImage* i *pickImage* 23
Slika 4.26 Metoda *save* 23

Slika 4.27 Metoda *newPassword* 23
Slika 4.28 Metoda *clear* 24
Slika 4.29 Metoda *signOut* 24
Slika 4.30 Metoda *delete* 24
Slika 5.1 Sučelje početnog zaslona 25
Slika 5.2 Sučelje prijave 26
Slika 5.3 Sučelje registracije 26
Slika 5.4 Sučelje za odabir preferencija 27
Slika 5.5 Sučelje profila korisnika 27
Slika 5.6 Prikaz skočnog izbornika 28
Slika 5.7 Sučelje za prikaz prijedloga novih dizajna 28
Slika 5.8 Skočni izbornik 29
Slika 5.9 Prikaz prazne galerije 29
Slika 5.10 Sučelje zaslona za uređivanje profila 30
Slika 5.11 Promjena lozinke 30
Slika 5.12 Prikaz početnih preferencija u Realtime Database 31
Slika 5.13 Prikaz preferencija nakon dodavanja dizajna u galeriju 31
Slika 5.14 Predloženi dizajn aplikacije 31
Slika 5.15 Prikaz dodanih dizajna 31
Slika 5.16 Lista ključnih riječi nakon dodavanja više dizajna 31

POPIS TABLICA

Tablica 3.1 Moguće vrijednosti dužine, oblika i tehnike nokta 7
Tablica 3.2 Preporučeni oblici nokta prema obliku ruke i prstiju 8
Tablica 3.3 Preporučeni oblici nokta prema duljini nokta 8

SAŽETAK

U ovom završnom radu opisan je razvoj mobilne Android aplikacije za pomoć pri biranju usluga manikiranja korištenjem razvojne platforme Flutter i programskog jezika Dart u programskom okruženju Android Studio. Aplikacija omogućuje registraciju i/ili prijavu korisnika, postavljanje početnih preferencija dužine i oblika nokte i tehnike izrade manikure, prikaz profila korisnika s galerijom spremljenih dizajna i mogućnost zahtjeva za predlaganje novog dizajna uz višekriterijsko odlučivanje. Autentifikacija korisnika i spremanje podataka u bazu ostvareno je pomoću Firebase Authentification, Realtime Database i Storaga. Provedeno je ispitivanje spremanja dizajna i predlaganja novih dizajna za dva ispitna slučaja i utvrđena je ispravnost rada aplikacije.

Ključne riječi: manikura, mobilna Android aplikacija, preporuka usluge, višekriterijsko odlučivanje,

ABSTRACT

A Mobile Android Application to Help Choosing Manicure Services

In this final thesis, the process of developing an Android application that helps with choosing manicure services using the Flutter framework and Dart programming language inside of Android Studio integrated development platform is described. The application enables user registration and/or login, setup of initial preferences for nail length and shape as well as manicure making technique, a user profile display with a gallery of saved designs and the ability to make a request for new designs with multi-criteria decision making. User authentication and database storing are accomplished using Firebase Authentication, Realtime Database and Storage. Testing of saving designs and recommending new designs was conducted for two test cases and correctness of the app has been established.

Keywords: manicure, mobile Android application, service reccomendetion, multi-criteria decision making

PRILOZI

Prilog 1. Završni rad „Mobilna android aplikacija za pomoć pri biranju usluga manikiranja“ u *.docx* formatu.

Prilog 2. Završni rad „Mobilna android aplikacija za pomoć pri biranju usluga manikiranja“ u *.pdf* formatu.

Prilog 3. Programski kod mobilne aplikacije