

# Mobilna Android aplikacija za prilagodljivo kupovanje proizvoda zasnovano na sustavu preporuka

---

**Goluža, Dino**

**Undergraduate thesis / Završni rad**

**2002**

*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://um.nsk.hr/um:nbn:hr:200:036115>

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

*Download date / Datum preuzimanja:* **2024-11-25**

*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 preddiplomski studij**

**MOBILNA ANDROID APLIKACIJA ZA  
PRILAGODLJIVO KUPOVANJE PROIZVODA  
ZASNOVANA NA SUSTAVU PREPORUKA**

**Završni rad**

**Dino Goluža**

**Osijek, 2022.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 14.09.2022.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na  
preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Dino Goluža
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4200, 24.07.2018.
<b>OIB Pristupnika:</b>	16190319795
<b>Mentor:</b>	Prof.dr.sc. Goran Martinović
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Mobilna Android aplikacija za prilagodljivo kupovanje proizvoda zasnovano na sustavu preporuka
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	U završnom radu treba proučiti i opisati načine planiranja nabavke/kupovanja hrane i kozmetičkih proizvoda uzimajući u obzir zdravstvene potrebe, alergije i ostala ograničenja kupaca, kao i zaštitu okoliša. Na temelju zdravstvenih potreba, alergija i ograničenja kupaca, te kriterija zaštite okoliša, koristeći mogućnost prepoznavanja svojstava proizvoda očitavanjem bar koda i QR koda proizvoda, treba definirati model i programsku arhitekturu mobilne aplikacije, kao i sustav preporuka za kupovanje proizvoda zasnovan na višekriterijskim postupcima odlučivanja. Nadalje, treba razviti mobilnu aplikaciju s bazom podataka i ugrađenim sustavom preporuka za kupovanje proizvoda koja će omogućiti prepoznavanje svojstava proizvoda na temelju očitanoog koda, definiranje profila korisnika, te davanje preporuka za kupovanje proizvoda. Mobilnu aplikaciju treba ispitati i analizirati za odgovarajuće profile korisnika i skupine proizvoda. Tema rezervirana za: Dino Goluža
<b>Prijedlog ocjene završnog rada:</b>	Vrlo dobar (4)

<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	14.09.2022.
<b>Datum potvrde ocjene od strane Odbora:</b>	21.09.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 21.09.2022.

**Ime i prezime studenta:**

Dino Goluža

**Studij:**

Preddiplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

R4200, 24.07.2018.

**Turnitin podudaranje [%]:**

12

Ovom izjavom izjavljujem da je rad pod nazivom: **Mobilna Android aplikacija za prilagodljivo kupovanje proizvoda zasnovano na sustavu preporuka**

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

i sumentora ,

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

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak završnog rada .....	1
2. POTPORA RAČUNALNIH TEHNOLOGIJA PRI KUPOVINI .....	2
2.1. Alergija .....	2
2.2. Alergeni u hrani .....	2
2.3. Barkod.....	2
2.4. QR kod.....	3
2.5. Slična programska rješenja .....	4
2.5.1. Food Scanner .....	5
2.5.2. INFOOD .....	6
3. MODEL I ARHITEKTURA APLIKACIJE ZA PRILAGODLJIVO KUPOVANJE PROIZVODA.....	7
3.1. Funkcijske i ne funkcijske značajke aplikacije za kupovanje proizvoda.....	7
3.2. Potrebne radnje i programski pristup za uspješan rad aplikacije.....	8
3.2.1. Kreiranje korisničkog računa.....	8
3.2.2. Skeniranje barkoda i generiranje QR koda.....	9
3.2.3. Model baze podataka .....	9
3.3. Arhitektura aplikacije.....	9
4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE .....	11
4.1. Programske tehnologije korištene pri izradi aplikacije.....	11
4.1.1. Mobilni operacijski sustav Android .....	11
4.1.2. Programski jezik Kotlin.....	11
4.1.3. Android Studio .....	12
4.1.4. Firebase.....	13
4.2. Programsko rješenje aplikacije sa strane korisnika .....	14

4.2.1.	Zaslon za prijavu .....	14
4.2.2.	Zaslon za registraciju korisnika.....	15
4.2.3.	Zaslon za dodavanje alergija .....	16
4.2.4.	Glavni zaslon aplikacije.....	17
4.2.5.	Zaslon generiranja QR koda.....	20
4.3.	Programsko rješenje na bazi poslužitelja .....	21
4.3.1.	Room baza podataka.....	21
5.	RAD APLIKACIJE, ISPITIVANJE I ANALIZA REZULTATA.....	24
5.1.	Upute za korištenje aplikacije za kupovinu proizvoda .....	24
5.2.	Uvjeti i korisnički slučajevi za ispitivanje rada aplikacije.....	31
5.3.	Analiza rada aplikacije.....	33
6.	ZAKLJUČAK .....	34
	Literatura .....	35
	Životopis.....	36
	ABSTRACT .....	38
	PRILOZI.....	39

## **1. UVOD**

Razvojem tehnologije i napretkom čovječanstva, čovjek nastoji olakšati i ubrzati što više zadataka koje obavlja. Danas svatko posjeduje pametni uređaj preko kojeg obavlja razne zadatke, od plaćanja računa do kupovanja na internet trgovinama. U ovom završnom radu potrebno je programski ostvariti mobilnu aplikaciju za kupovanje namirnica unutar trgovine uz pomoć pametnog uređaja koji bi trebao ubrzati proces kupovine.

Predloženi model je ugrađen u mobilno okruženje koje podržava registriranje i prijavu korisnika, odabir i spremanje alergija korisnika, skeniranje i prepoznavanje proizvoda zajedno s alergenima koje sadrži te kupovina istih.

U poglavlju dva opisano je i dani su opći podaci o barkodovima, QR kodovima, alergijama, alergenima te postojeća slična rješenja. U poglavlju tri prikazana je arhitektura same aplikacije. Poglavlje četiri prikazuje i objašnjava tehnologije korištene za izradu aplikacije. Poglavlje pet prikazuje rad aplikacije i postupak testiranje i analize projektnog zadatka.

### **1.1. Zadatak završnog rada**

U završnom radu potrebno je proanalizirati i objasniti načine planiranja nabavke/kupovanja hrane i kozmetičkih proizvoda uzimajući u obzir zdravstvene potrebe, alergije i ostala ograničenja kupaca, kao i zaštitu okoliša. Na temelju zdravstvenih potreba, alergija i ograničenja kupaca, te kriterija zaštite okoliša, koristeći mogućnost prepoznavanja svojstava proizvoda očitavanjem barkoda i QR koda proizvoda, treba definirati model i programsku arhitekturu mobilne aplikacije, kao i sustav preporuka za kupovanje proizvoda zasnovan na višekriterijskim postupcima odlučivanja. Nadalje, potrebno je razviti aplikaciju za mobilne uređaje s bazom podataka i ugrađenim sustavom preporuka za kupovanje proizvoda koja će omogućiti prepoznavanje svojstava proizvoda na temelju očitanoog koda, definiranje profila korisnika, te davanje preporuka za kupovanje proizvoda. Mobilnu aplikaciju treba testirati i analizirati za različite profile korisnika i skupine proizvoda.



## **2. POTPORA RAČUNALNIH TEHNOLOGIJA PRI KUPOVINI**

U ovom dijelu završnog rada će biti objašnjeno što je alergija, što su i koji su alergeni u hrani te neke osnovne informacije o barkodu i QR kodu. Zatim će biti prikazane slične aplikacija koje trenutno postoje i koje je moguće koristiti.

### **2.1. Alergija**

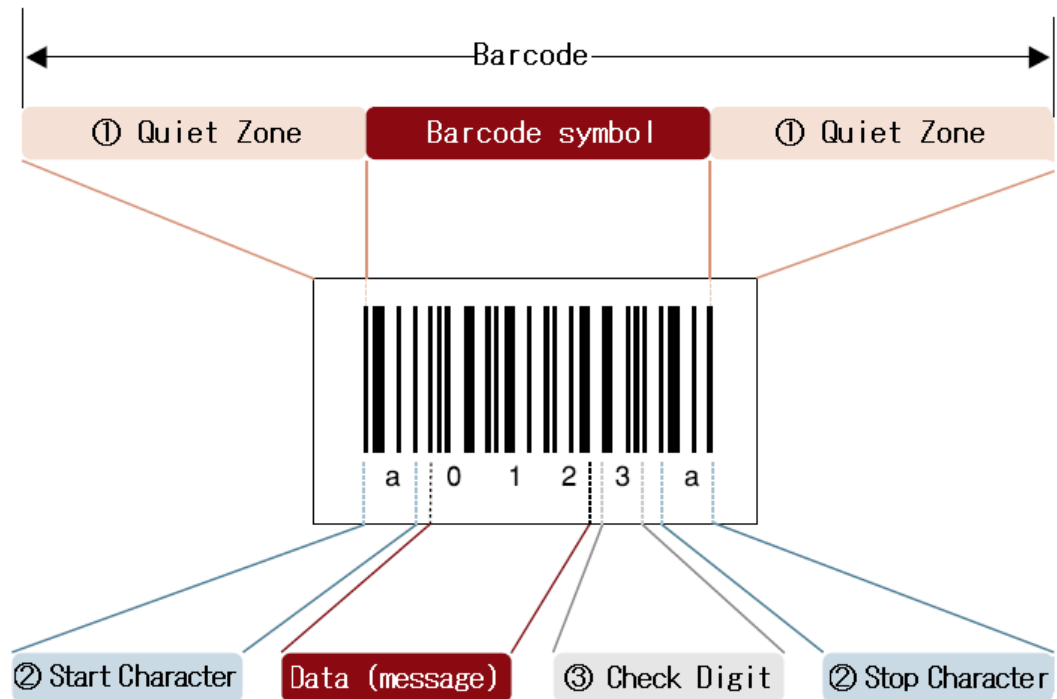
Prema [1], alergija je neuobičajeni i neprimjereni odgovor našeg imunološkog sustava na različite čimbenike iz okoliša koji su uobičajeno neopasne tvari, a nazivaju se antigenima ili alergenima. Simptomi alergije reakcije mogu biti opći ili lokalizirani na organ, odnosno organski sustav, putem kojeg je alergen ušao u tijelo. Ponekad se lokalizirani simptomi razvijaju izrazito brzo te mogu ugroziti život bolesnika. Alergiju može imati svatko, neovisno o dobi i spolu, iako su češće kod osoba koje imaju alergijske bolesti u obitelji.

### **2.2. Alergeni u hrani**

Prema [2], alergije na hranu je svaka reakcija imunološkog sustava koja se javlja nakon konzumiranja neke određene hrane, odnosno odgovor imunološkog sustava na prehrambene bjelančevine. Prema [3], alergija na hranu ima gotovo svaka hrana ili dodatak hrani koji može izazvati alergijsku reakciju, ali najčešće su namirnice koje sadrže alergene: jaja, žitarice, rakove, ribu, kikiriki, soju, mlijeko, orašasto voće, celer, sezam. Prema [4], procjenjuje se da alergije na hranu ima oko 8% djece ispod 7 godina starosti te oko 4% odraslih. Također, većina djece starenjem ipak prestanu biti alergična na neku hranu. Lako je zamijeniti alergije na hranu i intoleranciju na hranu. Intolerancija na hranu je manje ozbiljan problem i ne uključuje imunološki sustav.

### **2.3. Barkod**

Prema [5], barkod ili crtični kod je niz crnih i bijelih linija koji nakon skeniranja omogućuju uvid u informacije o proizvodu. Ranije su prodavači/prodavačice morali znati napamet cijenu ili šifru proizvoda, dok danas uz pomoć čitača za barkod računalo samo prepoznaje o kojem se proizvodu radi. Barkod je jedan od izuma koji se susreće svakodnevno. Uvođenjem barkoda ubrzao se protok informacija i povećala se pouzdanost. Barkodovi imaju primjenu u industriji, trgovanju, zdravstvu i vojsci. Prema [5], barkod se sastoji od tri zone. Prva i zadnja zona je tiha zona, a između njih se nalazi barkod znak. Unutar tihe zone nalazi se prazan prostor koji je veličine 2.5mm. Zona barkod znaka započinje početnim znakom, nakon njega slijede podateci, zatim kontrolni broj i na kraju završni znak. Na slici 2.1. prikazan je barkod, a na slici 2.2. prikazan je čitač barkoda.



Slika 2.1. Primjer barkoda



Slika 2.2. Čitač barkoda

## 2.4. QR kod

Prema [6], QR kod je podvrsta barkoda a sastoji se od dvodimenzionalnog koda koji se sastoji od crnih pravokutnika i kvadrata raspoređenih u kvadratni uzorak na bijeloj pozadini. Prvi QR kod sustav je izumljen 1994. od japanske tvornice Toyote. QR kod može spremiti puno više podataka

od standardnog barkoda i može biti očitano čak i ako je oštećen do 30%. Uzorak na QR kodu predstavlja binarni kod koji se skriva podatke. QR čitač prepoznaje oznaku na temelju tri velika kvadrata izvan QR koda. Kada prepozna ta 3 kvadrata, zna da sve što se nalazi unutar njih predstavlja QR kod. Na slici 2.3. prikazan je primjer QR koda.



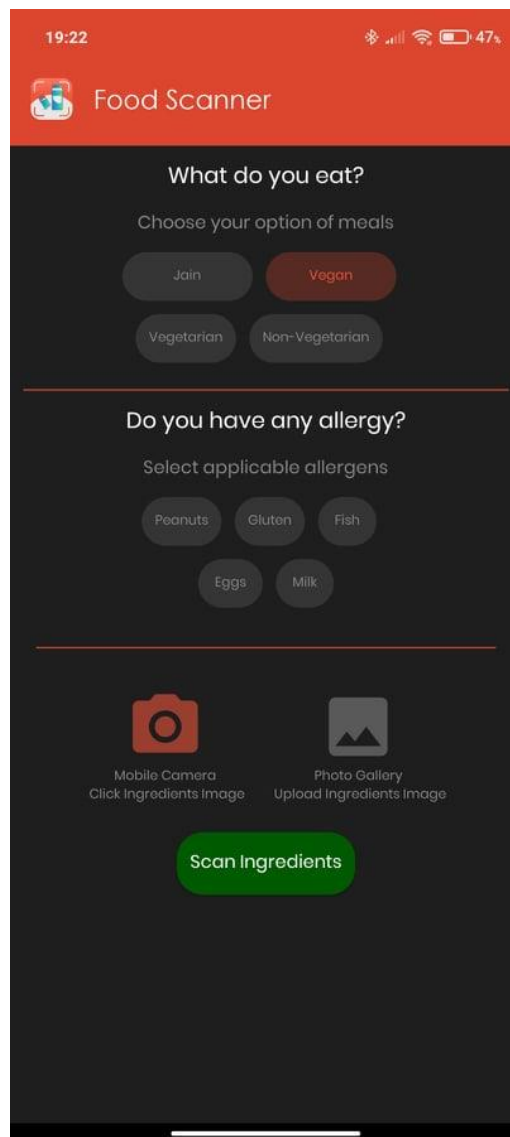
Slika 2.3. Primjer QR koda

## 2.5. Slična programska rješenja

U ovom dijelu će biti prikazana rješenja koja se nalaze na platformi Trgovina Play jer se ondje nalaze mobilne aplikacije isključivo za Android operacijski sustav poput aplikacije koja je opisana u ovom završnom radu. Trenutno nema aplikacije na platformi koja ima iste funkcionalnosti kao i ova, ali ima nekoliko aplikacija koje pomoću barkoda prepoznaju proizvod te pomoću njega provjeravaju alergene. Također, niti jedna aplikacija nije s našeg tržišta tako da ne podržava naše proizvode.

### 2.5.1. Food Scanner

Food Scanner je aplikacija koja skenira sastojke pogodne za jain, vegane, vegetarijance i ljude alergične na kikiriki, gluten, ribu, jaja i/ili mlijeko. Na slici 2.4. prikazana je aplikacija Food Scanner



Slika 2.4. Zaslona aplikacije Food Scanner

## 2.5.2. INFOOD

INFOOD aplikacija omogućuje da se odabere velika količina sastojaka te skeniranjem barkoda proizvoda, aplikacija obavještava korisnika ima li proizvod te sastojke ili ne. Aplikacija radi i kada pristup internetu nije omogućen. Na slici 2.5. i 2.6. prikazan je izgled INFOOD aplikacije.



Slika 2.5. Zaslona aplikacije INFOOD



Slika 2.6. Zaslona aplikacije INFOOD

### **3. MODEL I ARHITEKTURA APLIKACIJE ZA PRILAGODLJIVO KUPOVANJE PROIZVODA**

U ovom dijelu radu objašnjen je model aplikacije te njena arhitektura. Također, objašnjeni su i zahtjevi same aplikacije kao i korišteni alati pri izradi aplikacije.

#### **3.1. Funkcijske i ne funkcijske značajke aplikacije za kupovanje proizvoda**

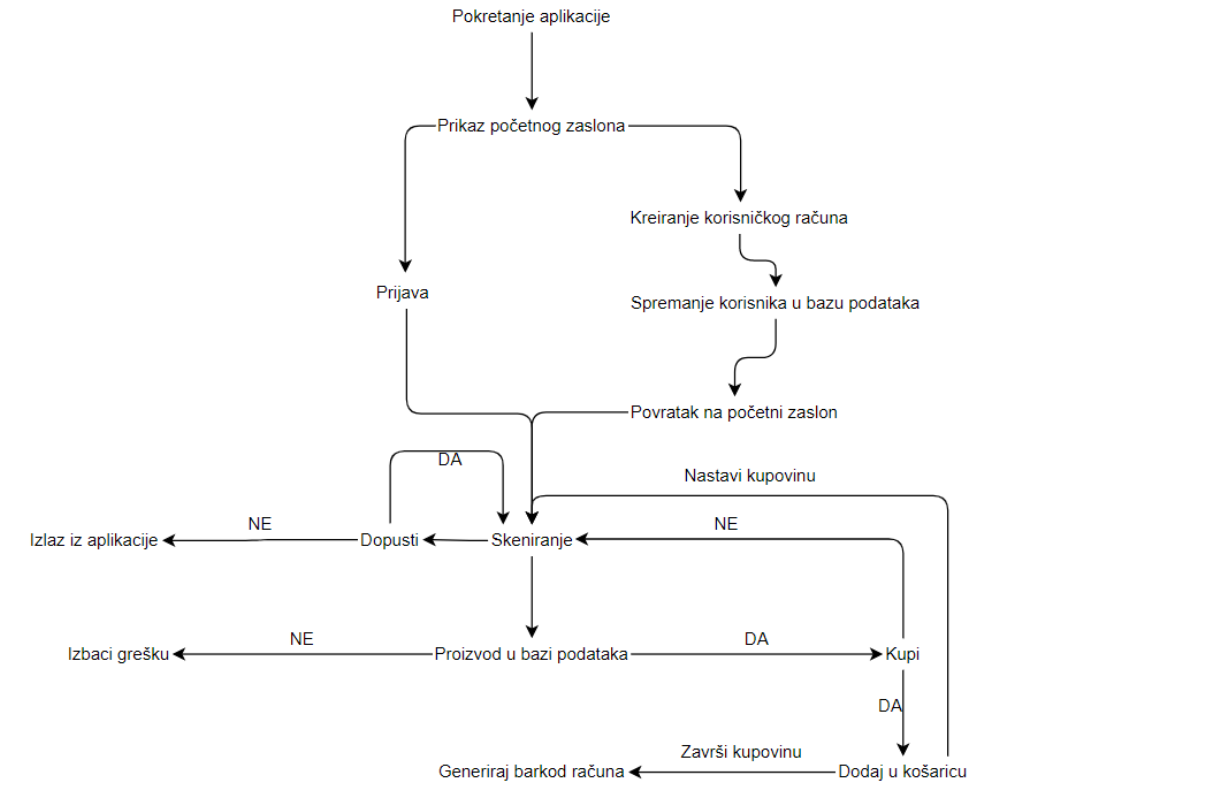
Zbog jednostavnijeg prikaza rada aplikacije u ovome dijelu će biti opisane značajke aplikacije te će biti prikazan dijagram rada aplikacije. Prilikom prvog korištenja aplikacije korisnik se mora ili prijaviti ili registrirati kako bi njegovi alergeni ostali zapamćeni. Klikom korisnika na registraciju, aplikacija ga vodi *Activity* na kojem će kreirati profil, a uspješnom prijavom korisnika aplikacija ga vodi na *Main Activity*. Korisnik preko *Main Activity*-a može pristupiti te promijeniti svoje podatke ili krenuti u kupnju.

Funkcionalni zahtjevi aplikacije su :

- Pri ulasku u aplikaciju korisnik kreira korisnički račun
- Spremanje korisničkog računa u bazu podataka
- Odabir alergena na koje je korisnik alergičan
- Skeniranje barkoda proizvoda te uspoređivanje alergena unutar proizvoda s korisnikovim alergenima
- Generiranje QR koda koji služi kao završni QR kod računa za plaćanje na blagajni

Ova aplikacija zahtjeva minimalnu verziju Androida 6.0 *Marshmallow*. Svi korisnici koji imaju verziju Androida ispod minimalne verzije neće biti u mogućnosti koristiti aplikaciju. Trenutna najnovija verzija Androida je 11.0, ali izlaskom novih verzija Androida korisnici neće biti u mogućnosti koristiti aplikaciju ukoliko ne bude nadogradnji.

Na slici 3.1 može se vidjeti dijagram rada aplikacije. Dijagramom je opisano što korisnik može raditi u samoj aplikaciji te njene funkcionalnosti. Iz dijagrama je vidljivo da je korisnik obavezan napraviti korisnički račun kako bi koristio aplikaciju.



Slika 3.1. Arhitektura aplikacije

## 3.2. Potrebne radnje i programski pristup za uspješan rad aplikacije

U ovom dijelu završnoga rada bit će objašnjeni i opisani algoritmi određenih postupaka u aplikaciji te pozadinskih radnji same aplikacije koje korisnik ne može vidjeti.

### 3.2.1. Kreiranje korisničkog računa

Za ulazak u aplikaciju potrebno je napraviti korisnički račun pomoću kojeg se prijavljuje u aplikaciju te koristi. Za kreiranje korisničkog računa koristi se korisnikova adresa e-pošte te lozinka koja treba biti najmanje šest znakova duga. Nakon toga korisnik može odabrati alergene. Trenutno u aplikaciji postoji najčešći alergeni u hrani te korisnik odabire na što je alergičan. Korisnik može odabrati alergije na: kikiriki, orašasto voće, riba, školjke, jaja, mlijeko, žito, soja, rakovi te badem. Na temelju tih podataka aplikacija zna korisnikove alergije te ih uspoređuje s sastojcima unutar proizvoda koje želite kupiti.

### 3.2.2. Skeniranje barkoda i generiranje QR koda

Skeniranje barkoda obavlja se pomoću *ZXingScannera* koji je implementiran s interneta. Pritiskom na tipku „*scan barcode*“ uključuje se kamera te ako korisnik prvi puta skenira u aplikaciji potrebno je dozvoliti pristup kameri. Pomoću kamere mobilnog uređaja korisnik skenira barkod te je rezultat skeniranja neki niz znakova. QR kod pritiskom na tipku „*generate QR Code*“ izbacuje jedinstveni QR kod na zaslon uređaja koji predstavlja listu proizvoda koje korisnik želi kupiti.

### 3.2.3. Model baze podataka

U mobilnoj aplikaciji koristi se *Room Database* baza podataka čije implementiranje i rad će biti opisan u ovom dijelu. Baza podataka je korištena kako bi se omogućilo spremanje korisničkog računa te kako se korisnik ne bi morao svaki puta registrirati pri ulasku u aplikaciju. Korisnički račun se sastoji od adrese e-pošte te *boolean* tipova podataka alergena kikirikija, orašastog voća, ribe, školjki, jaja, mlijeka, žita, soje, rakova i badema. Navedeni podaci se spremaju u tablicu naziva *user* koja ima attribute iste kao korisnički račun. Podatci koji se spremaju u bazu podataka dohvaćaju se pomoću elemenata *checkbox* za alergene te dva *editText-a* za e-poštu i lozinku. Pritiskom na tipku „*SIGN UP*“ trenutni se korisnik sprema u bazu povlačenjem podataka iz nabrojanih elemenata.

## 3.3. Arhitektura aplikacije

U ovom dijelu rada opisana je podjela aplikacije na korisnički dio (eng. *Frontend*) i na dio koji koji se izvršava u pozadini same aplikacije, odnosno poslužiteljski dio (eng. *Backend*).

Dio aplikacije koji je vidljiv korisniku, odnosno korisnički dio:

- Zaslon za registraciju korisnika – zaslon na kojem korisnik pravi svoj korisnički račun
- Zaslon za prijavu korisnika – zaslon na kojem se već postojeći korisnici mogu prijaviti
- Početni zaslon aplikacije – zaslon na kojem se prikazuje popis naših alergija, te proizvodi u košarici
- Zaslon za dodavanje nadogradnje alergena – zaslon na kojem korisnik može dodati nove alergije ukoliko ih otkrije
- Završi zaslon – zaslon na kojem se generira QR kod završnog računa kupovine

Poslužiteljski dio aplikacije koji nije vidljiv korisniku:



- *Room database* - baza podataka gdje se spremaju svi proizvodi
- *Firebase* – baza podataka koja sprema i omogućuje prijavu i registraciju
- Barkod skener - algoritam po kojem radi skeniranje barkoda

## **4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE**

U ovom poglavlju objašnjeno je koje su programske tehnologije korištene, u kojem programskom jeziku je napisana aplikacija te nešto o razvojnoj okolini.

### **4.1. Programske tehnologije korištene pri izradi aplikacije**

U ovom dijelu rada objašnjeni su programi i alati korišteni prilikom izrade mobilne aplikacije.

#### **4.1.1. Mobilni operacijski sustav Android**

Prema [7], Linux je temelj svakog Android operacijskog sustava koji je namijenjen za pametne telefone, tablete te računala koji se nalaze unutar automobila. Također, ovaj operacijski sustav je otvorenog tipa te se sve više nalazi u pametnim televizorima i pametnim satovima. Android je kupljen od strane Google-a 2005. godine tako da su unutar samo Android OS-a implementirani Google servisi kao Gmaila, tražilice Google, Google diska, trgovine aplikacija Google Play-a i slično. Operacijski sustav je poveznica između korisnika i samog uređaja, npr. ukoliko korisnik ima želju pustiti neku pjesmu, sustav mu omogućuje gumb koji korisnik mora pritisnuti a zatim sustav naređuje uređaju na koji način će to obaviti. Kako je Android operacijski sustav otvorenog tipa, svaki proizvođač može prilagoditi sustav svojim kupcima. Tako su mnogi proizvođači tipa Xiaomi, Samsung, LG i drugi napravili svoje verzije operacijskih sustava koji se temelje na Androidu. Prema [8], posljednja inačica Androida OS je Android 13. Sve inačice Android OS-a dobile su ime po slatkijima ( Cupcake, Donut, Ice Cream Sandwich, KitKat,...).

#### **4.1.2. Programski jezik Kotlin**

Prema [9], Kotlin je suvremeni programski jezik predstavljen široj javnosti 2011 godine, a predstavlja alternativu Java programskom jeziku. Oba programska jezika ciljaju na JVM („Java Virtual Machine“). Ideja je bila napraviti novi jednostavan jezik koji se lako uči i koji bi mogao zamijeniti Javu. Budući da je dizajniran za Java virtualni stroj, može čitati i kod koji je napisan u Java programskom jeziku. Trenutno, sve više korisnika pokušava naučiti programirati u Kotlin programskoj jeziku zbog njegove jednostavnosti. Aplikacija za kupovanje koja je opisana u ovom radu napisana je u Kotlin programskom jeziku.

Na slici 4.1 prikazana je razlika između Java i Kotlin programskog jezika.

```

class Person { //JAVA
    private String name;

    public Person(string name) {
        this.name = name ;
    }

    public String getName() {
        return name;
    }
    public void SetName(String name) {
        this.name = name ;
    }
}

```

```

data class Person(val name: String)

```

```

public void createAndPrintPerson() {
    String name = "Ivan" ;
    Person person = new Person(name);

    printName(person.getName());
}

```

```

fun createAndPrintPerson() {
    val name = "Ivan" ;
    val person = Person(name);

    printName(person.name);
}

```

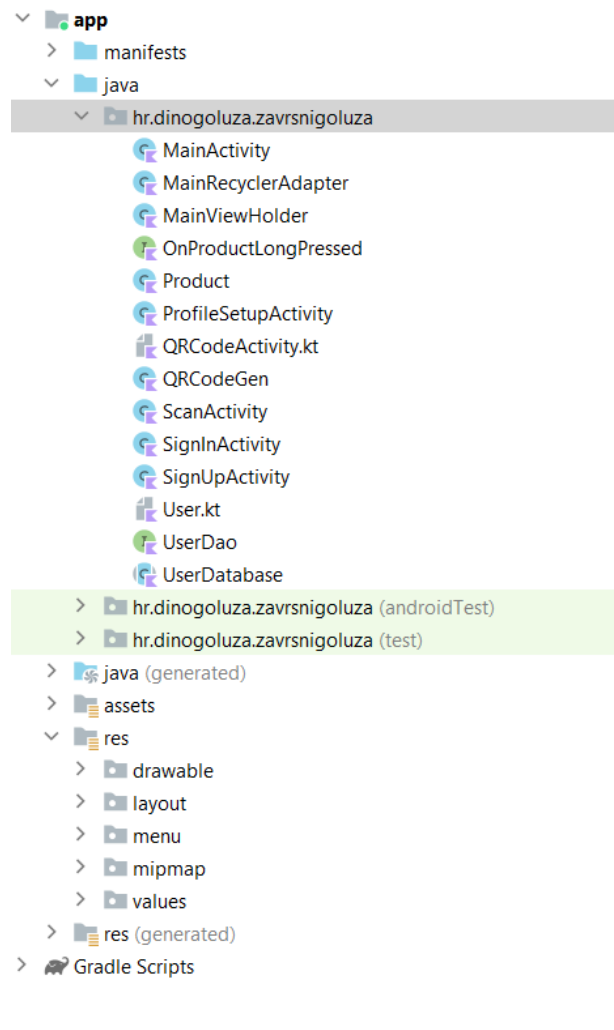
Slika 4.1. Razlika između Java i Kotlin programskog jezika

### 4.1.3. Android Studio

Prema [10], Android Studio je službena i integrirana razvojna okolina za razvoj Android aplikacija za mobitele, pametne televizore, tablete te automobile. Temelji se na Java IDE (*integrated development environment*) te ugrađuje razvojne alate za uređivanje koda. Android Studio je dostupan na svim platformama: Windows, Mac i Linux. Aplikacije pisane u ovom programu podržane su i u Java i u Kotlin programskom jeziku. Android Studio koristi „*Instant Push*“ značajku koja omogućuje instalaciju programirane aplikacije direktno na mobilni uređaj korisnika te nakon toga može pratiti korištenje resursa uživo dok aplikacija radi. Android Studio koristi gradle sustav, emulator, refaktoriranje koda te ima ugrađenu podršku za Github. Emulator se koristi ukoliko korisnik ne želi instalirati aplikaciju odmah na uređaj nego ju instalira na integrirani virtualni uređaj te ju tamo može pokrenuti i testirati. Struktura projekta u Android Studiju je u obliku stabla, a sadrži slijedeće stvari:

- Manifest – u ovoj mapi nalazi se AndroidManifest.xml datoteka u kojoj se može promijeniti ime aplikacije te tražiti razna dopuštenja poput korištenja kamere, mikrofona i slično
- Java – mapa u kojoj se nalaze svi dijelovi programa koji su pisani u Kotlin/Java programskom jeziku
- Res – mapa u kojoj se nalaze svi resursi aplikacije kao što su izgled pojedinih zaslona, slike, boje i slično

Na slici 4.2. prikazana je struktura aplikacije u Android Studiju.



**Slika 4.2.** Primjer strukture aplikacije u Android Studiju

#### 4.1.4. Firebase

Prema [11], Firebase je razvojna platforma za mobilne i web aplikacije. Pomoću njega može se jednostavno i brzo izraditi aplikacije visoke kvalitete. U ovoj aplikaciji Firebase je korišten kako bi se profili korisnika spremali u bazu podataka te implementirao sustav prijave i registracije korisnika u aplikaciju. Pomoću Firebaseove web aplikacije moguće je vidjeti popis svih korisnika te je od tamo moguće ukloniti korisnike ili resetirati njihovu lozinku.

Na slici 4.3. moguće je vidjeti sliku zaslona nekoliko napravljenih korisničkih računa.

Identifier	Providers	Created ↓	Signed In	User UID
marer@gmail.com	✉	Sep 7, 2022	Sep 7, 2022	tal3KOIE3OWkQZzLVfYAokXainn1
babababa@gmail.com	✉	Sep 6, 2022	Sep 6, 2022	3BOMvH0ckReApPvQLFBhvHJcZ...
dinogole@gmail.com	✉	Sep 5, 2022	Sep 7, 2022	qSKYYou8NiVgj9s6yK00PV9UADk1
ll@gmail.com	✉	Aug 29, 2022	Aug 29, 2022	xtr6wwa4GBQ3xe2ttJ7pFnBSqES2
aadd@gmail.com	✉	Aug 23, 2022	Aug 23, 2022	vcj3AjjpVZRNHghHsMiQeiClgNu2
abab@gmail.com	✉	Aug 13, 2022	Aug 13, 2022	Bd4FdFcP0JQ4IGb0xn78nDVBsxn2
dd@gmail.com	✉	Aug 12, 2022	Aug 12, 2022	p1KB7fYRhpDZ1IQABJvomVKZmb...
cc@gmail.com	✉	Aug 12, 2022	Aug 12, 2022	B4I53AcJWUPJSOB3JQ8F6q7hgvv1
bb@gmail.com	✉	Aug 12, 2022	Aug 12, 2022	MHwUCFziwkbFVkc09n2IQXo2ZBy2
aa@gmail.com	✉	Aug 12, 2022	Aug 12, 2022	PVivNFkBitS1UTnG4gw49gdoRnX2
fsg@gmail.com	✉	Aug 12, 2022	Aug 12, 2022	A710qOIFGuZU4VSjeV0T7fHGfPw1

**Slika 4.3.** Firebase baza podataka korisnika

## 4.2. Programsko rješenje aplikacije sa strane korisnika

Ovaj dio rada će potkrijepiti navedene funkcionalnosti s programskim kodom.

### 4.2.1. Zaslona za prijavu

Ulaskom u aplikaciju korisniku se prikazuje ovaj zaslon. Sastoji se od dva *editTexta*, tipke i *TextView*-a koji služi kao tipka. Upisivanjem u *editTextove* e-poštu, a zatim lozinku te pritiskom na tipku „*sign up*“ na kojem je postavljen *listener* aplikacija povlači podatke iz *editTextova* te ih prebacuje u stringove pomoću metode *toString()* i sprema ih u varijable *email* i *password*. Nakon toga, provjerava sadrži li baza podataka traženog korisnika te ukoliko sadrži otvara se glavni zaslon aplikacije koristeći objekt klase *Intent* kako je prikazano na slici 4.4.

```

binding.button.setOnClickListener { it: View!
    val email = binding.emailEt.text.toString()
    val password = binding.passET.text.toString()

    if (email.isNotEmpty() && password.isNotEmpty()) {

        firebaseAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener { it: Task<AuthResult!>
            if (it.isSuccessful) {
                val intent = Intent( packageContext: this, MainActivity::class.java)
                startActivity(intent)
            } else {
                Toast.makeText( context: this, it.exception.toString(), Toast.LENGTH_SHORT).show()
            }
        }
    } else {
        Toast.makeText( context: this, text: "Empty Fields Are not Allowed !!", Toast.LENGTH_SHORT).show()
    }
}

```

Slika 4.4. Programski kod postupka prijave korisnika

Također, potrebno je bilo uvesti Firebaseov API a to je napravljeno dodavanjem koda sa slike 4.5.

```
import com.google.firebase.auth.FirebaseAuth
```

Slika 4.5. Dodavanje Firebaseovog API-ja

#### 4.2.2. Zaslona za registraciju korisnika

Ako korisnik nema napravljen račun tada treba stisnuti *TextView* „Not Registered Yet, Sign Up!“. Nakon toga se otvara zaslon sa tri *EditTexta* gdje korisnik treba unijeti svoju e-poštu koji još uvijek nije u sustavu, te odabrati lozinku s najmanje šest znakova, a zatim ju upisati još jednom kao potvrdu. E-pošta, lozinka i potvrda lozinke čitaju se i spremaju u varijable koje se zatim predaju instanci baze podataka Firebase. Na slici 4.6. je prikazano kako se odvija postupak registracije. Ukoliko je registracija prošla uspješno stvara se početni objekt klase User čiji atributi za alergije su postavljeni u „false“ te se otvara glavni zaslon aplikacije.

```

binding.button.setOnClickListener { it: View!
    val email = binding.emailEt.text.toString()
    val password = binding.passEt.text.toString()
    val confirmPassword = binding.confirmPassEt.text.toString()

    if (email.isNotEmpty() && password.isNotEmpty() && confirmPassword.isNotEmpty()) {
        if (password == confirmPassword) {

            firebaseAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener { it: Task<AuthResult!>
                    if (it.isSuccessful) {
                        userDatabase!!.userDao.saveUser(
                            User(
                                0,
                                email,
                                peanuts = false,
                                treeNuts = false,
                                fish = false,
                                shellfish = false,
                                eggs = false,
                                milk = false,
                                wheat = false,
                                soy = false,
                                crab = false,
                                almond = false
                            )
                        )
                    }
                }
            val intent = Intent(packageContext: this, SignInActivity::class.java)

```

Slika 4.6. Programski kod postupka registracije korisnika

### 4.2.3. Zaslona za dodavanje alergija

Nakon uspješne prijave korisnik ukoliko nema nikakvih alergija može koristiti aplikaciju odmah, ali ukoliko ima alergije na neku hranu treba pritisnuti tri točkice u gornjem desnom uglu i pritisnuti „*update your allergies*“. Nakon toga, korisniku se otvara zaslon na kojem može odabrati 10 različitih alergija. Alergije na koje korisnik može biti pozitivan te ih odabrati su: kikiriki, orašasto voće, riba, školjke, jaja, mlijeko, žito, soja, rakovi i bademi. Alergije su predstavljene checkboxom te odabirom alergija i pritiskom tipke „*save*“ odabrane alergije se spremaju i povezuju zajedno s korisnikovom adresom e-pošte te se vraća na glavni zaslon. Vraćanje na početni zaslon radi se pomoću objekta klase Intent. Na slici 4.7. vidljiv je dio koda kojim se dodaju alergeni. Također, ukoliko korisnik otkrije nove alergije ili vremenom prestane biti alergičan na nešto, uvijek može dodati ili ukloniti određene alergije.

```

binding.btnSave.setOnClickListener { it: View!
    if (binding.cbPeanuts.isChecked) {
        userDatabase!!.userDao.addPeanutsAllergy(user!!.email.toString())
    }
    if (binding.cbTreeNuts.isChecked) {
        userDatabase!!.userDao.addTreeNutsAllergy(user!!.email.toString())
    }
    if (binding.cbFish.isChecked) {
        userDatabase!!.userDao.addFishAllergy(user!!.email.toString())
    }
    if (binding.cbShellfish.isChecked) {
        userDatabase!!.userDao.addShellfishAllergy(user!!.email.toString())
    }
    if (binding.cbEggs.isChecked) {
        userDatabase!!.userDao.addEggsAllergy(user!!.email.toString())
    }
    if (binding.cbMilk.isChecked) {
        userDatabase!!.userDao.addMilkAllergy(user!!.email.toString())
    }
    if (binding.cbWheat.isChecked) {
        userDatabase!!.userDao.addWheatAllergy(user!!.email.toString())
    }
    if (binding.cbSoy.isChecked) {
        userDatabase!!.userDao.addSoyAllergy(user!!.email.toString())
    }
    if (binding.cbCrab.isChecked) {
        userDatabase!!.userDao.addCrabAllergy(user!!.email.toString())
    }
    if (binding.cbAlmond.isChecked) {
        userDatabase!!.userDao.addAlmondAllergy(user!!.email.toString())
    }
    val intent = Intent( packageContext: this, MainActivity::class.java)
    startActivity(intent)
}

```

Slika 4.7. Programski kod dodavanja alergija korisnika

#### 4.2.4. Glavni zaslon aplikacije

Glavni zaslon aplikacije sastoji se od: popisa naših alergija, košarice proizvoda koje korisnik želi uzeti, tipke za skeniranje proizvoda te tri točkice u gornjem desnom uglu gdje se može promijeniti korisnikove alergije, generirati QR kod cijelog računa ili se odjaviti iz aplikacije. Korisnikove alergije izvlače se iz baze podataka preko adrese e-pošte i spremaju se u listu listOfAllergies. Ovaj dio programskom koda vidljiv je na slici 4.8.



```

if (user != null) {
    binding.tvName.text = user.email.toString()

    val currentUser = userDatabase.userDao.getCurrentUser(user.email.toString())

    if (currentUser.peanuts) {
        listOfAllergies.add("peanuts")
    }
    if (currentUser.treeNuts) {
        listOfAllergies.add("tree nuts")
    }
    if (currentUser.fish) {
        listOfAllergies.add("fish")
    }
    if (currentUser.shellfish) {
        listOfAllergies.add("shellfish")
    }
    if (currentUser.eggs) {
        listOfAllergies.add("eggs")
    }
    if (currentUser.milk) {
        listOfAllergies.add("milk")
    }
    if (currentUser.wheat) {
        listOfAllergies.add("wheat")
    }
    if (currentUser.soy) {
        listOfAllergies.add("soy")
    }
    if (currentUser.crab) {
        listOfAllergies.add("crab")
    }
    if (currentUser.almond) {
        listOfAllergies.add("almond")
    }
}

```

**Slika 4.8.** Programski kod dodavanja alergija u listu

Pritiskom na tipku „*scan barcode*“ otvara se zaslon na kojem pomoću kamere korisnik skenira barkodove proizvoda. Nakon što skenira barkod proizvoda iz baze podataka na temelju barkoda izvlače se alergeni koji se nalaze u skeniranom proizvodu te se spremaju u listu naziva `listOfAllergens` što je vidljivo na slici 4.9.

```

val listOfAllergens: ArrayList<String> = arrayListOf()
if (item.peanuts == 1) {
    listOfAllergens.add("peanuts")
}
if (item.treeNuts == 1) {
    listOfAllergens.add("tree nuts")
}
if (item.fish == 1) {
    listOfAllergens.add("fish")
}
if (item.shellfish == 1) {
    listOfAllergens.add("shellfish")
}
if (item.eggs == 1) {
    listOfAllergens.add("eggs")
}
if (item.milk == 1) {
    listOfAllergens.add("milk")
}
if (item.wheat == 1) {
    listOfAllergens.add("wheat")
}
if (item.soy == 1) {
    listOfAllergens.add("soy")
}
if (item.crab == 1) {
    listOfAllergens.add("crab")
}
if (item.almond == 1) {
    listOfAllergens.add("almond")
}
}

```

**Slika 4.9.** Programski kod dodavanja alergena u listu

Nakon toga, metodom `RetainAll()` izbacuju se alergeni iz liste koji ne odgovaraju korisnikovim alergijama te varijabla `count` predstavlja broj alergena koji se podudaraju s korisnikovim alergijama. Zatim se izbacuje prozor koji prikazuje koji je proizvod pronađen, koliko i koji problematični alergeni se nalaze u njemu, te korisnik ima izbor želi li svejedno dodati namirnicu u košaricu ili ne. Objašnjeni dio koda nalazi se na slici 4.10.

```

val count = listOfAllergens.count()

MaterialAlertDialogBuilder(context: this)
    .setMessage(
        "Product detected:" + " " + item.name + "\n" + "Found" + " " + count + " " +
        "problematic allergy:" + " " + listOfAllergens.joinToString(separator: ", ") + "\n" + "Do you still wish to add item to cart?"
    )
    .setPositiveButton("YES") { _, _ ->
        userDatabase.userDao.addToCart(p0.toString())
        val intent = Intent(packageContext: this, MainActivity::class.java)
        intent.putExtra(RESULT, p0.toString())
        startActivity(intent)
    }
    .setNegativeButton("NO") { dialog, _ ->
        dialog.dismiss()
        startActivity(intent)
    }
    .show()

```

Slika 4.10. Programski dio skeniranja proizvoda

Ukoliko je skenirani proizvod problematičan za korisnika proizvod u košarici će biti crvene boje, a ukoliko je siguran biti će žute boje. Također popis alergena iz proizvoda će biti napisan kao potencijalne opasnosti zbog drugih osoba. Dio koda koji je objašnjen prikazan je na slici 4.11.

```

listOfAllergies.retainAll(allergies)
if (listOfAllergies.count() != 0) {
    itemView.setBackgroundColor(Color.parseColor(colorString: "#ff0000"))
}
else itemView.setBackgroundColor(Color.parseColor(colorString: "#FFC801"))

```

Slika 4.11. Programski kod boje proizvoda u košarici

#### 4.2.5. Zaslون generiranja QR koda

Nakon skeniranja svih proizvoda koje korisnik želi uzeti, potrebno je pritisnuti 3 točkice u gornjem desnom uglu te odabrati „generate QR Code“. Aplikacija uzima popis svih namirnica te predaje QR kod na popis istih. Programski kod prikazan je na slici 4.12.

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityQRCodeBinding.inflate(layoutInflater)
    setContentView(binding.root)

    products = intent.getStringArrayListExtra( name: "products")!!

    context = this
    try{
        generateQRCode()
    }catch (e:Exception){
        Toast.makeText( context: this, text: "error generating QR Code", Toast.LENGTH_LONG).show()
    }
}
}

```

Slika 4.12. Programski kod stvaranja QR koda

## 4.3. Programsko rješenje na bazi poslužitelja

### 4.3.1. Room baza podataka

Za potrebe ove aplikacija korištena je lokalnu bazu podataka za proizvode. Prema [12], Room je biblioteka koju preporuča Google kao jedna od komponenti tzv. „*Android Architecture Components*“ pristupa. To je omotač (eng. wrapper) oko SQLite baze te predstavlja sloj apstrakcije koji pojednostavljuje rad s bazom podataka. Room se koristi češće nego SQLite jer je puno jednostavnija, a posjeduje sve karakteristike kao i sam SQLite. Room obuhvaća tri glavne komponente, a to su Database (baza podataka koja sadrži sve podatke te omogućuje povezivanje s aplikacijom), Entity (klasa koja predstavlja tablicu baze podataka) i DataAccessObject – DAO (objekt za pristup podacima koji mora biti ili sučelje ili apstraktna klasa). U skriptu *gradle* potrebno je implementirati ovu bazu podataka kako bi bila korištena što je vidljivo na slici 4.13.

```

//room database:
room_version = "2.4.2"
room = "androidx.room:room-runtime:$room_version"
room_compiler = "androidx.room:room-compiler:$room_version"
room_ktx = "androidx.room:room-ktx:$room_version"

```

Slika 4.13. Implementacija Room baze podataka

Zatim potrebno je kreirati klasu koja predstavlja tablicu u bazi podataka. Baza podataka radi na principu „koliko tablica, toliko i klasa“. Tablica stvorena u aplikaciji ima naziv *Product* i njezini

elementi su id, barkod, naziv, uKošarici, kikiriki, orašasti plodovi, riba, školjke, jaja, mlijeko, žito, soja, rak, bademi. Programski kod za izradu tablice Product vidljiv je na slici 4.14., a na slici 4.15. prikazani su podaci u bazi podataka.

```

@Entity(tableName = "product")
data class Product(
    @PrimaryKey(autoGenerate = true)
    val id: Int,
    val barcode: String,
    val name: String,
    val inCart: Int,
    val peanuts: Int,
    val treeNuts: Int,
    val fish: Int,
    val shellfish: Int,
    val eggs: Int,
    val milk: Int,
    val wheat: Int,
    val soy: Int,
    val crab: Int,
    val almond: Int
)

```

Slika 4.14. Programski kod stvaranja tablice Product

id	barcode	name	inCart	peanuts	treeNuts	fish	shellfish	eggs	milk	wheat	soy	crab	almond
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
0	8076802085831	Semi-white bread	0	0	0	0	0	1	1	1	1	0	0
1	8076802085832	Dukat milk	0	0	0	0	0	0	1	0	1	0	1
2	8076802085833	Gouda cheese	0	0	0	0	0	1	1	0	0	0	0
3	8076802085834	Breaded fish fillet	0	0	0	1	0	1	1	1	0	0	0
4	8076802085835	Nut mix	0	1	1	0	0	0	0	0	0	0	1

Slika 4.15. Prikaz podataka u bazi podataka

Nakon formiranja tablice, nužno je stvoriti DAO i definirati metode za upravljanje tablicom. Metoda „addToCart“ i „removeFromCart“ dodaje, odnosno uklanja proizvode iz košarice. Metoda „getItemsInCart“ vraća listu proizvoda koje je korisnik odlučio uzeti dok metoda „getItemByBarcode“ vraća objekt klase Product. Definirane su vidljive na slici 4.16.

```
@Query("UPDATE product SET inCart='1' WHERE barcode=:barcode")
fun addToCart(barcode: String)

@Query("UPDATE product SET inCart='0' WHERE barcode=:barcode")
fun removeFromCart(barcode: String)

@Query("SELECT * FROM product WHERE inCart='1'")
fun getItemsInCart(): List<Product>

@Query("SELECT * FROM product WHERE barcode=:barcode")
fun getItemByBarcode(barcode: String): Product
```

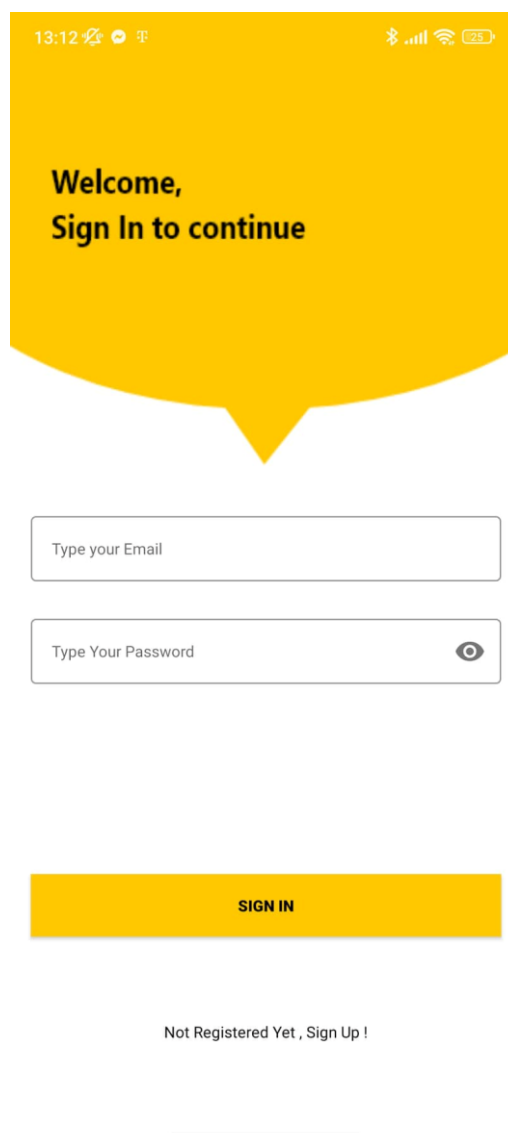
**Slika 4.16.** Metode za upravljanje klasom Product

## 5. RAD APLIKACIJE, ISPITIVANJE I ANALIZA REZULTATA

Ovaj dio rada prikazuje na koji način se koristi mobilna aplikacija za kupovanje te su dane upute za njeno pravilno korištenje. Prikazat će se nekoliko različitih slučajeva ispitivanja aplikacije kako bi se istražile i provjerile razne njene karakteristike. Najprije će se prikazati uputstva za upotrebu aplikacije, a nakon toga različiti slučajevi ispitivanja aplikacije. Na kraju odjeljka prokomentirat će se dobiveni rezultati.

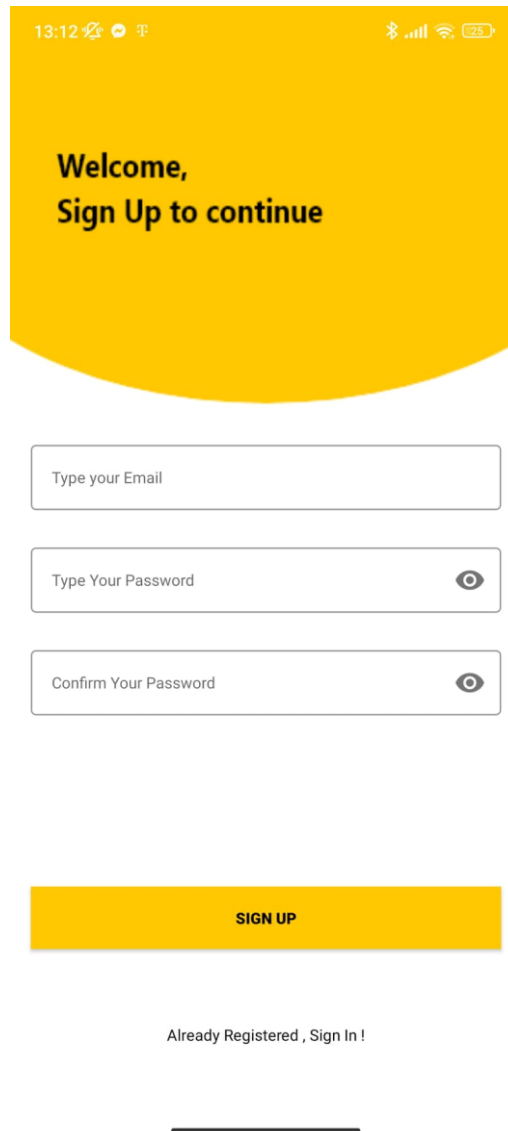
### 5.1. Upute za korištenje aplikacije za kupovinu proizvoda

Prilikom otvaranja aplikacije korisniku se pali zaslon za prijavu koji se sastoji od dva polja gdje treba upisati e-poštu i lozinku. Na slici 5.1. prikazan je zaslon za prijavu korisnika.



Slika 5.1. Zaslon za prijavu korisnika

Ukoliko korisnik nema račun treba stisnuti dio ekrana gdje piše „*Not Registered Yet, Sign Up !*“. Zatim se otvara zaslon za registraciju korisnika gdje korisnik treba unijeti svoju adresu e-pošte te unijeti lozinku dva puta kako bi stvorio račun. Na slici 5.2. prikazan je zaslon za registraciju korisnika.



13:12 100% 25%

Welcome,  
Sign Up to continue

Type your Email

Type Your Password

Confirm Your Password

SIGN UP

Already Registered , Sign In !

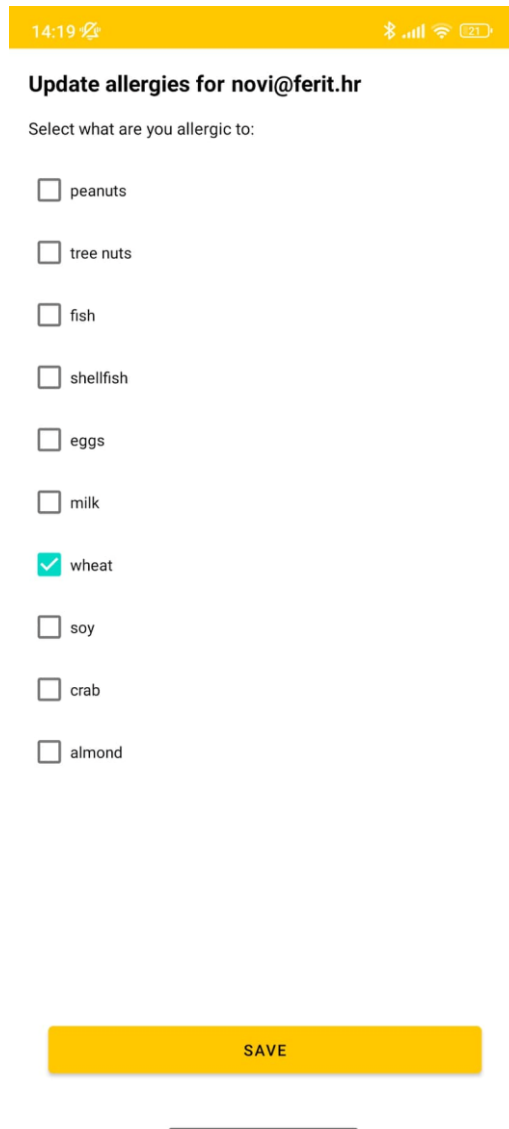
**Slika 5.2.** Zaslon za registraciju korisnika

Nakon uspješne prijave ili registracije korisniku se pali početni zaslon. Ukoliko korisnik nema nikakvih alergija može odmah krenuti kupovati namirnice pritiskom na tipku „scan barcode“. Ukoliko korisnik ima alergije, treba stisnuti tri točkice u gornjem desnom uglu te stisnuti „Update your allergies“ gdje mu se pojavljuje popis checkbox lista svim alergija. Na slici 5.3. prikazan je glavni zaslon aplikacije, a na slici 5.4. prikazan je zaslon gdje korisnik može odabrati alergije.





**Slika 5.3.** Glavni zaslon korisnika nakon prijave



**Slika 5.4.** Zaslona odabira alergija

Na slici 5.5. prikazan je glavni zaslon aplikacije nakon odabira alergena. Na zaslonu je uvijek vidljivo koje alergije ima korisnik.



**Slika 5.5.** Glavni zaslon nakon odabire alergija

Ukoliko korisnik ne treba odabrati alergije ili ako ih je odabrao i vratio se na početni zaslon, može skenirati barkodove proizvoda. Pritiskom na tipku „scan barcode“ pali se kamera uređaja te u označeno polje treba staviti barkod proizvoda. Na slici 5.6. prikazan je zaslon u vrijeme skeniranja proizvoda.



**Slika 5.6.** Zaslón skeniranja barkoda

Nakon prepoznavanja proizvoda izbacuje se poruka koja upozorava ukoliko namirnica sadrži opasne alergene i može se odabrati želi li korisnik ubaciti namirnicu u košaricu. Na slici 5.7. vidljiv je zaslón kada korisnik ima proizvode u košarici.



**Slika 5.7.** Dodani proizvodi u košaricu

Nakon što je završeno skeniranje svih proizvoda može se generirati završni QR kod koji predstavlja završni kod cijelog računa koji se treba skenirati na blagajni u trgovini. Na slici 5.8. prikazan je zaslon s generiranim QR kodom.

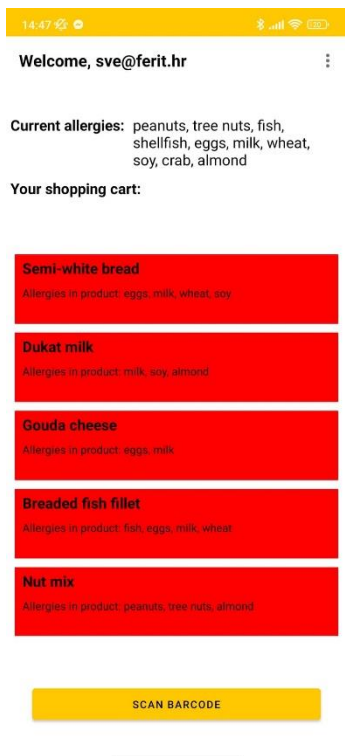


Sucesfully

**Slika 5.8.** Završni kod računa

## 5.2. Uvjeti i korisnički slučajevi za ispitivanje rada aplikacije

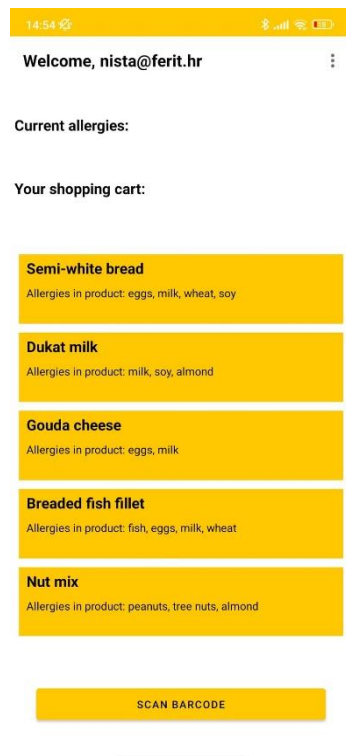
U ovom dijelu aplikacije koristit će se različiti profili korisnika kako bi se vidjelo ponašanje aplikacije. U prvom slučaju korisnik će biti alergičan na sve izabrane alergene, u drugom samo na neke, a u trećem neće biti alergičan ni na što. Na slici 5.9. prikazan je izgled zaslona kada su sve namirnice u košarici opasne za korisnika, na slici 5.10. nalazi se samo jedna opasna namirnica u košarici, a na slici 5.11. sve namirnice su sigurne za korisnika.



**Slika 5.9.** Slučaj kada je korisnik alergičan na sve



**Slika 5.10.** Slučaj kada je korisnik alergičan na neki sastojak



**Slika 5.11.** Slučaj kada korisnik nema alergija

U prvom primjeru korisnik je alergičan na sve ponuđene alergene. Skeniranjem proizvoda iz baze podataka i dodavanje istih u košaricu može se primijetiti kako se crveni svaka namirnica u košarici jer svaka namirnica ima barem neki od alergena što je vidljivo u bazi podataka sa slike 4.15. U drugom slučaju korisnik je alergičan samo na žito. Iz baze podataka, ali i iz popisa alergena koji se nalazi ispod naziva proizvoda u košarici, može se iščitati svi alergeni iz proizvoda. Na slici 5.10. može se vidjeti kako samo „semi-white bread“ ima u sebi alergen za žito i zbog toga je ta namirnica obojana crveno u košarici. U trećem slučaju korisnik nema nikakvih alergena te skeniranjem proizvoda i ubacivanjem u košaricu nema proizvoda koji su obojani crveno nego su svi sigurni za korisnika te su obojani žutom bojom. Također, nakon svake kupnje i generiranja QR koda korisnik uvijek dobije kod koji je moguće skenirati i koji vraća listu skeniranih proizvoda.

### **5.3. Analiza rada aplikacije**

Aplikacija je jednostavna za korištenje, izgled same aplikacije nije kompliciran te su sve funkcionalnosti vidljive i vrlo im je lako pristupiti. Korisnik je nužan napraviti račun kako bi mogao koristiti aplikaciju što mu dozvoljava i korištenje aplikacije na drugom/ više uređaja. Korisnik ne mora dati svoje osobne podatke što većina korisnika cijeni jer te informacije baš i ne vole dijeliti sa svima. Korisnik mora jedino podijeliti svoju adresu e-pošte i alergije te ukoliko korisnik ima neku lažnu adresu e-pošte, nitko ne zna tko je on. Aplikacija radi u svim testiranim slučajevima te uvijek točno izbacuje upozorenje ukoliko se korisnikove alergije poklapaju s alergenima u namirnicama. Također, aplikacija izbacuje i ostale poznate alergene unutar nekog proizvoda kako bi korisnik mogao pripaziti ukoliko kupuje namirnice za neku drugu osobu ili kako ne bi ponudio tu namirnicu nekome tko je alergičan.



## 6. ZAKLJUČAK

Od početaka društvenoga života ljudi pojavila se i trgovina te razmjena stvari između njih. Svake godine ljudi nastoje što više modernizirati i ubrzati proces kupovine. Svakoga dana ljudi obavljaju kupovinu te je od velike važnosti da je taj proces brz i jednostavan za svakog pojedinca. U ovom završnom radu obrađena je tema kreiranja mobilne aplikacije za prilagodljivo kupovanje proizvoda. Aplikacija je napisana u programskom jeziku Kotlin te je popraćena lokalnom bazom podataka Room i Firebase bazom podataka koja omogućuje registraciju i prijavu korisnika, odnosno kupca. Potrebno je bilo kreirati klasu Product koja predstavlja namirnice koje je moguće kupiti. Nakon toga bilo je potrebno definirati objekt klase DAO koja definira metode za upravljanje objektima klase iz baze podataka. Zatim je implementiran skener barkoda koji funkcionira na način da parametre barkoda vraća u obliku stringa pomoću kojeg se dohvaćaju informacije o namirnici.

Nakon testiranja aplikacije, rješenje je zadovoljavajuće. Odgovor na čitanje barkoda i prepoznavanja opasnosti za korisnika je brzo i točno. Skeniranje radi odlično i ako se očitava barkod s zaslona računala. Aplikacija izbacuje crveno upozorenje ukoliko je korisnik ubacio u košaricu namirnicu koja je opasna za njega, a ukoliko je sigurna namirnica ostaje žute boje.

## Literatura

- [1] N. Tudorić, Živjeti s alergijama: O alergijama [online], Pliva zdravlje, 2009, Dostupno na: <https://www.plivazdravlje.hr/tekst/clanak/16160/O-alergijama.html> [Pristupljeno 10 veljače 2021].
- [2] Alan R. Gaby, Integrative Medicine (Fourth Edition) Chapter 31 - Food Allergy and Intolerance, 2018, str. 310-318
- [3] Hrvatski zavod za javno zdravstvo, Alergije – kad preburno reagira imunski sustav [online], 2019, Dostupno na: <https://www.hzjz.hr/sluzba-zdravstvena-ekologija/alergije-kad-preburno-reagira-immunosni-sustav/> [Pristupljeno 10 veljače 2021]
- [4] MSD priručnik dijagnostike i terapije, Alergija na hranu [online], 2018, Dostupno na: <http://www.msd-prirucnici.placebo.hr/msd-prirucnik/immunologija-i-alergije/alergije-i-druge-preosjetljivosti/alergija-na-hranu> [Pristupljeno 10 veljače 2021]
- [5] S. Barnaki i N. Boroša, Bar kodovi i QR kodovi [online] <http://web.studenti.math.pmf.unizg.hr/~sanbarn/> [Pristupljeno 7 rujna 2022]
- [6] PVZG, Što je QR kod i za što ga sve možeš koristiti? [online], 2020, Dostupno na: <https://pvzg.hr/blog/sto-je-qr-kod-i-za-sto-ga-sve-mozes-koristiti/> [Pristupljeno 7 rujna 2022]
- [7] C. Schmidt, What is Android? Here is a complete guide for beginners [online], Tech & Community, 2016, Dostupno na: <https://www.nextpit.com/what-is-android> [Pristupljeno 7 rujna 2022]
- [8] GeeksforGeeks, History of Android [online], 2020, Dostupno na: <https://www.geeksforgeeks.org/history-of-android/> [Pristupljeno 7 rujna 2022]
- [9] L. Ardito, R. Coppola, G. Malnati, M. Torchiano, Effectiveness of Kotlin vs. Java in android app development tasks, Information and Software Technology, 2020. str. 1-3
- [10] Tech Target, Android Studio [online], 2018, Dostupno na: <https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio> [Pristupljeno 8 rujna 2022]
- [11] Google Help, About Firebase and AdMob [online], Dostupno na: <https://support.google.com/admob/answer/6360054?hl=en> [Pristupljeno 8 rujna 2022]
- [12] Web programiranje, Rad sa SQLite bazom u Androidu uz pomoć Room biblioteke [online], Dostupno na: <https://www.webprogramiranje.org/rad-sa-sqlite-bazom-u-androidu-uz-pomoc-room-bibiloteke/> [Pristupljeno 8 rujna 2022]

## **Životopis**

Dino Goluža rođen je 24.2.2000. u Osijeku, a živi i odrasta u Đakovu. Nakon pohađanja Osnovne škole Ivan Goran Kovačić 2014. godine upisuje opću gimnaziju Antun Gustav Matoš. Srednju školu završava 2018.godine te upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer računarstvo. Uz hrvatski jezik, koristi engleski jezik u pismu i govoru. Poznaje nekoliko programskih jezika poput C, C++, C#, Java, Kotlin, HTML, CSS i SCSS. Uz sve studentske obaveze, uspješno se bavi rukometom dugi niz godina.

---

Potpis autora

## SAŽETAK

U ovom završnom radu osmišljena je i isprogramirana Android mobilna aplikacija za kupovinu namirnica zasnovana na preporukama. Napravljena je u programskom okruženju Android Studio gdje je korišten Kotlin programski jezik za funkcionalnost, a za dizajn opisni jezik XML. Aplikacija omogućuje registraciju korisnika kako bi njegove alergije ostale zapamćene. Uz pomoć alergena u proizvodu i korisnikovih alergija aplikacija prepoznaje je li skenirani proizvod potencijalno opasan za zdravlje korisnika te ga o tome obavještava. Po završetku skeniranja svih proizvoda korisnik generira završi QR kod koji predstavlja račun svih skeniranih proizvoda.

**Ključne riječi:** alergeni, alergije, android mobilna aplikacija, barkod, trgovina, QR kod

## **ABSTRACT**

In this Bachelor's thesis I deal with the design and programming of Android mobile application for grocery shopping based on recommendations. It was made in the Android Studio software environment where Kotlin programming language was used for functionality and the descriptive language XML was used for design. The application allows users to register so that their allergies are remembered. With the help of the allergen in the product and the user's allergies, the application recognizes whether the scanned product is potentially dangerous to the user's health and notifies it accordingly. Upon completion of scanning of all products, the user generates a complete QR code representing the account of all scanned products.

**Keywords:** allergens, allergies, android mobile application, barcode, shop, QR code

## **PRILOZI**

Prilog 1. Završni rad u formatu .docx

Prilog 2. Završni rad u formatu .pdf

Prilog 3. Projekt izrađene mobilne aplikacije