MOBILNA ANDROID APLIKACIJA ZA PRAĆENJE KORIŠTENJA PREHRAMBENIH PROIZVODA NA TEMELJU SUSTAVA PREPORUKA I RASPOREĐIVANJA

Bešlić, Kristina

Undergraduate thesis / Završni rad

2021

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:200:276097

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij

MOBILNA ANDROID APLIKACIJA ZA PRAĆENJE KORIŠTENJA PREHRAMBENIH PROIZVODA NA TEMELJU SUSTAVA PREPORUKA I RASPOREĐIVANJA

Završni rad

Kristina Bešlić

Osijek, 2021.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PRAĆENJE KORIŠTENJA PREHRAMBENIH PROIZVODA I STANJE U PODRUČJU 2

2.1. Problem praćenja isteka rokova prehrambenih proizvoda 2

2.2. QR kod 2

2.3. Sustavi preporuka 3

2.4. Postupak raspoređivanja 4

2.5. Prikaz postojećih rješenja 5

2.5.1. KitchenPal 5

2.5.2. Magic Fridge 6

2.5.3. Best Before 7

2.6. Idejno rješenje mobilne aplikacije 8

3. MODEL MOBILNE APLIKACIJE 9

3.1. Očitavanje QR kodova 9

3.2. Pohrana proizvoda u bazu podataka Firebase 10

3.3. Redoslijed korištenja proizvoda 10

3.4. Obavijesti sustava 11

3.5. Sustav preporuka 11

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE 12

4.1. Korišteni alati i tehnologije 12

4.1.1. Android Studio 12

4.1.2. Programski jezik Java 12

4.1.3. XML 12

4.1.4. Dexter 12

4.1.5. ZXing 13

4.1.6. Firebase 14

4.2. Predložak mobilne arhitekture MVP 14

4.3. Programsko rješenje mobilne aplikacije 15

4.3.1. QR čitač 15

4.3.2. Rukovanje podacima....18
4.3.3. Početni zaslon aplikacije....21
4.3.4. Postupak raspoređivanja....22
4.3.5. Obavijesti aplikacije i sustav preporuka....23
5. NAČIN RADA I ISPITIVANJE RADA APLIKACIJE....26
5.1. Način rada mobilne aplikacije....26
5.2. Prikaz rada aplikacije....26
5.2.1. Prikaz početnog zaslona....26
5.2.2. Prikaz čitača QR koda....27
5.2.3. Prikaz zaslona za unos proizvoda....29
5.2.4. Prikaz zaslona s receptima....30
5.3. Ispitivanje rada aplikacije....32
5.3.1. Ispitni slučaj 1 – sučelje nakon instalacije aplikacije....32
5.3.2. Ispitni slučaj 2 – proizvodu je valjan odnosno istekao rok uporabe....33
5.3.3. Ispitni slučaj 3 – proizvodu je istekao rok uporabe i obavijesti aplikacije....35
5.4. Analiza rada aplikacije....35
6. ZAKLJUČAK....37
LITERATURA....38
POPIS SLIKA....40
SAŽETAK....42
ABSTRACT....43
ŽIVOTOPIS....44
PRILOZI....45

1. UVOD

Praćenje korištenja prehrambenih proizvoda od iznimne je važnosti prvenstveno zbog zdravstvenih razloga osobe koja konzumira proizvode. Također, nakon isteka roka, proizvodi gube svježinu i izvorni okus. Cilj ovoga završnog rada bio je olakšati korisniku korištenje proizvoda pred istek roka uporabe te omogućiti lakše praćenje isteka rokova uporabe proizvoda.

Android aplikacija za praćenje korištenja proizvoda korisniku će pojednostaviti praćenje proizvoda koje ima u kućanstvu, a također bit će korisna pri odlasku u kupovinu jer su svi proizvodi sadržani u aplikaciji. Ulazak u aplikaciju omogućit će izlistanje svih proizvoda koje korisnik posjeduje. Također, praćenje proizvoda kojima je rok uporabe istekao bit će jednostavnije tako što će u aplikaciji oni biti crvene boje koja upozorava korisnika da proizvod više nije valjan. Svakodnevne obavijesti aplikacije podizat će svijest korisnika kako postoje proizvodi kojima uskoro istječe rok trajanja. Unos novih proizvoda bit će olakšan jer proizvodi korišteni u ovoj aplikaciji sadrže QR kod koji korisnik može jednostavno očitati kamerom mobilnog uređaja.

U drugom poglavlju teorijski će biti opisane funkcionalnosti aplikacije te će biti navedena i opisana postojeća programska rješenja koja se bave izazovima učinkovitog korištenja prehrambenih proizvoda. Treće poglavlje kroz model aplikacije sadrži opis njenih funkcionalnosti koje će biti programski ostvarene. Četvrto poglavlje prikazuje opis korištenih programskih tehnologija te programsko ostvarenje mobilne aplikacije. Detaljan prikaz načina rada i ispitivanje aplikacije s analizom bit će opisani u petom poglavlju rada.

1.1. Zadatak završnog rada

U teorijskom dijelu završnog rada treba proučiti i opisati izazove i probleme pravovremene nabave, čuvanja i trošenja prehrambenih proizvoda, načine njihovog rješavanja različitim pristupima s naglaskom na korištenje sustava preporuka i postupaka raspoređivanja. Uzimajući u obzir mogućnosti skeniranja QR kodova proizvoda pri kupnji, rokove trajanja i potrošnju, treba predložiti model, arhitekturu i dizajn mobilne aplikacije s bazom podataka. Mobilna aplikacija treba omogućiti učinkovitu nabavu, čuvanje i potrošnju proizvoda na temelju skeniranja QR kodova, ugrađenog sustava preporuka prijedloga načina korištenja proizvoda, kao i redoslijeda korištenja s obavijestima dobivenih korištenjem odgovarajućeg postupka raspoređivanja. U praktičnom dijelu završnog rada, opisanu mobilnu aplikaciju treba programski ostvariti korištenjem prikladnih programskih tehnologija, jezike i razvojne okoline, te je ispitati za odgovarajuće ulazne podatke i načine korištenja.

2. PRAĆENJE KORIŠTENJA PREHRAMBENIH PROIZVODA I STANJE U PODRUČJU

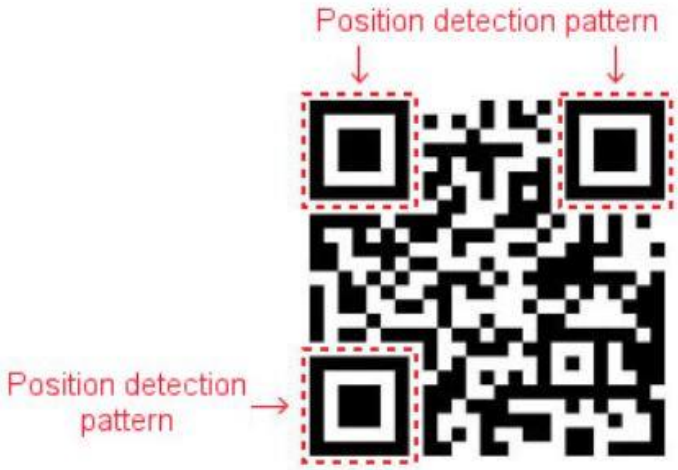
U ovom poglavlju bit će opisan problem praćenja isteka rokova prehrambenih proizvoda, funkcionalnosti koje će imati mobilna aplikacija te će biti prikazana postojeća programska rješenja problema korištenja prehrambenih proizvoda.

2.1. Problem praćenja isteka rokova prehrambenih proizvoda

Prema [1], praćenje isteka roka prehrambenih proizvoda od izuzetne je važnosti jer konzumiranje proizvoda nakon isteka roka trajanja može uzrokovati zdravstvene probleme osobe koja je proizvod konzumirala. Također, nakon isteka roka moguće je smanjenje kvalitete i svježine proizvoda te gubitak izvornog okusa proizvoda. Mobilna aplikacija za praćenje korištenja prehrambenih proizvoda riješila bi problem jer se umjesto korisnika brine o isteku roka uporabe proizvoda.

2.2. QR kod

Prema [2], QR kod (engl. *quick response code*) dvodimenzionalni je barkod koji je izumila japanska tvrtka Denso Wave 1994. godine kako bi pratila proces proizvodnje vozila. Takva vrsta kodova korištena je u različitim poljima poput proizvodnje, zdravstva te prijenosa informacija i dobara. Kako bi se kod mogao što brže očitati, prema [3], osmišljena su tri uzorka za pronalaženje ispravnog položaja koda (engl. *position detection pattern*), kao što je vidljivo na slici 2.1. Kod može sadržavati oko 7000 različitih vrsta znakova. Također, sadrži funkciju za detekciju i ispravljanje pogrešaka koja omogućuje očitavanje šifriranog sadržaja unatoč djelomičnoj oštećenosti ili nedostatku koda. Razvitak pametnih telefona uzrokovao je porast korištenja QR kodova u marketinškim aplikacijama. Pametni telefoni korišteni su kako bi preko kamere uređaja i aplikacije za čitanje QR kodova zabilježili QR kod te dešifrirali tekst šifriran unutar njega kako bi se mogla izvesti odgovarajuća radnja. U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda, QR kod će se koristiti kako bi se učitali podaci o proizvodu.



Slika 2.1. Uzorci za pronalaženje ispravnog položaja koda [3]

2.3. Sustavi preporuka

Prema [4], sustavi preporuka programski su alati i metode za davanje preporuka za koje se smatra da bi bile korisne za neku osobu (korisnika). Preporuke su vezane za različite procese donošenja odluka, na primjer što kupiti, koji film pogledati, koju glazbu slušati ili koje vijesti pročitati. Glavni cilj sustava preporuka je dati točnu preporuku za određenog korisnika te smanjiti pretrpanost informacijama prilagođavanjem njihovog prikaza korisničkom interesu. Individualne preporuke definiraju se na temelju korisnikovih prošlih odabira te njegovog profila. Prema [5], takvi sustavi su zamjena za algoritam pretraživanja jer predlažu korisnicima informacije koje možda ne bi sami pronašli. Sustavi preporuka krenuli su se razvijati 1990-ih kada su se istraživači krenuli baviti problemom preporučivanja koji je ovisio o metodama ocjenjivanja. Prema [6], korišteni su kako bi predvidjeli ocjenu korisnika za pojedinu stavku korištenjem sustava za filtriranje informacija iz liste koja sadrži korisnikove prošle ocjene. Također, ocjenu mogu predvidjeti pronalaženjem sličnih karakteristika s ostalim stavkama ili pronalaženjem zajedničkih interesa korištenjem demografskih podataka s korisnikovog profila. Neke od stranica koje koriste sustave preporuka su Netflix, Amazon i Yahoo.

Postoje različite vrste filtriranja, a neke od njih su temeljene na sadržaju, temeljene na zajedničkim interesima i hibridne vrste. Preporuke temeljene na sadržaju oslanjaju se na opis stavki te korisničke ukuse o kojima se informacije dobivaju preko praćenja njihovog profila. Također, preporučuju se stavke slične onima koje je korisnik u prošlosti pozitivno ocijenio. Preporuke temeljene na zajedničkim interesima kreiraju se tako da se prikupljaju i analiziraju informacije o

ponašanju korisnika u smislu njegovih povratnih informacija, ocjena i aktivnosti. Nakon analize tih informacija stvaraju se preporuke temeljene na sličnostima između različitih korisnika koji imaju slične interese. Također, preporuke temeljene na demografskim obilježjima funkcioniraju na sličan način, samo što se osim ocjena korisnika kao informacije za filtriranje uzimaju i dob, spol, demografsko područje, obrazovanje i interesi. Hibridni pristup je spoj preporuka temeljenih na sadržaju i zajedničkim interesima. Takav pristup ima više metoda od kojih su neke istovremene preporuke različitih stavki, zatim sustavi koji imaju više preporuka te korisniku prikazuju onu koja bi mu najbolje odgovarala na temelju njegovih interesa i na kraju, liste preporuka koje na vrhu imaju preporuke s najvišim ocjenama te na dnu one koje su najniže ocijenjene.

Sve vrste sustava imaju svoje prednosti i nedostatke, a neke od njih bit će iznesene u nastavku. Prednosti sustava temeljenih na sadržaju su poštivanje korisnikove privatnosti te mogućnost predlaganja novih stavki koje bi se korisniku mogle svidjeti. Predlaganje je temeljeno na sličnostima između karakteristika pojedinih stavki. Nedostaci takvih sustava su što preporuke ne ovise o ocjenama korisnika pa samim time nije uključena ocjena kvalitete nekog proizvoda. Također, da bi se kreirala ispravna lista preporuka potrebno je analizirati sve značajke pojedinih stavki. Prednosti sustava temeljenih na zajedničkim interesima su da ne koriste informacije o demografskim obilježjima, ali sustav pronalazi slične stavke između korisnika te kreira preporuke prema tome. Također, mogu predložiti korisniku stavke koje nisu unutar njegovih interesa, ali bi mu se možda svidjele. Nedostaci su problem preporuke stavki novom korisniku (engl. *cold start problem*) te kvaliteta sustava ovisi o stavci u listi koja ima najveću ocjenu. Prednosti filtriranja na temelju demografskih obilježja su što se preporuke mogu formirati i prije nego što je korisnik ocijenio neku stavku, ali nedostatak toga je problem privatnosti korisnika. Hibridni pristup kombinira sve prednosti sustava temeljenih na filtriranju sadržaja i zajedničkih interesa te time povećava korisničko zadovoljstvo, ali se javlja problem preporuke stavki novom korisniku. U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda, sustav preporuka bit će korišten pri prijedlogu načina iskorištavanja proizvoda kroz različite recepte.

2.4. Postupak raspoređivanja

Prema [7], postupak raspoređivanja podrazumijeva uređivanje i vremensko pridjeljivanje zadataka. Ovisno o problemu, raspoređivanje se može podijeliti na statičko, dinamičko i mješovito. Statičko raspoređivanje se koristi kada su rasporedi stalni, odnosno nepromjenjivi zbog stalnih prioriteta pridijeljenih zadacima za vrijeme trajanja aplikacije. Svaka promjena ili novi zahtjev nalaže raspoređivaču potpuno preraspoređivanje zadataka, stoga postupci statičkog raspoređivanja

ne mogu savladati nepredviđene pojave ili odstupanja. Unatoč nefleksibilnosti, statičko raspoređivanje povoljno je koristiti zbog niskih troškova. Algoritmi za dinamičko raspoređivanje predviđeni su za rad sa zadacima s nepredviđenim vremenom dolaska te trenutno reagiraju na novonastalu situaciju. Mješovito raspoređivanje radi po principu da zadatke s unaprijed poznatim vremenskim značajkama raspoređuje statički i tome prilagođava druge prilikom njihova dolaska. Prema [8], zadatak je logički slijed pri izvršavanju nekog posla. Zadatak može biti određen prema zahtjevima o završetku, prekidivosti i dolasku. Zahtjevi o završetku predstavljaju krajnji rok do kojega zadatak treba biti izvršen, dok prekidivost znači kako svaki zadatak može biti prekinut u izvršavanju po dolasku novog zadatka. Prema dolasku, postoje dvije vrste zadataka, periodni i aperiodni. Periodni zadaci mogu biti raspoređeni prema algoritmu sa stalnim prioritetima zadataka (engl. *rate monotonic scheduling, RMS*) ili prema algoritmu za obavljanje zadataka s najranijim završetkom (engl. *earliest deadline first, EDF*). Aperiodni zadaci imaju proizvoljno vrijeme dolaska i najbolji algoritam za njihovo rješavanje je onaj s minimalnim vremenom odgovora na zadatak, minimalnim brojem zadataka s koji nisu izvršeni u roku te s maksimalnim korištenjem resursa u danoj radnoj okolini. Periodni zadaci moraju biti izvršeni prije krajnjeg roka izvršavanja zadatka, dok aperiodni zadaci moraju poštovati krajnji rok izvršavanja zadatka, ali po mogućnosti sa što kraćim vremenom odgovora na zadatak. U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda koristit će se mješovito raspoređivanje zadataka. Također, koristit će se prioritetno svrstavanje zadataka koje će imati svrhu pri svrstavanju proizvoda prema datumima isteka roka uporabe sortirajući ih uzlaznim redoslijedom.

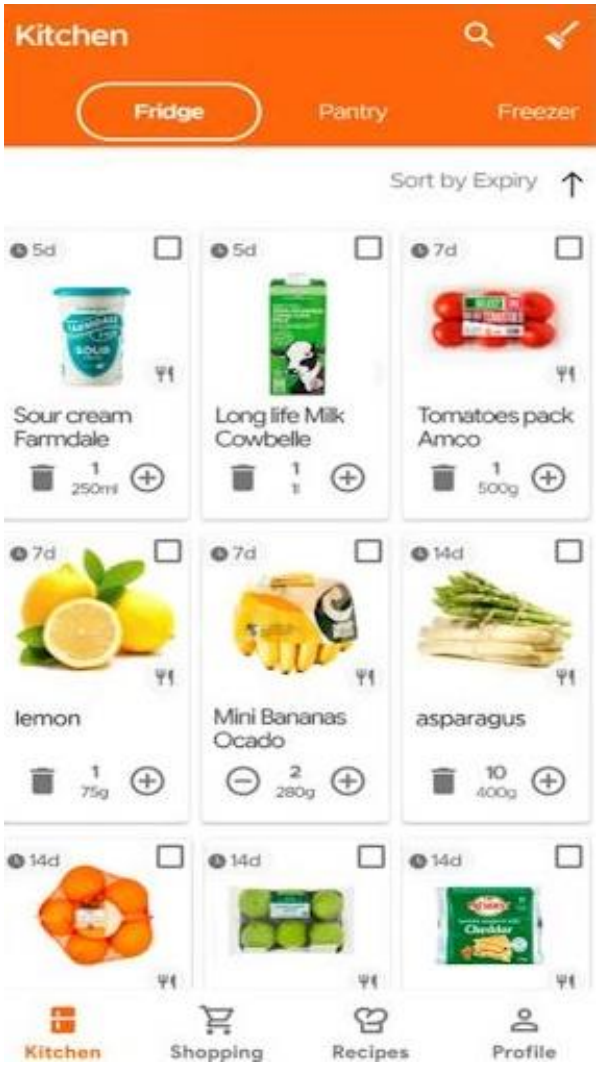
2.5. Prikaz postojećih rješenja

Na tržištu postoji velik broj aplikacija koje omogućuju praćenje isteka roka prehrambenih proizvoda te daju prijedloge načina njihova iskorištavanja. U nastavku će biti opisana tri programska rješenja za rješavanje problema praćenja i iskorištavanja prehrambenih proizvoda.

2.5.1. KitchenPal

KitchenPal je, prema [9], aplikacija koja pruža mogućnosti praćenja proizvoda koje korisnik trenutačno posjeduje, usporedbu tih proizvoda te davanje recepata u kojima se proizvodi mogu iskoristiti. Nadalje, korisnik može organizirati proizvode prema vrsti te postaviti datume isteka roka proizvoda te za iste primati upozorenja kada im je rok uporabe pri kraju. Aplikacija omogućuje i skeniranje omiljenih proizvoda te dijeljenje aplikacije s ukućanima kako bi više članova moglo dodavati nove proizvode i pregledavati postojeće. Također, pruža mogućnost kreiranja liste za kupovinu te vlastitih recepata. Aplikacija omogućava održavanje korisnikova

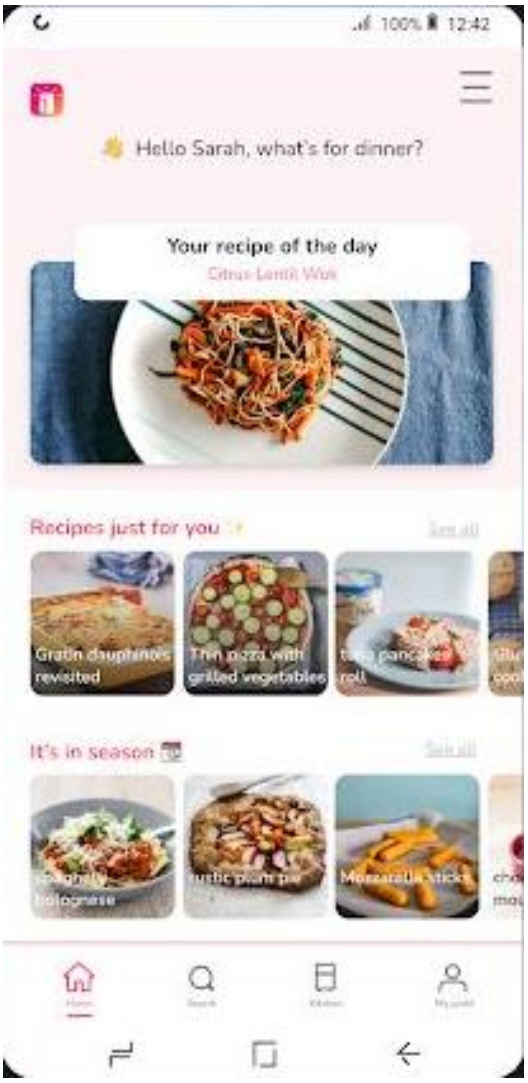
zdravlja i dijete. Također, korisnici mogu skenirati i uspoređivati nutritivne vrijednosti između različitih proizvoda i proizvođača. Slika 2.2 prikazuje korisničko sučelje aplikacije KitchenPal.



Slika 2.2. Korisničko sučelje aplikacije KitchenPal

2.5.2. Magic Fridge

Magic Fridge, prema [10], je aplikacija koja omogućava odabir prehrambenih proizvoda te pristup receptima koji odgovaraju korisniku. Također, pruža mogućnost prilagodbe recepata na temelju sastojaka koje korisnik trenutno ima. Slika 2.3 prikazuje korisničko sučelje aplikacije Magic Fridge.



Slika 2.3. Korisničko sučelje aplikacije Magic Fridge

2.5.3. Best Before

Best Before je, prema [11], aplikacija koja pruža mogućnost skeniranja bar kodova proizvoda kako bi se popunile informacije i slika proizvoda te datum isteka roka. Korisniku šalje podsjetnike o istjecanju roka proizvoda. Aplikacija pruža mogućnost kreiranja listi za kupnju. Slika 2.4 prikazuje korisničko sučelje aplikacije Best Before.

Slika 2.4. Korisničko sučelje aplikacije Best Before

2.6. Idejno rješenje mobilne aplikacije

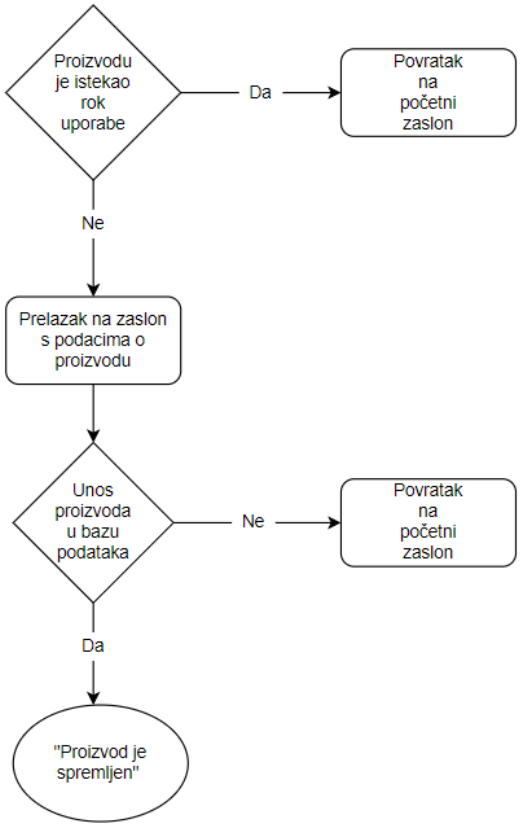
Mobilna aplikacija po uzoru na prethodno spomenute aplikacije treba biti korisna za praćenje isteka rokova uporabe prehrambenih proizvoda. Preko QR koda korisniku će omogućavati dodavanje informacija o proizvodu. Kroz prijedloge recepata informirat će korisnika na koji način proizvode može iskoristiti te će kroz obavijesti upozoravati korisnika o isteku roka uporabe proizvoda. Cilj aplikacije je poboljšati načine iskorištavanja prehrambenih proizvoda te smanjiti ili, ukoliko je moguće, bacanje prehrambenih proizvoda pretvoriti u prošlost. Model aplikacije te njene funkcionalnosti bit će detaljnije opisane u trećem poglavlju.

3. MODEL MOBILNE APLIKACIJE

U ovom poglavlju opisan je model mobilne Android aplikacije za korištenje prehrambenih proizvoda i načini ostvarenja potrebnih funkcionalnosti.

3.1. Očitavanje QR kodova

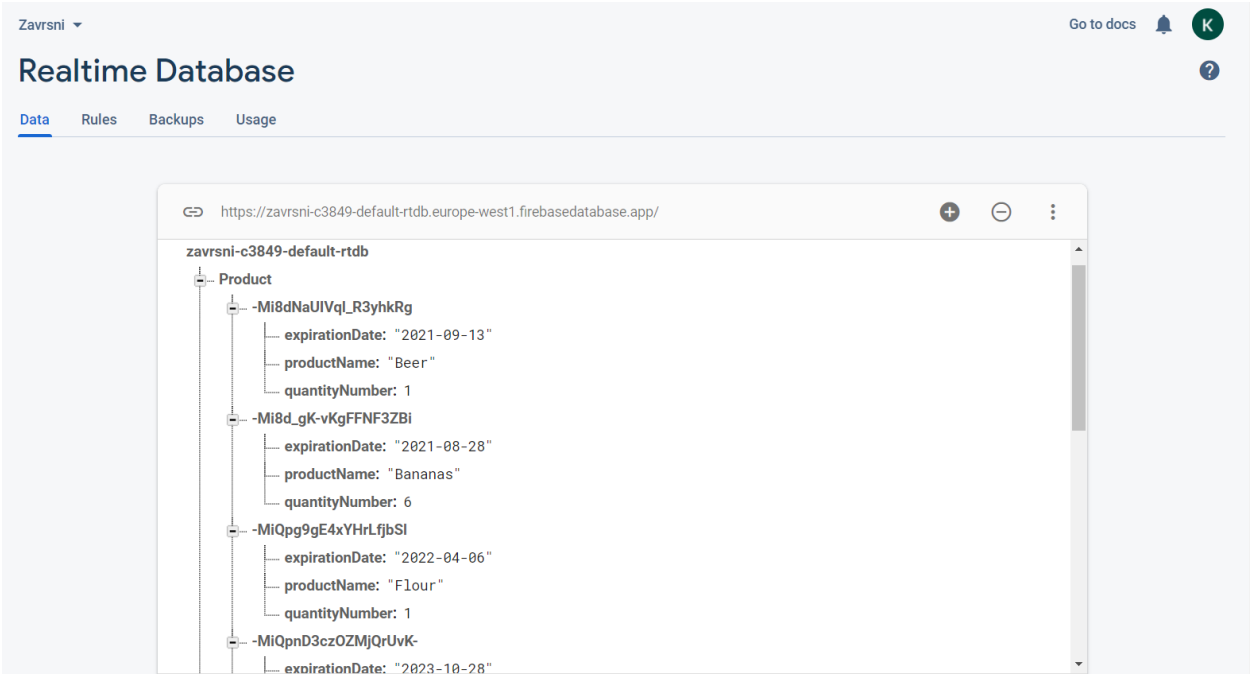
U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda kao jedna od funkcionalnosti bit će ostvaren QR čitač jer će se korisniku pružiti mogućnost očitavanja QR kodova. QR kod prilagođen za korištenje unutar aplikacije sadržavat će naziv proizvoda te datum isteka roka uporabe proizvoda. Nakon očitavanja QR koda, ukoliko proizvodu nije istekao rok uporabe, korisnik će biti usmjeren na zaslon s podacima o proizvodu kojeg se želi unijeti u bazu podataka Firebase. Zaslon s podacima o proizvodu sadržavat će naziv proizvoda i datum isteka roka uporabe koji će se automatski postaviti iz QR koda. Također, sadržavat će i količinu proizvoda koju će korisnik sam podešavati. Nakon što korisnik podesi količinu na željeni broj, pružit će mu se mogućnost unosa proizvoda u bazu podataka ili povratak na početni zaslon. Sve navedeno vidljivo je na dijagramu tijeka na slici 3.1.



Slika 3.1. Tijek aktivnosti nakon očitavanja QR koda

3.2. Pohrana proizvoda u bazu podataka Firebase

Proizvodi se u bazu podataka Firebase unose s podacima o nazivu, količini i datumu isteka roka uporabe. Slika 3.2 prikazuje kako je naziv proizvoda spremljen u varijablu *productName*, količina je spremljena u *quantityNumber*, dok je datum isteka roka spremljen u *expirationDate*. Podatke o proizvodu korisnik će moći popuniti preko očitanog QR koda ili će ih moći unijeti ručno. Pritiskom na gumb za spremanje proizvoda, proizvod je unesen u bazu podataka.



Slika 3.2. Prikaz proizvoda u bazi podataka Firebase

3.3. Redoslijed korištenja proizvoda

Sustav raspoređivanja proizvoda bit će korišten u dijelu aplikacije gdje će biti smješteni proizvodi kojima će uskoro isteći rok uporabe. Na vrhu liste bit će proizvodi kojima najranije istječe rok uporabe što se prema teorijskom opisu sustava raspoređivanja odnosi na mješovito raspoređivanje s EDF tehnikom raspoređivanja, što je algoritam za raspoređivanje zadataka s najranijim krajnjim završetkom zadatka. Ako je nekom proizvodu istekao rok, datum isteka roka uporabe bit će označen crvenom bojom kako bi korisnik bio upozoren da je proizvodu istekao rok uporabe. Proizvod kojemu je istekao rok uporabe neće biti automatski uklonjen iz baze podataka kako bi korisnik bio upozoren da postoji prehrambeni proizvod kojemu je rok istekao te kako bi ga trebao ukloniti iz kućanstva. Nakon što je proizvod potrošen ili uklonjen iz kućanstva ukoliko mu je rok istekao, korisnik će imati mogućnost izbrisati taj proizvod iz svoje aplikacije, a samim tim i iz

baze podataka. Također, postojalo je još načina na koje se mogao ostvariti sustav raspoređivanja. Na primjer, u sustavu preporuka mogli su prvo biti recepti koji sadrže proizvode koji imaju najraniji istek roka uporabe, zatim recepti koji sadrže ostale proizvode iz baze podataka. Koristila bi se također EDF tehnika mješovitog raspoređivanja zadataka.

3.4. Obavijesti sustava

Ukoliko postoje proizvodi kojima uskoro istječe rok uporabe, obavijesti aplikacije bit će aktivirane. Obavijesti aplikacije će se prikazivati svaki dan u određeno vrijeme, sve dok bude proizvoda kojima uskoro istječe rok uporabe. U slučaju kada neće postojati proizvodi kojima je rok uporabe pri kraju, obavijesti aplikacije neće dolaziti.

3.5. Sustav preporuka

U mobilnoj Android aplikaciji, sustav preporuka bit će korišten kako bi korisniku na temelju proizvoda u bazi podataka predlagao recepte u kojima ih on može iskoristiti. Prema teorijskom opisu sustava preporuka, sustav preporuka koji će biti korišten u mobilnoj aplikaciji za korištenje prehrambenih proizvoda bit će sustav temeljen na hibridnom filtriranju informacija. Preporuke će biti ostvarene na način da se na vrhu liste prikazuju recepti koji sadrže proizvode kojima uskoro istječe rok uporabe, dok će prema dnu biti ostali recepti unutar kojih se nalaze proizvodi iz baze podatka. To je ujedno i jedna od metoda hibridnog sustava preporuka, samo se u teoriji ona temelji na ocjenama korisnika, dok će u mobilnoj aplikaciji biti temeljena na sastojcima koje pojedini recept sadržava i njihovom datumu isteka roka uporabe. Kao što je spomenuto u teorijskom dijelu, na početku korištenja aplikacije javit će se problem preporučivanja recepata novom korisniku jer baza podataka neće sadržavati proizvode, stoga se ni recepti neće moći formirati dok se ne unesu proizvodi. Također, svi proizvodi iz baze podataka morali bi se u potpunosti preklapati sa sastojcima u receptima, što u nekim slučajevima neće funkcionirati jer bi korisnik morao unijeti sve proizvode koje ima u kućanstvu, što je teško za očekivati za teško kvarljive proizvode poput soli, šećera i tako dalje. Pritiskom na obavijest koju će prikazivati mobilna aplikacija za korištenje prehrambenih proizvoda, korisnik će biti usmjeren na recepte u kojima će moći iskoristiti proizvode koji uskoro ističu, ali i ostale proizvode koje posjeduje. Recepti će sadržavati sliku, naziv te sastojke koji su potrebni za njihovu izradu te korake izrade određenog jela.

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE

U ovom poglavlju opisano je programsko rješenje mobilne Android aplikacije, korištene programske tehnologije i implementiranje traženih funkcionalnosti.

4.1. Korišteni alati i tehnologije

4.1.1. Android Studio

Android Studio [12], koristi se kao službena okolina za razvoj mobilnih aplikacija. Nudi razne značajke za poboljšanje učinkovitosti pri izradi aplikacija poput brzog emulatora, unificiranog okruženja koje pruža mogućnost za razvoj aplikacija za sve Android uređaje. Također, promjene na aplikaciji moguće su bez ponovnog pokretanja. Nadalje, nudi predloške kodova i integrirani GitHub sustav te podršku za C++. Android Studio podržava korištenje programskih jezika Java i Kotlin.

4.1.2. Programski jezik Java

Java [13], je objektno orijentirani programski jezik razvijen 1995. godine u tvrtki Sun Microsystems. Razvio ga je James Gosling i drugi inženjeri te tvrtke. Sintaksa Jave slična je C i C++ programskim jezicima. Java virtualni stroj (engl. *java virtual machine*) interpretira i kompajlira Java bytecode i tim omogućuje izvođenje programa pisanih u Javi na svim operacijskim sustavima koji sadrže JVM bez potrebe za izmjenom koda. Java je jedan od najkorištenijih programskih jezika. Prvotno ime jezika bilo je Oak, zatim Green i konačno je preimenovan u Java.

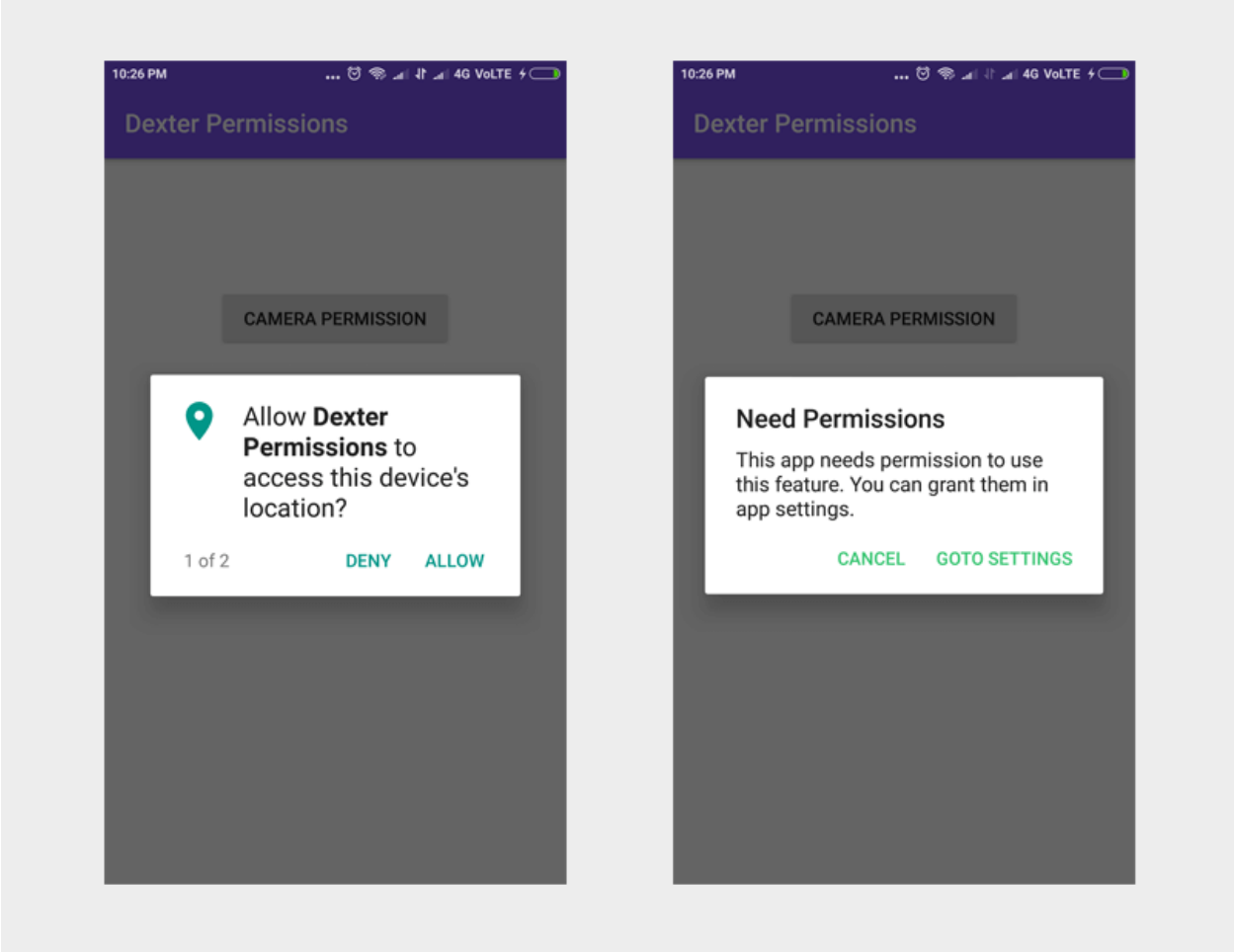
4.1.3. XML

XML (engl. *extensible markup language*) je, prema [14], jezik za označavanje podataka koji propisuje pravila za šifriranje podataka u format razumljiv i računalima i ljudima. Takav način omogućava korištenje oznaka unutar kojih je tekst koji može biti pročitan i razumljiv bez potrebe za posebnim čitačem ili interpreterom. XML se koristi za stvaranje korisničkog sučelja.

4.1.4. Dexter

Dexter [15], je Android biblioteka koja pojednostavljuje proces dodavanja dopuštenja u aplikaciju te rukovanje istima. Također, sadrži funkcionalnost koja omogućava korisnicima da daju ili ne daju pojedina dopuštenja unutar same aplikacije, a ne sva dopuštenja pri instalaciji iste. Takva mogućnost pruža korisniku veću kontrolu unutar aplikacije, ali od programera zahtijeva pisanje veće količine koda kako bi ju ostvario, što je s Dexter bibliotekom značajno pojednostavljeno.

Slika 4.1 prikazuje primjer traženja dopuštenja od korisnika ostvareno s Dexter bibliotekom. U slučaju koji je prikazan na slici, korisnika se traži dopuštenje za pristup lokaciji mobilnog uređaja i još jedno dopuštenje koje nije vidljivo na slici 4.1. U mobilnoj aplikaciji za korištenje prehrambenih proizvoda, dopuštenje će od korisnika biti traženo za pristup kameri mobilnog uređaja.



Slika 4.1. Primjer traženja dopuštenja korisnika [15]

4.1.5. ZXing

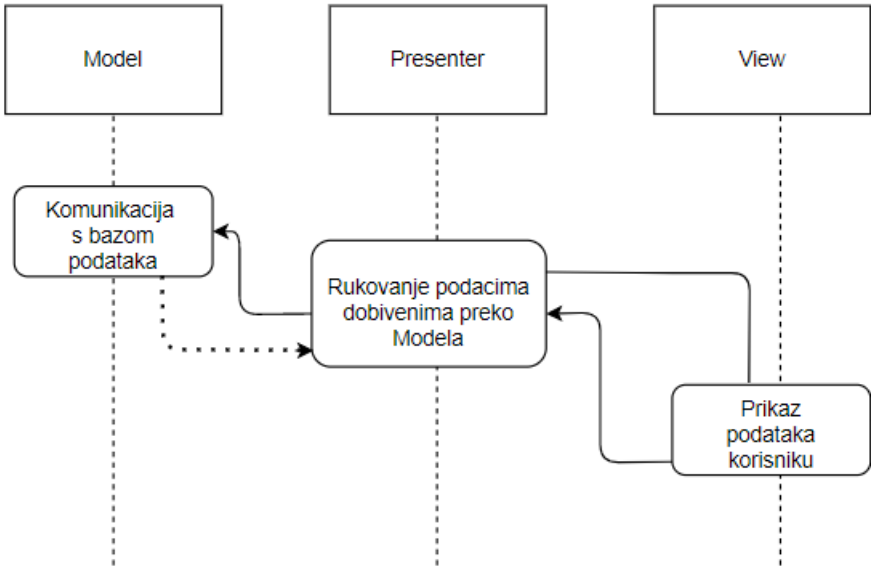
ZXing (engl. *zebra crossing*) je, prema [16], biblioteka otvorenog koda (engl. *open-source*) za obradu više-formatnih 1D/2D barkodova implementirana u Javi. Također, ima mogućnost implementacije u drugim jezicima. Ujedno je i glavna biblioteka koja pruža podršku za razvoj QR kodova u Javi.

4.1.6. Firebase

Firebase [17], je platforma razvijena 2011. godine od strane Firebase, Inc. te se koristi za kreiranje web i mobilnih aplikacija. 2014. godine kupljena je od strane Google-a. Firebase-ov prvi proizvod bio je Firebase Realtime Database, baza podataka koja će se ujedno koristiti u mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda. Baza podataka dodaje, briše i mijenja podatke u stvarnome vremenu. Firebase se također koristi i za poboljšanje kvalitete aplikacije kroz poboljšanje grafike, analiziranje rada aplikacije te testiranje iste.

4.2. Predložak mobilne arhitekture MVP

MVP, prema [18], predstavlja arhitekturu *Model-View-Presenter*. Prezenter (engl. *Presenter*) je posrednik između modela (engl. *Model*) i pogleda (engl. *View*) te je sva logika prikaza sadržaja obavljena s njegove strane. On je odgovoran za dohvaćanje podataka iz modela te ispravan prikaz istih u pogledu. Model brine o podacima koji će kasnije biti prikazani u pogledu u smislu dohvaćanja i rukovanja podacima. Pogled služi samo kao pružatelj podataka korisniku. U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda, model će predstavljati interakciju s bazom podataka, prezenter će biti odgovoran logiku prikaza podataka koje će pogled prikazivati u RecyclerView-u. Na slici 4.2 prikazana je arhitektura MVP koja je napravljena po uzoru na [18].



Slika 4.2. Predložak programske arhitekture MVP [18]

4.3. Programsko rješenje mobilne aplikacije

4.3.1. QR čitač

Na glavnom zaslonu aplikacije nalazi se gumb za unos novih proizvoda. Pritiskom na taj gumb, korisnik je preusmjeren na aktivnost (engl. *activity*) koji sadrži QR čitač te gumb ukoliko korisnik želi sve podatke unijeti ručno. QR čitač ostvaren je preko već spomenute ZXing biblioteke. Za korištenje ZXing biblioteke, bilo ju je potrebno ugraditi u *build.gradle* unutar *dependencies* (Slika 4.3). Također, na slici je prikazana i Dexter biblioteka. Kao što je već spomenuto, ona služi za lakše rukovanje dopuštenjima unutar aplikacije.

```
implementation 'me.dm7.barcodescanner:zxing:1.9.13'
implementation 'com.karumi:dexter:5.0.0'
```

Slika 4.3. Biblioteke za ostvarivanje QR koda i dopuštenja za kameru

Slika 4.4 prikazuje korištenje Dexter biblioteke koja brine o dopuštenjima vezanima za korištenje kamere mobilnog uređaja korisnika aplikacije. Ukoliko je korisnik dopustio korištenje kamere, kao što je vidljivo iz prikaza koda, ugrađena metoda biblioteke pokreće kameru.

```
Dexter.withActivity(this)
        .withPermission(Manifest.permission.CAMERA)
        .withListener(new PermissionListener() {
            @Override
            public void onPermissionGranted(PermissionGrantedResponse response) {
                scannerView.setResultHandler(insertProductDataActivity.this);
                scannerView.startCamera();
            }
```

Slika 4.4. Primjena Dexter biblioteke

Rukovanje rezultatom nalazi se unutar metode *handleResult()* koja kao argument prima rezultat očitavanja QR koda, što je vidljivo na slici 4.5. Rezultat očitavanja QR koda spremljen je u varijablu tipa String, a dohvaća se preko metode *getText()*. Radi lakšeg rukovanja rezultatom, spremljen je unutar polja tipa String, a svaki poseban dio polja dobiven je preko metode *split()* koja kao argument prima izraz prema kojemu je rezultat podijeljen. Kao izraz uzet je „%%“ te su odgovarajući dijelovi stringa spremljeni u polje čiji su elementi spremljeni u varijable odgovarajućeg naziva. Vrijednosti varijabli su nakon provjera valjanosti poslane u aktivnost s

informacijama o proizvodu. Kako bi kasnije provjere davale valjane rezultate, bilo je potrebno provjeriti je li dan i mjesec manji od 10 te ukoliko je, dodati 0 ispred, kao što je vidljivo iz koda na slici 4.5.

```
@Override
public void handleResult(Result result) {
    String scanResult  = result.getText();

    String parts[]     = scanResult.split( regex: "%%");
    String productName = parts[0];
    String day         = parts[1];
    String month       = parts[2];
    String year        = parts[3];

    if(Integer.parseInt(day)   < 10){
        day   = "0" + day;
    }

    if(Integer.parseInt(month) < 10){
        month = "0" + month;
    }
```

Slika 4.5. Rukovanje rezultatom QR čitača

Zbog provjere je li rok uporabe valjan ili je istekao, prvo je dohvaćena instanca klase Calendar, preko nje su dohvaćeni trenutna godina, mjesec i dan te su spremljeni u odgovarajuće varijable. Za ispravan prikaz mjeseca, trenutni mjesec je inkrementiran za 1 jer varijabla *MONTH* u klasi Calendar kreće od 0. Nakon toga slijedi niz provjera valjanosti roka uporabe koji se nalazi unutar QR koda. Ukoliko je datum isteka roka uporabe valjan, preko metode *putExtra()* podaci su poslani u aktivnost koja sadrži informacije o proizvodu (Slika 4.6). Ukoliko datum nije valjan, prikazan je Snackbar koji korisnika obavještava kako je rok istekao te mu omogućava povratak na početni zaslon, kao što je vidljivo na slici 4.7.

```
Calendar calendar = Calendar.getInstance();
int currentDay    = calendar.get(Calendar.DATE);
int currentMonth  = calendar.get(Calendar.MONTH) + 1;
int currentYear   = calendar.get(Calendar.YEAR);

if(Integer.parseInt(year) == currentYear && Integer.parseInt(month) == currentMonth && Integer.parseInt(day) < currentDay){
    showSnackbar();
}else if(Integer.parseInt(year) == currentYear && Integer.parseInt(month) < currentMonth){
    showSnackbar();
}else if(Integer.parseInt(year) < currentYear){
    showSnackbar();
}else{
    Intent intent = new Intent( packageContext: insertProductDataActivity.this, insertProductActivity.class);
    intent.putExtra( name: "productName", productName);
    intent.putExtra( name: "day"        , day);
    intent.putExtra( name: "month"      , month);
    intent.putExtra( name: "year"       , year);
    startActivity(intent);
}
```

Slika 4.6. Provjera valjanosti roka uporabe proizvoda

```
private void showSnackbar(){
    Snackbar snackbar = Snackbar.make(insertWithBtn, text: "Product expired.", Snackbar.LENGTH_INDEFINITE)
            .setAction( text: "OK", new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent( packageContext: insertProductDataActivity.this, MainActivity.class);
                    startActivity(intent);
                }
            });
    snackbar.show();
}
```

Slika 4.7. Prikaz obavijesti o isteku roka

Ukoliko se korisnik odluči za ručni unos svih podataka, pritiskom na gumb biva preusmjeren na aktivnost s poljima za unos podataka, gdje su sva polja prazna, a datum postavljen na trenutni datum te se ne može unijeti datum koji je u prošlosti. Osluškivanje pritiska gumba ostvaren je pozivanjem metode *setOnClickListener()*. Kako se gumb nalazi u istoj aktivnosti kao i QR čitač, aktivnost s podacima o proizvodu očekuje rezultat te je iz tog razloga poslan prazan string za ime proizvoda i trenutni datum, što je vidljivo na slici 4.8.

```
insertProductDataBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar calendar = Calendar.getInstance();
        int day           = calendar.get(Calendar.DATE);
        int month         = calendar.get(Calendar.MONTH);
        int year          = calendar.get(Calendar.YEAR);

        Intent intent = new Intent( packageContext: insertProductDataActivity.this, insertProductActivity.class);
        intent.putExtra( name: "productName", value: "");
        intent.putExtra( name: "day"  , Integer.toString(day));
        intent.putExtra( name: "month", Integer.toString(month));
        intent.putExtra( name: "year" , Integer.toString(year));
        startActivity(intent);
    }
});
```

Slika 4.8. Extras ukoliko korisnik odluči unijeti podatke o proizvodu ručno

Slika 4.9 prikazuje metodu koja osigurava da korisnik ne može postaviti datum u prošlosti.

```
private void setMinDate() { datePicker.setMinDate(System.currentTimeMillis() - 1000); }
```

Slika 4.9. Trenutni datum omogućen kao minimalni

4.3.2. Rukovanje podacima

Za dohvaćanje podataka iz aktivnosti koja sadrži QR čitač, u klasi „insertProductActivity“ korištena je metoda *getIntent()* koja vraća objekt klase Intent te je na njemu pozvana metoda *getExtras()* koja dohvaća sve podatke poslane iz neke druge aktivnosti te ih sprema u objekt klase Bundle. Zbog spremanja podataka u odgovarajuće varijable, na objektu klase Bundle pozvane su odgovarajuće metode *get()* s argumentom ključa u kojega su spremljeni podaci iz aktivnosti iz koje se dohvaćaju. Slika 4.10 prikazuje dohvaćanje podataka te njihovo postavljanje u odgovarajuća polja.

```
private void setData(){
    Bundle bundle = getIntent().getExtras();

    String name = bundle.getString( key: "productName");
    productName.setText(name);

    Calendar calendar = Calendar.getInstance();
    String day   = bundle.getString( key: "day");
    String month = bundle.getString( key: "month");
    String year  = bundle.getString( key: "year");
    calendar.set(Calendar.YEAR , Integer.parseInt(year));
    calendar.set(Calendar.MONTH, Integer.parseInt(month));
    calendar.set(Calendar.DATE , Integer.parseInt(day));
    datePicker.updateDate(Integer.parseInt(year),  month: Integer.parseInt(month) - 1, Integer.parseInt(day));
}
```

Slika 4.10. Dohvaćanje i postavljanje podataka u odgovarajuća polja

Kako bi podaci bili spremljeni u bazu podataka, bilo je potrebno napraviti klasu koja sadrži atribute koji predstavljaju podatke koje će baza podataka sadržavati te metode za dohvaćanje i postavljanje istih. Kod predloška mobilne arhitekture taj dio predstavlja model. U metodi napravljenoj za spremanje podataka u bazu podataka, bilo je potrebno napraviti objekt te klase. Pritiskom na gumb „Add product“ podaci su poslani u bazu podataka. Također, nakon pritiska gumba, prikazana je poruka koja korisniku govori kako su podaci spremljeni te su sve vrijednosti postavljene na početne. Vrijednosti poslane u bazu podataka su ime proizvoda, datum isteka roka te količina. Sve navedeno je vidljivo u kodu na slici 4.11.

```
addProductBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int number = Integer.parseInt(quantity.getText().toString().trim());
        int day   = datePicker.getDayOfMonth();
        int month = datePicker.getMonth() + 1;
        int year  = datePicker.getYear();

        String fm = "" + month;
        String fd = "" + day;
        if(month < 10){
            fm = "0" + month;
        }
        if(day < 10){
            fd = "0" + day;
        }

        product.setProductName(productName.getText().toString().trim());
        product.setQuantityNumber(number);
        product.setExpirationDate("" + year + "-" + fm + "-" + fd);
        reference.push().setValue(product);

        Toast.makeText(getApplicationContext(), text: "Product saved.", Toast.LENGTH_SHORT).show();

        productName.setText("");
        quantity.setText("1");

        Calendar calendar = Calendar.getInstance();
        int YEAR  = calendar.get(Calendar.YEAR);
        int MONTH = calendar.get(Calendar.MONTH);
        int DAY   = calendar.get(Calendar.DATE);
        datePicker.updateDate(YEAR, MONTH, DAY);
    }
});
```

Slika 4.11. Spremanje podataka u bazu podataka te postavljanje vrijednosti polja na početne

Također, na zaslonu postoje dvije ikonice za povećavanje i smanjivanje količine proizvoda. One su tipa ImageView te su na njima pozvane metode za osluškivanje na klik. Ukoliko se korisnik odluči na pritisak ikonice za povećavanje količine proizvoda, metoda za osluškivanje klika pozvat će se na „increaseQuantity“ te će povećati količinu proizvoda na korisnikovu željenu vrijednost. U slučaju kada korisnik želi smanjiti količinu proizvoda, postavljen je dodatan uvjet koji provjerava je li količina proizvoda veća od 1. Ukoliko je, korisniku je dopušteno smanjivati količinu, no u slučaju da je količina jednaka 1, korisnik ju ne može smanjiti, kao što je vidljivo u kodu na slici 4.12. Količina proizvoda se povećava i smanjuje unutar EditText-a te se korisniku također pruža mogućnost da ne koristi ikonice za povećavanje i smanjivanje količine nego da sam upiše željenu vrijednost.

```
increaseQuantity.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Integer increaseQuantity = Integer.parseInt(quantity.getText().toString());
        increaseQuantity = increaseQuantity + 1;
        quantity.setText("" + increaseQuantity);
    }
});
decreaseQuantity.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Integer quantityValue = Integer.parseInt(quantity.getText().toString());

        if(quantityValue > 1){
            Integer decreaseQuantity = Integer.parseInt(quantity.getText().toString());
            decreaseQuantity = decreaseQuantity - 1;
            quantity.setText("" + decreaseQuantity);
        }
    }
});
```

Slika 4.12. Promjena količine proizvoda na klik

4.3.3. Početni zaslon aplikacije

Na glavnom zaslonu aplikacije nalaze se dva fragmenta, jedan sadrži sve proizvode iz baze podataka, dok drugi sadrži proizvode koji ističu kroz 10 dana. Prelazak s jednog fragmenta na drugi omogućen je preko TabLayout-a, koji pritiskom na određeni prozor prikazuje odabrani fragment, dok ViewPager omogućava izmjenu fragmenata na način da ga korisnik može prebaciti prelaskom prsta preko ekrana u desno, odnosno u lijevo. TabLayout povezan je s ViewPager-om kako bi se klikom na određeni prozor mogao promijeniti i fragment. Za funkcioniranje ViewPager-a bio je potreban adapter koji je predan ViewPager-u u metodi *setAdapter()*. Adapter omogućava prikaz željenog fragmenta te rukuje brojem fragmenata, njihovim nazivima i pozicijom, odnosno brine o tome da bude prikazan ispravan fragment točnog imena. Slika 4.13 prikazuje rad s fragmentima.

```
tabLayout.setupWithViewPager(viewPager);

VPAdapter vpAdapter = new VPAdapter(getSupportFragmentManager(), FragmentPagerAdapter.BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
vpAdapter.addFragments(new ProductFragment(),  title: "Products");
vpAdapter.addFragments(new NearlyExpiredFragment(),  title: "Nearly expires");
viewPager.setAdapter(vpAdapter);
```

Slika 4.13. Rad s fragmentima

Unutar fragmenata smješten je RecyclerView, za koji je također bio potreban adapter. Unutar RecyclerView-a su u oba slučaja smješteni naziv proizvoda, količina te datum isteka roka koji su dohvaćeni iz baze podataka. Također, svaki proizvod unutar RecyclerView-a može biti obrisan pritiskom na ikonicu za brisanje proizvoda. U predlošku mobilne arhitekture, taj dio predstavlja prezentera koji dohvaća podatke iz modela te brine o tome kako će oni biti prikazani. Pogled je odgovoran samo za prikaz podataka korisniku. Na slici 4.14 prikazan je dio koda koji je odgovoran za brisanje određenog proizvoda iz RecyclerView-a, a samim time i iz baze podataka. Taj dio koda nalazi se u adapteru unutar metode odgovorne za rukovanje pojedinačnim prikazom svakog proizvoda iz baze podataka. Proizvod se iz baze podataka briše preko metode *removeValue()*, a točan proizvod koji se treba obrisati dobiva se preko metode *getKey()*.

```
holder.delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseDatabase.getInstance("https://zavrsni-c3849-default-rtdb.europe-west1.firebasedatabase.app").getReference().child("Product")
                .child(getRef(position).getKey()).removeValue();
    }
});
```

Slika 4.14. Brisanje proizvoda iz baze podataka

4.3.4. Postupak raspoređivanja

Kako bi se ostvarilo raspoređivanje proizvoda prema datumu isteka roka, uzlaznim redoslijedom, bilo je potrebno dohvatiti trenutni datum u kalendaru. Trenutnom datumu je preko metode *add()* bilo potrebno dodati 10 kako bi se dobio datum koji slijedi za 10 dana. Vrijednosti trenutnog datuma spremljene su u varijable *year*, *month* i *day* te je učinjena provjera jesu li dan i mjesec manji od 10 te ukoliko jesu, dodana im je „0“ ispred kako bi kasnije usporedbe datuma davale točne rezultate. Taj postupak prikazan je na slici 4.15.

```
Calendar currentDate = Calendar.getInstance();
currentDate.add(Calendar.DATE, amount: 10);

String year  = String.valueOf(currentDate.get(Calendar.YEAR));
String month = String.valueOf(currentDate.get(Calendar.MONTH) + 1);
String day   = String.valueOf(currentDate.get(Calendar.DATE));

if (currentDate.get(Calendar.MONTH) < 10)
    month = "0" + month;
if (currentDate.get(Calendar.DATE) < 10)
    day   = "0" + day;
```

Slika 4.15. Rukovanje vrijednostima iz kalendara

Preko metode *orderByChild()* datumi isteka roka poredani su uzlaznim redoslijedom, dok metoda *endAt()* osigurava da se prikazuju samo proizvodi kojima je datum isteka roka unutar 10 dana od trenutnog datuma. Primjena tih metoda vidljiva je na slici 4.16.

```
FirebaseRecyclerOptions<nearlyExpiredModel> options =
        new FirebaseRecyclerOptions.Builder<nearlyExpiredModel>()
                .setQuery(FirebaseDatabase.getInstance("https://zavrsni-c3849-default-rtdb.europe-west1.firebasedatabase.app") FirebaseDatabase
                        .getReference() DatabaseReference
                        .child("Product").orderByChild("expirationDate") Query
                        .endAt("" + year + "-" + month + "-" + day), nearlyExpiredModel.class)
                .build();
```

Slika 4.16. Postupak eliminiranja datuma koji ističu za više od 10 dana

Kako bi korisnik bio upozoren da je proizvodu istekao rok uporabe, datum isteka roka uporabe kod proizvoda kojima je rok uporabe istekao označen je crvenom bojom. Problem proizvoda kojima je istekao rok uporabe riješen je usporedbom trenutnog datuma i datuma isteka roka uporabe određenog proizvoda te ukoliko je datum isteka roka uporabe manji od trenutnog datuma, boja teksta je postavljena na crvenu. Provjera je prikazana u kodu na slici 4.17.

```
String expDate = model.getExpirationDate();
String parts[] = expDate.split( regex: "-");

String year  = parts[0];
String month = parts[1];
String day   = parts[2];

Calendar calendar = Calendar.getInstance();
int currentDay = calendar.get(Calendar.DATE);
int currentMonth = calendar.get(Calendar.MONTH) + 1;
int currentYear = calendar.get(Calendar.YEAR);

if(Integer.parseInt(year) == currentYear && Integer.parseInt(month) == currentMonth && Integer.parseInt(day) < currentDay){
        holder.expirationDate.setTextColor(Color.parseColor( colorString: "#FF0000"));
}else if(Integer.parseInt(year) == currentYear && Integer.parseInt(month) < currentMonth){
        holder.expirationDate.setTextColor(Color.parseColor( colorString: "#FF0000"));
}else if(Integer.parseInt(year) < currentYear){
        holder.expirationDate.setTextColor(Color.parseColor( colorString: "#FF0000"));
}
```

Slika 4.17. Postupak rješavanja problema proizvoda kojima je istekao rok uporabe

4.3.5. Obavijesti aplikacije i sustav preporuka

Kako bi se korisniku prikazala obavijest, prvo je potrebno provjeriti inačicu Androida. Naime, prema [19], inačica Android 8.0 te više inačice zahtijevaju kreiranje instance klase NotificationChannel, jer se radi o novoj klasi koja nije podržana u biblioteci, a ono se odvija unutar metode *notificationChannel()*. Na taj način se daje točna informacija na koji kanal će pristizati obavijesti. Kako se kanal mora kreirati prije dostavljanja bilo kakve obavijesti na inačicama

Android 8.0 i višima, kod na slici je potrebno izvršiti što prije nakon pokretanja aplikacije. Iz tog razloga, metoda *notificationChannel()* se poziva unutar glavne aktivnosti (engl. *main activity*) u metodi *onStart()*, kao što je vidljivo na slici 4.18.

```
@Override
protected void onStart() {
    super.onStart();
    notificationChannel();
}

private void notificationChannel(){
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        CharSequence name  = "Notification";
        String id          = "NearlyExpiresNotification";
        String description = "Description";
        int importance     = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(id, name, importance);
        channel.setDescription(description);

        NotificationManager notificationManager = (NotificationManager) this.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.createNotificationChannel(channel);
    }
}
```

Slika 4.18. Metoda „notificationChannel“

Da bi se osiguralo ispravno prikazivanje obavijesti, metoda *showNotification()* pozvana je u adapteru RecyclerView-a koji se nalazi unutar fragmenta s proizvodima koji ističu kroz 10 dana. Pozvana je unutar metode koja rukuje svakim pojedinim prikazom proizvoda. Kako se nalazi u toj metodi, svakim dodavanjem proizvoda koji će isteći kroz 10 dana, automatski se aktivira obavijest te obavještava korisnika kako će proizvod isteći kroz 10 dana. Obavijest je također postavljena da se pojavljuje svaki dan u 9 sati i 30 minuta, ukoliko postoje proizvodi kojima uskoro istječe rok. Kako bi se kreiranje same obavijesti razdvojilo od rukovanja ponavljanja obavijesti, bilo je potrebno napraviti dodatnu klasu imena „NotificationReceiver“ koja je pozvana unutar Intenta. Ponavljanje obavijesti svakodnevno omogućeno je preko poziva metode *setRepeating()*. Također, za novije inačice Androida bilo je potrebno postaviti dodatan uvjet koji osigurava kako će obavijest biti prikazana i kada je uređaj u mirovanju. Sve od navedenog prikazano je na slici 4.19.

```
private void showNotification(Context context) {
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.HOUR_OF_DAY, 9);
    calendar.set(Calendar.MINUTE, 30);
    calendar.set(Calendar.SECOND, 00);

    Intent intent = new Intent(context, NotificationReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(context, requestCode: 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);

    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pendingIntent);
    }
}
```

Slika 4.19. Metoda „showNotification“

Obavijest je izgrađena u klasi „NotificationReceiver“ od strane Buildera, na kojemu je između ostaloga postavljen i naslov obavijesti, sadržaj te mogućnost klika na obavijest. Nakon pritiska na obavijest, korisnik je usmjeren na zaslon s receptima, na čemu je ujedno primijenjen i sustav preporuka. Nakon pritiska na obavijest, obavijest se automatski uklanja, što je postavljeno preko metode *setAutoCancel()*. Sve navedeno je vidljivo u kodu na slici 4.20.

```
public class NotificationReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Intent repeatingIntent = new Intent(context, RecipesActivity.class);
        repeatingIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        PendingIntent pendingIntent = PendingIntent.getActivity(context, requestCode: 0, repeatingIntent, PendingIntent.FLAG_UPDATE_CURRENT);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(context, channelId: "NearlyExpiresNotification")
                .setContentIntent(pendingIntent)
                .setSmallIcon(R.drawable.ic_notification)
                .setContentTitle("Nearly expires!")
                .setContentText("You have products that expire in less than 10 days")
                .setAutoCancel(true);

        NotificationManagerCompat notificationManager = NotificationManagerCompat.from(context);
        notificationManager.notify( id: 200, builder.build());
    }
}
```

Slika 4.20. Klasa „NotificationReceiver“

Sustav preporuka mogao je biti ostvaren i na drukčiji način, kao što je navedeno u modelu mobilne aplikacije. Dakle, nakon pritiska na obavijest, recepti su mogli biti sortirani na način da bi na vrhu bili recepti koji sadrže proizvode kojima rok istječe kroz 10 dana, dok bi prema dnu bili recepti koji sadrže ostale proizvode iz baze podataka. Da je funkcionalnost ostvarena na taj način, bila bi primijenjena po uzoru na metodu hibridnog sustava preporuka koja sortira stavke prema tom modelu.

5. NAČIN RADA I ISPITIVANJE RADA APLIKACIJE

U ovom poglavlju prikazan je način rada aplikacije te funkcionalnosti koje aplikacija ima. Također, prikazani su ispitni slučajevi koji prikazuju rad aplikacije.

5.1. Način rada mobilne aplikacije

Nakon što je aplikacija instalirana, korisniku se prikazuje zaslon s dva gumba, od kojih jedan vodi do zaslona za skeniranje QR koda, dok drugi vodi na zaslon s receptima. Također, na početnom zaslonu nalaze se dvije kategorije. Prva prikazuje sve proizvode koje je korisnik unio, odnosno koje posjeduje dok druga kategorija prikazuje proizvode kojima rok istječe kroz 10 dana. Nakon instalacije te kategorije su prazne jer se u bazi podataka ne nalaze proizvodi.

Nakon pritiska na gumb za unos proizvoda, korisnika se usmjerava na zaslon koji sadrži čitač QR kodova te gumb za ručni unos podataka o proizvodu. Pri ulasku na taj zaslon, korisnika se traži dopuštenje za korištenje kamere. Ukoliko korisnik odbije koristiti kameru, dopušten mu je pritisak na gumb za ručni unos podataka. Dopuštenje o kameri korisnik može promijeniti u svakom trenutku u postavkama svog mobilnog uređaja. U trenutku kada korisnik da dopuštenje za kameru, odabir se automatski pamti, što bi značilo da ponovnim ulaskom na zaslon s čitačem QR koda korisnika se neće ponovno tražiti dopuštenje za korištenje kamere. Ukoliko se korisnik odluči za unos podataka preko QR koda, nakon očitavanja QR koda korisnik je automatski preusmjeren na zaslon za popunjavanje podataka o proizvodu. Podaci iz QR koda automatski su uneseni u polja za ime proizvoda te za datum isteka roka. Polje koje nije promijenjeno je polje za količinu proizvoda te u njemu stoji podrazumijevana vrijednost, koja je ujedno i minimalna te iznosi 1. Nakon što je korisnik popunio željene podatke, pruža mu se mogućnost spremanja proizvoda te mogućnost povratka na početni zaslon.

Pritiskom na gumb za prikaz recepata, korisnik je preusmjeren na zaslon s listom recepata. Recepti su sadržani u bazi podataka te nisu promjenjivi. Odabirom željenog recepta, korisnik se usmjerava na zaslon s detaljima o tom pojedinom receptu. Zaslon sadrži naziv recepta, sliku recepta, sastojke potrebne za pripremanje recepta te korake pripremanja recepta.

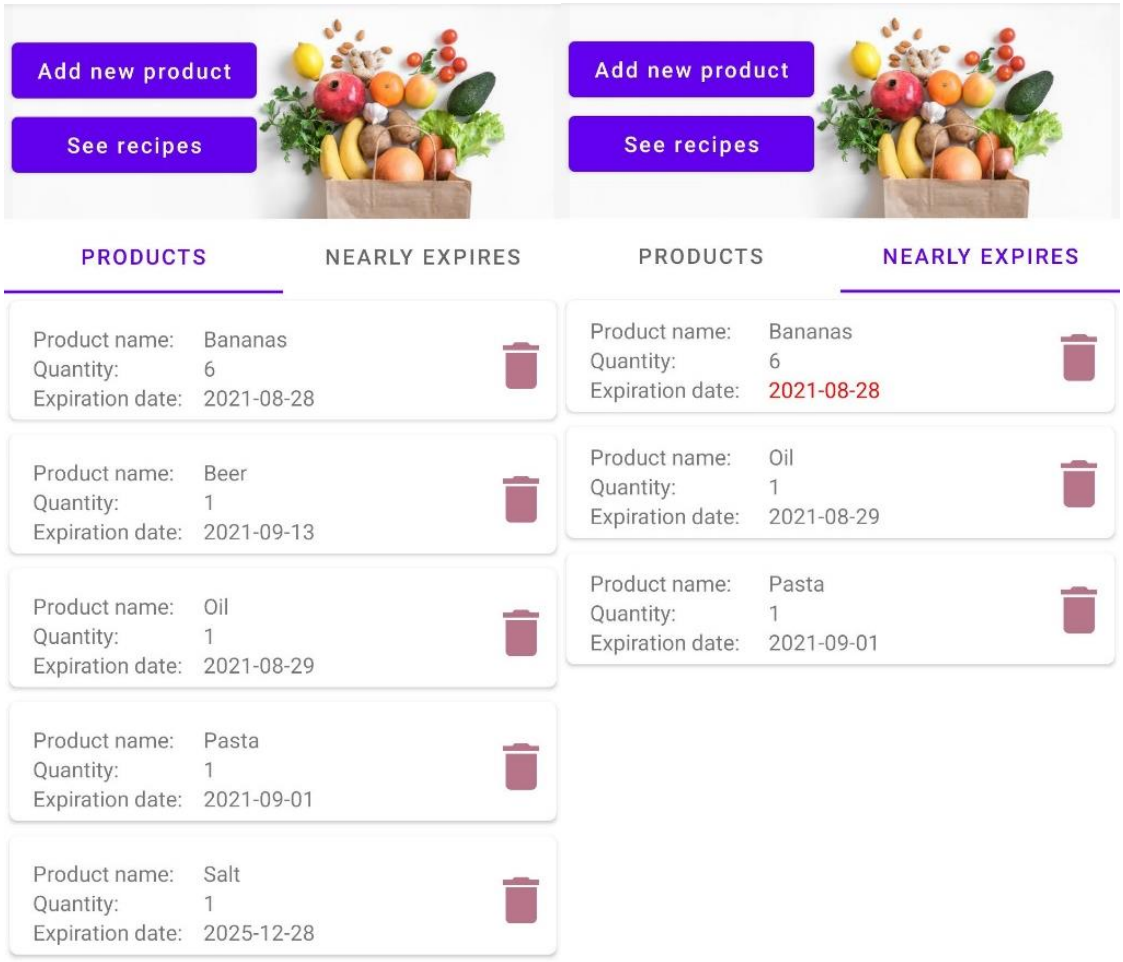
5.2. Prikaz rada aplikacije

5.2.1. Prikaz početnog zaslona

Na početnom zaslonu se, kao što je već spomenuto, nalaze dvije kategorije proizvoda koje prikazuju sve proizvode te proizvode kojima rok uporabe istječe kroz 10 dana. Prema slici 5.1,

korisnik svaki proizvod može ukloniti. Unutar kategorije „NEARLY EXPIRES“ može se vidjeti proizvod kojemu je datum isteka roka crvene boje. To bi značilo da je proizvodu istekao rok uporabe. Proizvod se automatski ne uklanja iz baze podataka iz razloga da bi korisnik aplikacije bio upozoren kako postoji proizvod kojemu je rok uporabe istekao.

Pritiskom na gumb „Add new product“, korisnik ima mogućnost prelaska na zaslon koji sadrži čitač QR koda (Slika 5.3) te pritiskom na gumb za prikaz recepata („See recipes“) preusmjeren je na zaslon koji sadrži nazive recepata te njihove slike. (Slika 5.5).

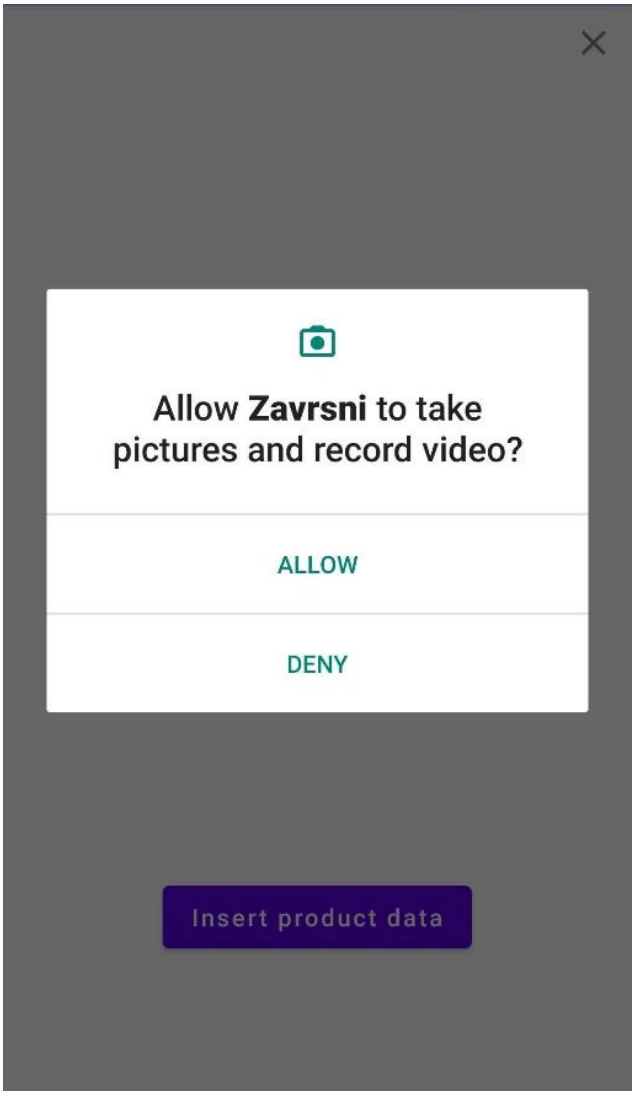


Slika 5.1. Početni zaslon aplikacije

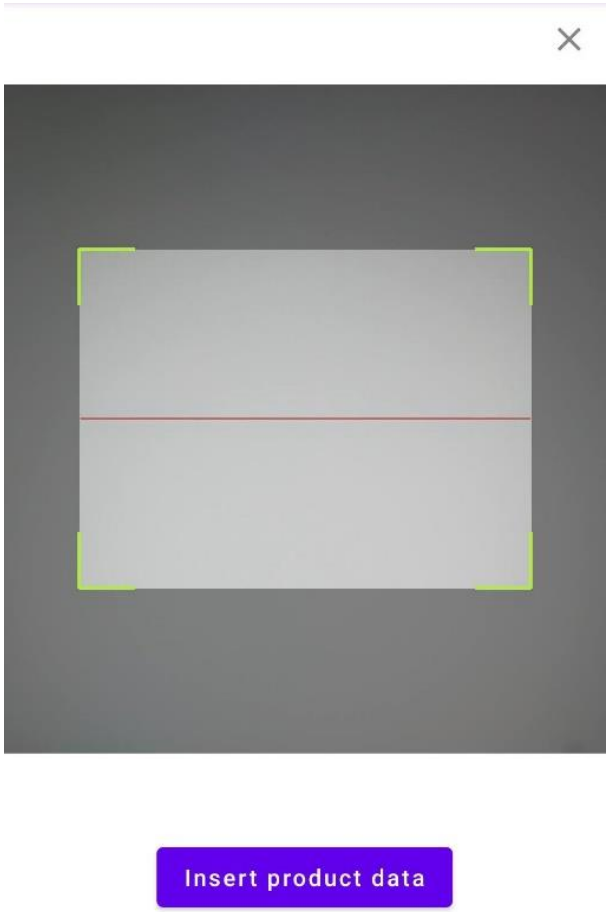
5.2.2. Prikaz čitača QR koda

Na zaslonu koji sadrži čitač QR koda, pri prvom ulasku korisnika se traži dopuštenje za pokretanje kamere (Slika 5.2). Pritiskom na gumb „ALLOW“ pali se kamera i korisniku se omogućava očitavanje QR koda. Ukoliko korisnik ne želi proizvod očitati korištenjem QR čitača, pruža mu se mogućnost da pritiskom na gumb „Insert product data“ podatke o proizvodu unese ručno. U

slučaju da se odluči za očitavanje QR koda, nakon što je kod očitan, korisnik je automatski preusmjeren na zaslon za unos detalja o proizvodu, a podaci iz QR koda popunjavaju polja o nazivu proizvoda i datumu isteka roka. Ukoliko je proizvodu istekao rok uporabe, nakon očitavanja QR koda prikazuje se poruka koja obavještava korisnika kako je rok istekao te mu pruža mogućnost povratka na početni zaslon. Detaljnije o tom slučaju bit će opisano u potpoglavlju 5.3.2.



Slika 5.2. Zahtjev za dopuštenje korištenja kamere uređaja



Slika 5.3. Sučelje s čitačem QR koda

5.2.3. Prikaz zaslona za unos proizvoda

Zaslon za unos proizvoda, kao što je vidljivo na slici 5.4, sadrži polje za unos naziva proizvoda, polje za količinu koja se može povećavati ili smanjivati na gumbove ili se količina može unijeti ručno. Također, sadrži prikaz oblika kalendara u kojemu se može podesiti datum isteka roka proizvoda. Nakon što je korisnik unio podatke, pritiskom na gumb „Add product“ prikazuje mu se poruka kako je proizvod unesen u bazu, dok pritiskom na gumb „BACK“ biva preusmjeren na početni zaslon.

Slika 5.4. Sučelje za unos proizvoda

5.2.4. Prikaz zaslona s receptima

Zaslon s receptima, kao što je već spomenuto, sadrži nazive recepata te njihove slike, što je vidljivo na slici 5.5. Nakon pritiska na željeni recept, korisnik je preusmjeren na zaslon s informacijama o sastojcima i koracima izrade odabranog recepta (Slika 5.6). Recepti se ne mogu dodavati niti uklanjati.

Slika 5.5. Sučelje recepata

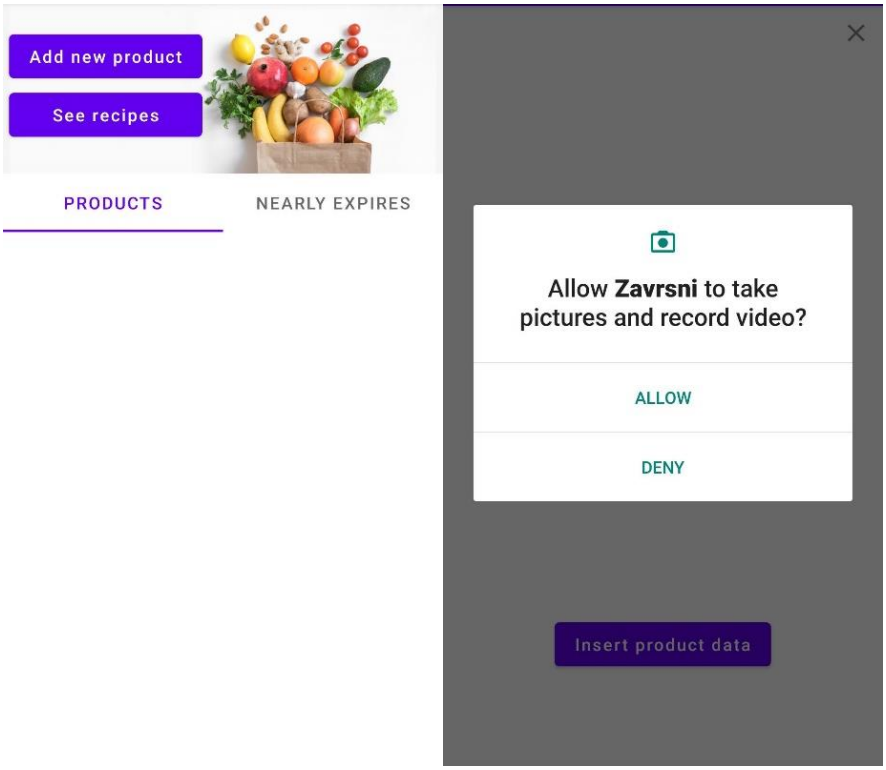
Slika 5.6. Sučelje informacija o receptu

5.3. Ispitivanje rada aplikacije

Kako bi se prikazale sve gore navedene funkcionalnosti, provedena su tri ispitna slučaja. U ispitnom slučaju 1 prikazan je rad aplikacije nakon što ju korisnik prvi puta pokrene. Ispitni slučaj 2 nadopunjavat će ispitni slučaj 1 i prikazat će rad aplikacije kada se korisnik odluči na očitavanje QR koda s proizvoda. Također, ispitni slučaj 2 sadržavat će dva QR koda koja će biti očitani u tom ispitnom slučaju. Jedan QR kod sadržavat će podatke o proizvodu kojemu nije istekao rok uporabe, dok će drugi sadržavati podatke o proizvodu kojemu je istekao rok uporabe. Ponašanje aplikacije nakon očitavanja oba bit će prikazana u tom slučaju. U ispitnom slučaju 3 bit će prikazan rad aplikacije kada postoje proizvodi kojima je istekao rok uporabe te ostvaren sustav obavijesti.

5.3.1. Ispitni slučaj 1 – sučelje nakon instalacije aplikacije

Nakon instalacije aplikacije, kao što je već spomenuto, u kategorijama koje sadrže sve proizvode koje korisnik posjeduje te one kojima uskoro istječe rok ne nalazi se niti jedna stavka jer je baza podataka u tom trenutku prazna. Korisnika se pri prvom ulasku na sučelje s čitačem QR koda s proizvoda traži dopuštenje za korištenje kamere uređaja kako bi QR kod mogao biti očitan. Sve navedeno vidljivo je na slici 5.7.



Slika 5.7. Prikaz sučelja aplikacije pri prvom korištenju aplikacije

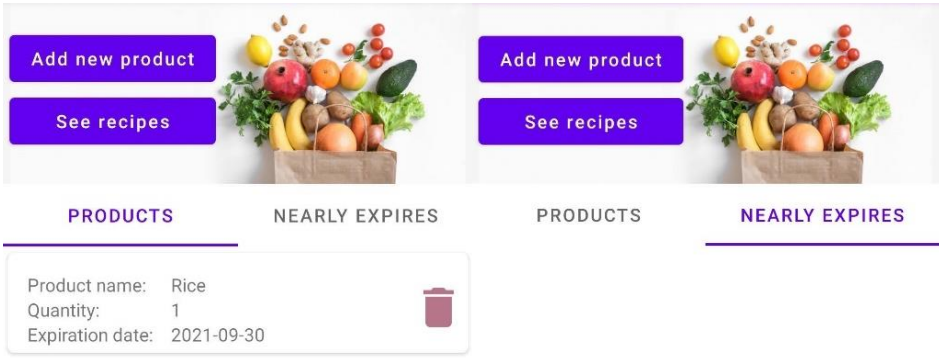
5.3.2. Ispitni slučaj 2 – proizvodu je valjan odnosno istekao rok uporabe

Ukoliko QR kod sadrži podatke o proizvodu kojemu je rok uporabe valjan, korisnik će nakon očitavanja QR koda biti preusmjeren na sučelje s podacima o proizvodu. Podaci o nazivu proizvoda i datumu isteka roka popunjavaju se iz QR koda. Korisnik iste može prilagođavati po želji. Slika 5.8 prikazuje QR kod te njegov sadržaj, dok slika 5.9 prikazuje sučelje za ispunjavanje podataka o proizvodu. Slika 5.10 prikazuje početni zaslon nakon što je proizvod spremljen. Proizvod se ne nalazi u kategoriji „NEARLY EXPIRES“ jer mu datum isteka roka nije kroz 10 dana od današnjeg datuma.

Rice%%30%%9%%2023

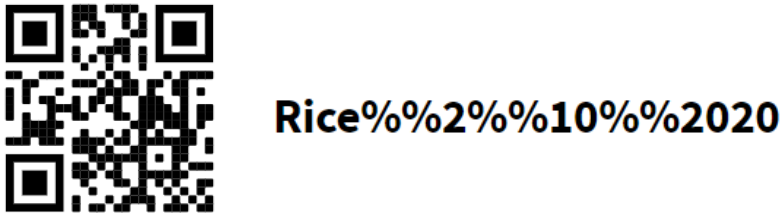
Slika 5.8. QR kod i njegov sadržaj

Slika 5.9. Sučelje s informacijama o proizvodu očitanim iz QR koda



Slika 5.10. Početni zaslon nakon unosa proizvoda

U slučaju kada QR kod sadrži podatke o proizvodu kojemu je istekao rok uporabe, kao na slici 5.11, korisniku se prikazuje poruka da je proizvodu istekao rok uporabe (Slika 5.12) te mu se pritiskom na gumb „OK“ pruža mogućnost povratka na početni zaslon.

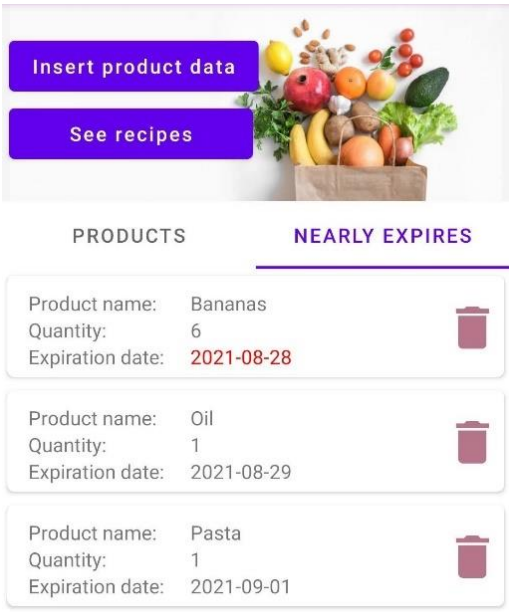


Slika 5.11. Prikaz QR koda za konkretni proizvod

Slika 5.12. Izgled sučelja nakon očitavanja QR koda proizvoda kojemu je istekao rok uporabe

5.3.3. Ispitni slučaj 3 – proizvodu je istekao rok uporabe i obavijesti aplikacije

Ukoliko unutar aplikacije postoje proizvodi kojima je rok uporabe istekao, datumi isteka roka uporabe tih proizvoda bit će označeni crvenom bojom, kao što je vidljivo na slici 5.13. Obavijesti aplikacije prikazane su na slici 5.14. Pritiskom na obavijest, korisnik biva preusmjeren na zaslon s receptima te je na taj način ostvaren sustav preporuka.



Slika 5.13. Prikaz ponašanja aplikacije u slučaju da postoji proizvod kojemu je istekao rok

Slika 5.14. Prikaz obavijesti za proizvode koji uskoro ističu

5.4. Analiza rada aplikacije

U mobilnoj aplikaciji za praćenje korištenja prehrambenih proizvoda ostvarena je funkcionalnost očitavanja QR koda proizvoda. U slučaju kada je rok uporabe istekao, proizvod se ne može spremiti u bazu podataka te se o tome obavještava korisnika. U slučaju kada je rok uporabe važeći, korisnika se usmjerava na drugu aktivnost preko koje unosi proizvode u bazu podataka. Također, ostvarena je funkcionalnost raspoređivanja proizvoda prema datumu isteka roka. Ta funkcionalnost je mogla biti ostvarena i kroz recepte, na način da bi se korisniku na vrhu liste

predlagali recepti u kojima može iskoristiti proizvode koji imaju najkraći rok trajanja, a zatim recepte koji sadrže ostale proizvode. Također, ostvaren je sustav obavijesti, koji ukoliko postoje proizvodi u bazi podataka kojima je rok isteka uporabe kroz 10 dana šalje obavijest svakodnevno dok se ti proizvodi ne potroše ili uklone. Sustav preporuka je djelomično ostvaren, odnosno, pritiskom na obavijest aplikacije korisnik biva usmjeren na aktivnost s receptima. Recepti su nepromjenjivi te ne ovise o proizvodima koji se nalaze u bazi podataka. Da je sustav preporuka implementiran na način na koji je opisan u modelu aplikacije, sučelje s receptima bi sadržavalo samo recepte koji sadrže sastojke koji se nalaze u bazi podataka.

6. ZAKLJUČAK

U ovom završnom radu razvijena je mobilna Android aplikacija za praćenje korištenja prehrambenih proizvoda. Cilj aplikacije je olakšati korisniku praćenje proizvoda koje ima u kućanstvu kroz praćenje rokova uporabe proizvoda. Uz to, aplikacija omogućuje čitanje QR koda proizvoda i stvaranje obavijesti koje svakodnevno pristižu ukoliko postoje proizvodi kojima uskoro istječe rok uporabe. Također, u mobilnoj aplikaciji nalaze se recepti korištenjem kojih se mogu učinkovito iskoristiti proizvodi u kućanstvu. Na taj način, bacanje hrane korištenjem aplikacije može postati prošlost. Aplikacija se oslanja na postupke raspoređivanja i sustav preporuka korištenja proizvoda. Pojednostavljeno raspoređivanje ostvareno je tako što se u kategoriji proizvoda kojima uskoro istječe rok trajanja proizvodi raspoređuju prema datumu isteka roka trajanja od najranijeg do najkasnijeg datuma isteka. Sustav preporuka omogućuje da se korisnik pritiskom na obavijest preusmjerava na recepte u bazi podataka.

Aplikacija je ostvarena u razvojnom okruženju Android Studio. Programski jezik korišten za ostvarivanje aplikacije je Java, dok je XML korišten kao jezik označavanja. Ispitni slučajevi su pokazali ispunjavanje funkcionalnosti aplikacije od kojih su neke ostvarene u potpunosti, a neke djelomično. U ostvarene funkcionalnosti aplikacije spada korištenje čitača QR koda, raspoređivanje proizvoda prema roku trajanja te sustav obavijesti, dok se sustav preporuka svodi na preusmjeravanje korisnika na recepte. U budućnosti postoji mogućnost poboljšanja i nadogradnje aplikacije tako da se sustav preporuka ostvari kroz predlaganje recepata čiji su sastojci sadržani u bazi podataka s proizvodima. Također, sustav raspoređivanja može se implementirati tako da se recepti raspoređuju prema razini prvenstva korištenja proizvoda koje ti recepti sadrže kao sastojke. Također, bilo bi moguće raditi predlaganje proizvoda i recepata na temelju profila korisnika, zdravstvenog stanja korisnika i sličnih značajki.

LITERATURA

[1] B. Bilska, D. Kołozyn-Krajewska, A. Łepecka, K. Marciniak-Łukasiak, K. Neffe-Skocinska, A. Szydłowska, M. Tomaszewska, M. Trzaskowska, D. Zielińska, Consumer Understanding of the Date of Minimum Durability of Food in Association with Quality Evaluation of Food Products After Expiration, International Journal of Environmental Research and Public Health, br. 5, sv. 17, str. 1-2, Ožujak, 2020.

[2] K.-C. Chou, R.-Z. Wang, The Nested QR Code, IEEE Signal Processing Letters, sv. 27, str. 1, Srpanj, 2020.

[3] Denso, https://www.denso-wave.com/en/technology/vol1.html, pristupljeno 09.07.2021.

[4] P. Desai, U. Panchal, B. Patel, Methods of Recommender System: A Review, 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), str. 1, Veljača, 2018.

[5] V. Artsibasov, V. Chystyak, M. Lobur, Y. Stekh, Methods and Tools for Building Recommender Systems, The Experience of Designing and Application of CAD Systems in Microelectronics, str. 1, Ukrajina, Studeni, 2015.

[6] M. H. Ibrahim, M. H. Khafagy, M. H. Mohamed, Recommender Systems Challenges and Solutions Survey, 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), str. 149 - 152, Egypt, Veljača, 2019.

[7] J. Radej, Sustavi za rad u stvarnom vremenu, Zagreb, 2009.

[8] L. Budin, Ž. Hocenski, G. Martinović, Undergraduate Teaching of Real-Time Scheduling Algorithms by Developed Software Tool, IEEE Transactions on Education, sv. 46, str. 185 - 188, Veljača, 2003.

[9] KitchenPal, https://play.google.com/store/apps/details?id=fr.icuisto.icuisto&hl=en_US&gl=US, pristupljeno 09.07.2021.

[10] Magic Fridge, https://play.google.com/store/apps/details?id=fr.haruni.frigomagic&hl=hr&gl=US, pristupljeno 09.07.2021.

[11] Best Before, https://play.google.com/store/apps/details?id=com.peytu.bestbefore&hl=en_US&gl=US, pristupljeno 09.07.2021.

[12] Android Studio, https://developer.android.com/studio/intro, pristupljeno 09.07.2021.

[13] Java, https://en.wikipedia.org/wiki/Java_(programming_language), pristupljeno 09.07.2021.

[14] XML, https://www.w3schools.com/xml/xml_whatis.asp, pristupljeno 09.07.2021.

[15] Dexter, https://www.geeksforgeeks.org/easy-runtime-permissions-in-android-with-dexter/, pristupljeno 04.09.2021.

[16] ZXing, https://github.com/zxing/zxing, pristupljeno 04.09.2021.

[17] Firebase, https://firebase.google.com/, pristupljeno 10.07.2021.

[18] Model View Presenter(MVP) in Android with a simple demo project, https://medium.com/cr8resume/make-you-hand-dirty-with-mvp-model-view-presenter-eab5b5c16e42, pristupljeno 05.09.2021.

[19] developers, https://developer.android.com/training/notify-user/channels, pristupljeno 04.09.2021.

POPIS SLIKA

Slika 2.1. Uzorci za pronalaženje ispravnog položaja koda [3] 3
Slika 2.2. Korisničko sučelje aplikacije KitchenPal 6
Slika 2.3. Korisničko sučelje aplikacije Magic Fridge 7
Slika 2.4. Korisničko sučelje aplikacije Best Before 8
Slika 3.1. Tijek aktivnosti nakon očitavanja QR koda 9
Slika 3.2. Prikaz proizvoda u bazi podataka Firebase 10
Slika 4.1. Primjer traženja dopuštenja korisnika [15] 13
Slika 4.2. Predložak programske arhitekture MVP [18] 14
Slika 4.3. Biblioteke za ostvarivanje QR koda i dopuštenja za kameru 15
Slika 4.4. Primjena Dexter biblioteke 15
Slika 4.5. Rukovanje rezultatom QR čitača 16
Slika 4.6. Provjera valjanosti roka uporabe proizvoda 17
Slika 4.7. Prikaz obavijesti o isteku roka 17
Slika 4.8. Extras ukoliko korisnik odluči unijeti podatke o proizvodu ručno 18
Slika 4.9. Trenutni datum omogućen kao minimalni 18
Slika 4.10. Dohvaćanje i postavljanje podataka u odgovarajuća polja 19
Slika 4.11. Spremanje podataka u bazu podataka te postavljanje vrijednosti polja na početne ... 20
Slika 4.12. Promjena količine proizvoda na klik 21
Slika 4.13. Rad s fragmentima 21
Slika 4.14. Brisanje proizvoda iz baze podataka 22
Slika 4.15. Rukovanje vrijednostima iz kalendara 22
Slika 4.16. Postupak eliminiranja datuma koji ističu za više od 10 dana 23
Slika 4.17. Postupak rješavanja problema proizvoda kojima je istekao rok uporabe 23
Slika 4.18. Metoda „notificationChannel“ 24
Slika 4.19. Metoda „showNotification“ 25
Slika 4.20. Klasa „NotificationReceiver“ 25
Slika 5.1. Početni zaslon aplikacije 27
Slika 5.2. Zahtjev za dopuštenje korištenja kamere uređaja 28
Slika 5.3. Sučelje s čitačem QR koda 29
Slika 5.4. Sučelje za unos proizvoda 30
Slika 5.5. Sučelje recepata 31
Slika 5.6. Sučelje informacija o receptu 31

Slika 5.7. Prikaz sučelja aplikacije pri prvom korištenju aplikacije 32
Slika 5.8. QR kod i njegov sadržaj 33
Slika 5.9. Sučelje s informacijama o proizvodu očitanim iz QR koda 33
Slika 5.10. Početni zaslon nakon unosa proizvoda 34
Slika 5.11. Prikaz QR koda za konkretni proizvod 34
Slika 5.12. Izgled sučelja nakon očitavanja QR koda proizvoda kojemu je istekao rok uporabe 34
Slika 5.13. Prikaz ponašanja aplikacije u slučaju da postoji proizvod kojemu je istekao rok 35
Slika 5.14. Prikaz obavijesti za proizvode koji uskoro ističu 35

SAŽETAK

U ovom završnom radu razvijena je mobilna Android aplikacija za praćenje korištenja prehrambenih proizvoda uz korištenje postupka raspoređivanja i stvaranja preporuka. U teorijskom dijelu rada opisane su funkcionalnosti mobilne aplikacije, prikazana su postojeća programska rješenja za praćenje rokova uporabe proizvoda, a model aplikacije definira njene funkcionalnosti i postupke kojima su one omogućene. U aplikaciji je omogućeno korištenje čitača QR koda, stvaranje obavijesti o valjanosti proizvoda i njihovo raspoređivanje prema isteku roka. Sustav preporuka samo je djelomično ostvaren. Programsko rješenje mobilne aplikacije zasnovano na predlošku programske arhitekture MVP ispitano je za više korisničkih slučajeva koji pokazuju navedene funkcionalnosti.

Ključne riječi: mobilna Android aplikacija, obavijesti, postupak raspoređivanja, prehrambeni proizvodi, recepti, sustav preporuka.

ABSTRACT

Title: The mobile Android application for tracking the usage of nutritions featured with methods of scheduling and making suggestions

This final paper sums up the process of mobile Android application development. The application defines activities for tracking the usage of nutritions featured with methods of scheduling and making suggestions. The theoretical part of the final paper describes the mobile application functionalities and needful algorithm for tracing products' "best used until" dates. The model of a mobile application explains the basis of the application's functionalities and procedures. Some of them are QR code reader, notifications about products' validity, and their scheduling according to "best used until" date. The recommendation system is partially implemented with other application functionalities. Program solution of mobile application based on the architectural pattern MVP gives positive testing results on several testing cases.

Keywords: mobile Android application, notifications, nutritions, recipes, the recommendation system, the scheduling method.

ŽIVOTOPIS

Kristina Bešlić rođena je 10. siječnja 2000. godine u Novoj Gradiški. Pohađala je Osnovnu školu „Mato Lovrak“ u Novoj Gradiški. 2014. godine upisuje opću gimnaziju u Novoj Gradiški koju završava 2018. godine. Iste godine upisuje preddiplomski studij računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI

Prilog 1. Završni rad „Mobilna android aplikacija za praćenje korištenja prehrambenih proizvoda na temelju sustava preporuka i raspoređivanja“ u *.docx* formatu

Prilog 2. Završni rad „Mobilna android aplikacija za praćenje korištenja prehrambenih proizvoda na temelju sustava preporuka i raspoređivanja“ u *.pdf* formatu

Prilog 3. Programski kod aplikacije