Mobilna aplikacija za preporuku i odabir kulinarskih recepta

Radić, Ivana

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Ivana Radić

Mobilna aplikacija za preporuku i odabir kulinarskih recepta

Pula, rujan, 2022

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Ivana Radić

Mobilna aplikacija za preporuku i odabir kulinarskih recepta
Diplomski rad

JMBAG: 0303068875, redoviti student

Studijski smjer: Informatika

Predmet: Mobilne aplikacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv.prof. dr. sc. Siniša Sovilj

Pula, rujan, 2022.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani **Ivana Radić**, kandidat za magistra **informatike** ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____Ivana Radić_____

U Puli, rujan 2022. godine

IZJAVA
o korištenju autorskog djela

Ja, Ivana Radić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „ Mobilna aplikacija za preporuku i odabir kulinarskih recepta“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan, 2022.godine

Potpis

_____Ivana Radić_____

Pula, 28. ožujka 2021.

DIPLOMSKI ZADATAK

Pristupnik:	**Radić Ivana 0802998383330**
Studij:	Sveučilišni diplomski studij Informatike
Naslov (hrv.):	**Mobilna aplikacija za preporuku i odabir kulinarskih recepta.**
Naslov (eng.):	Mobile application for recommending and selecting culinary recipes.
Opis zadatka:	Zadatak je izraditi mobilnu aplikaciju za preporuku kulinarskih recepta. U aplikaciji korisniku treba omogućiti registraciju, unos novih recepata (slike, video, opis sastojaka, pripreme, tipa recepta, tipa dijete i sl.). Omogućiti pretraživanje recepta po imenu, tipu dijete ili po namirnicama koje korisnik trenutno ima dostupne u svom hladnjaku. Nakon odabira željenog recepta, omogućiti modifikaciju recepata s obzirom na količinu, a sastojke koje korisnik nema dostupne u svom hladnjaku, a potrebni su za odabrani recept mogu se staviti i na listu za kupovinu. Omogućiti spremanja najdražih recepata na posebnu stranicu za lakše pretraživanje. Omogućiti društveno komentiranje i ocjenjivanje recepata i slika. Aplikacija se treba temeljiti na tehnologijama Java ili Kotlin koristeći Android Studio. Opisati sustav: korisničke scenarije, funkcionalnosti, izraditi potrebne UML dijagrame - klasne, *use case*, *use sequence*, opisati implementaciju te na kraju izraditi kratke korisničke upute.

Zadatak uručen pristupniku: 28. ožujka 2021.
Rok za predaju rada: 28. veljače 2022.

Mentor:

Siniša Sovilj

_____

doc.dr.sc. Siniša Sovilj

SADRŽAJ

1.UVOD

Broj korisnika pametnih telefona svake godine se sve više povećava tako da danas više od 83% ukupnog stanovništva koristi pametne telefone. Dva vodeća operativna sustava za pametne telefone su android i iOS. Android je korišteniji operativni sustav, koristi ga 71% populacije, jer su i cijene pametnih telefona koji koriste android za svoj operativni sustav pristupačnije većini populacije.

Android mobilna aplikacija Cook Nook napravljena je u Android Studiju koristeći Kotlin programski jezik. Cook Nook omogućava korisnicima da naprave vlastiti korisnički račun koristeći Firebase autentifikaciju. U Firebase Firestore spremaju se recepti koje korisnik unosi a u Firebase Storage sprema se slika recepta. Korisnik može pregledati vlastite recepte i obrisati ih, također može po imenu pretražiti recepte unesene od strane drugih korisnika.

Ova aplikacija idealna je za korisnike koji vole kuhati, korisnike koji imaju vlastite recepte koje žele podijeliti s drugima ili jednostavno imati na dohvat ruke. Naziv Cook Nook dolazi iz engleskog jezika te znači kutak za kuhanje.

Aplikacija se sastoji od šest aktivnosti, dva adaptera i dvije klase podataka. Prve dvije aktivnosti s kojima se korisnik susreće su aktivnosti za prijavu i registraciju. Nakon što korisnici unesu svoje podatke za prijavu odnosno registraciju otvara se aktivnost Recipes, ova aktivnost je početna stranica na kojoj korisnik ima pregled vlastitih recepata te upravljanje istima. Ova aktivnost nudi korisnicima koji imaju mnogo vlastitih recepata mjesto na koje ih mogu digitalno pohraniti da im uvijek budu na dohvat ruke.

Ako korisnik želi istražiti recepte drugih korisnika aplikacije to može učiniti na aktivnosti Search Recipes. Pri otvaranju ove aktivnosti prikažu se svi recepti spremljeni u bazu podataka te ih korisnik može pretražiti prema imenu željenog recepta. Klikom na recepte otvara se aktivnost View Recipe gdje korisnik ima detaljan pregled recepta.

Kako bi korisnik pohranio vlastite recepte u bazu podataka potrebno je da ode na aktivnost Add Recipe gdje unosi podatke o receptu te ga sprema.

2. ANDROID STUDIO

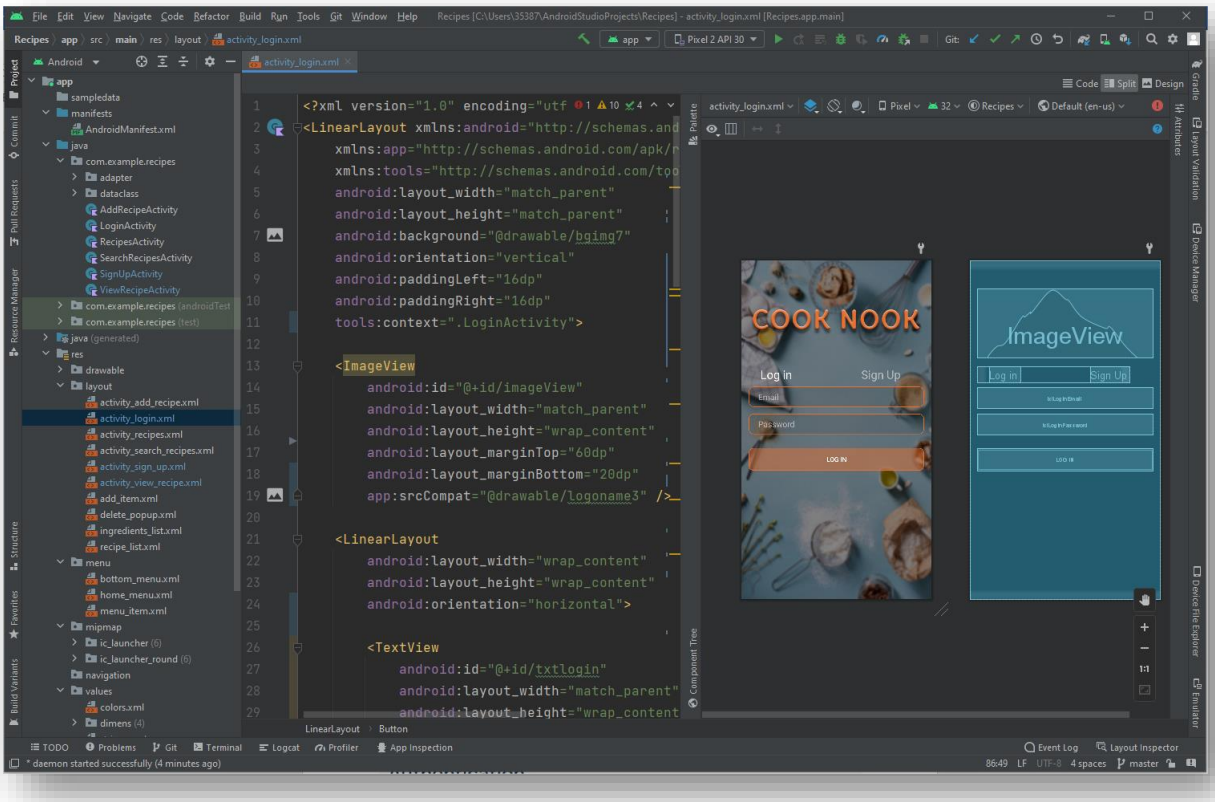
Android Studio je integrirano razvojno okruženje (IDE) za razvoj android aplikacija. Google i JetBrains su razvili android Studio 2013 godine. Dostupan je za Windows, Mac OS X i Linux operativni sustav.

U Android Studiju korisnik može kreirati i testirati svoju mobilnu aplikaciju na emulatoru ili na vlastitom uređaju također može pogledati dizajn svoje aplikacije u stvarnom vremenu. Android Studio koristi komplete za razvoj softvera (SDK) koji pokriva uključivanje potrebnih alata kao što su dokumenti i okviri. SDK također uključuje emulator mobilnog uređaja za testiranje razvojnog napretka. Emulator je alat koji pomaže pokrenuti android aplikaciju s korisnikovog računala kako bi mogao testirati hardversku i softversku kompatibilnost aplikacije s bilo kojeg uređaja. Sastoji se i od integriranog razvojnog okruženja (IDE) koji pomaže razvojnom programeru da napiše potrebne kodove za razvoj aplikacije kombiniranjem kompleta za razvoj softvera te upravljanjem istih s obzirom na korisnikove potrebe. JAVA i Kotlin su programski jezici koji se najviše koriste tijekom razvoja aplikacije u Android Studiju ali Android Studio podržava i ostale programske jezike. Android Studio koristi biblioteke koje pomažu korisniku u otkrivanju grešaka tijekom pisanja koda te njegovo programsko sučelje prepoznaje uzorke u kodiranju te pomaže korisniku davanjem prijedloga, te time pomaže korisniku da brže i učinkovitije piše kod. Značajka predložaka koda pomaže i novim i starim programerima kako bi se snašli koristeći Android Studio. Ovi predlošci koda sadrže različite vrste uzoraka koda kao navigacijski sustav kako bi korisnik mogao lakše napisati teške kodove. Provjera vlakana je također jedan od korisnih karakteristika Android Studija. Ova značajka pomaže u kontroli strukturalnog integriteta koda otkrivanjem slabih struktura u kodu pomaže korisniku da poboljša svoj kod. Android Studio također uključuje alate za testiranje kao što su JUnit4 i Functional-UI koji pomažu u testiranju okvira koda spremanjem napisanih kodova i generiranjem testova za kodove u korisničkom sučelju emulatora. Pomaže korisniku odrediti kontinuirani integritet aplikacije.

Neke od nedostataka Android Studija su to što zahtjeva minimalno 4 gigabajta radne memorije i i3 procesor tako da ga nije moguće koristiti na nekom slabijem računalu. Emulator iako je odličan za testiranje na raznim uređajima ili za korisnika koji ne

posjeduje android pametni telefon je jako spor te ga je potrebno često obrisati i napraviti novi kako bi bolje radio. Korištenje vlastitog android uređaja za testiranje aplikacije pomaže izbjeći ovaj problem.

InteliJ IDEA, Visual Studio, Eclipse, Xamarin i Xcode su popularne alternative i konkurenti Android Studiju. Za dizajn aplikacije Android Studio koristi XML opisni jezik, za razliku od HTML-a koji ima fokus na izgled podataka XML je dizajniran tako da prenosi podatke s fokusom na same podatke.



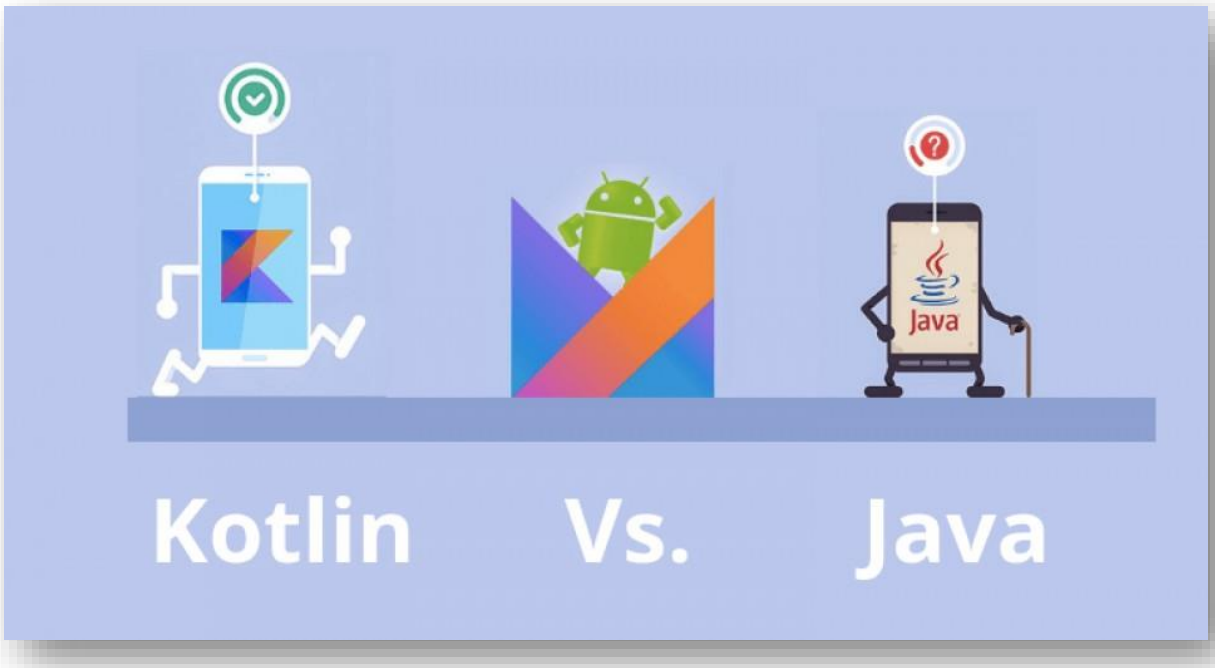
Slika 1: Sučelje Android Studija

3. KOTLIN

Za izradu mobilne aplikacije CookNook korišten je programski jezik Kotlin. Kotlin je programski jezik napravljen od strane JetBrains tvrtke 2011. godine. JetBrains je tvrtka koja prodaje integrirana razvojna okruženja za programske jezike. Otkako je nastao Kotlin postao je omiljeni jezik programera i zamijenio Javu u mnogim softverskim projektima.

Kotlinova kreacija nastala je nakon što je glavi programer Dmirty Jamerov tražio značajke koje nije mogao pronaći u Javi, Scali ili nekim drugim jezicima koji radi na JVM (Java Virtual Machine). Jamerov je želio jezik koji ima sve značajke modernijih programskih jezika, koji će raditi na JVM i kompajlirati jednako brzo kao Java.

Kotlin je osmišljen kao zamjena za Javu za Android operativne sustave. Osam godina nakon nastanka Kotlina 2019. Google je proglasio Kotlinom kao preferirani jezik za razvoj Android aplikacija.



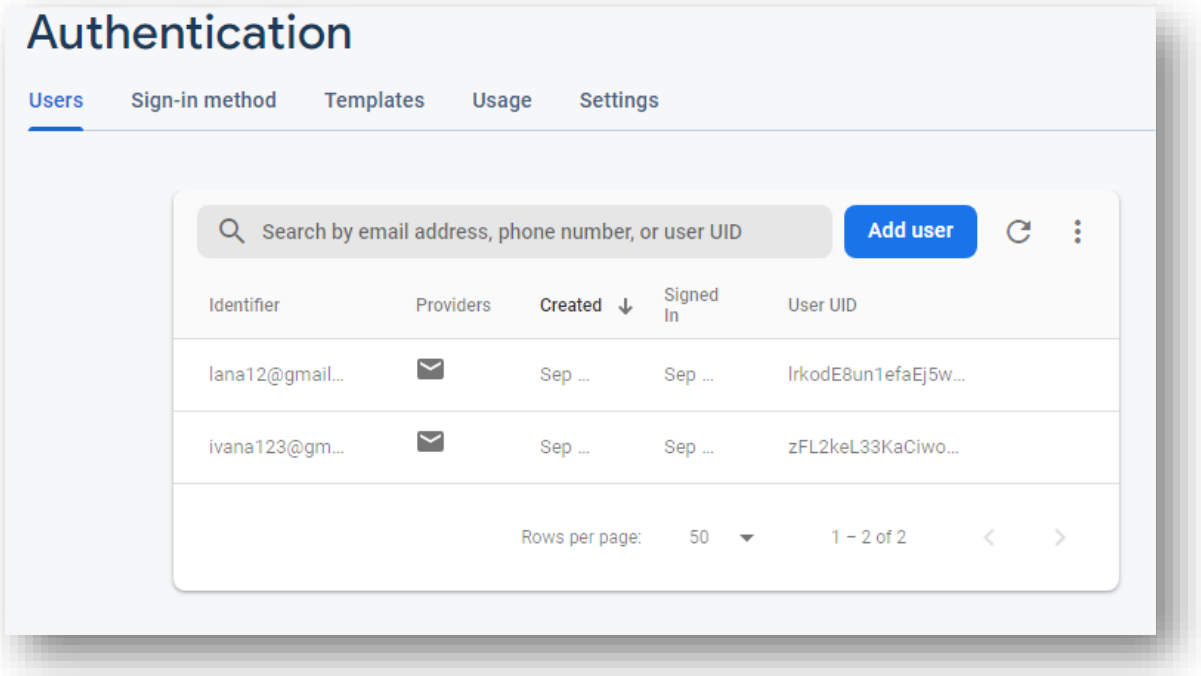
Slika 2: Kotlin VS Java (Robin Roy, 2021)

Razlike između Kotlina i Jave radi kojih programeri preferiraju Kotlin:

- Kotlin je sažet, štedi vrijeme koje bi korisnik inače potrošio na pisanje standardnog koda.
- Datoteke napisane u Javi mogu se pretvoriti u Kotlin samo pomoću skripte.
- Kotlin nema dodatnih troškova za vrijeme izvođenja.
- Kotlin pojednostavljuje asinkrono programiranje, asinkrono pozivanje mreže i baze podataka u Javi može biti nezgodno dok Kotlin za to ima korutine koje asinkrono programiranje čine jednostavnih i učinkovitim.
- Kotlin obrađuje Null-ove, Null u Javi može srušiti program dok se u Kotlinu to može spriječiti pomoću operatora varijablama.
- Kotlin može raditi na više platformi, može se pokrenuti gdje god se izvodi Java tako da se se može koristiti kod izrade višeplatformskih aplikacija.

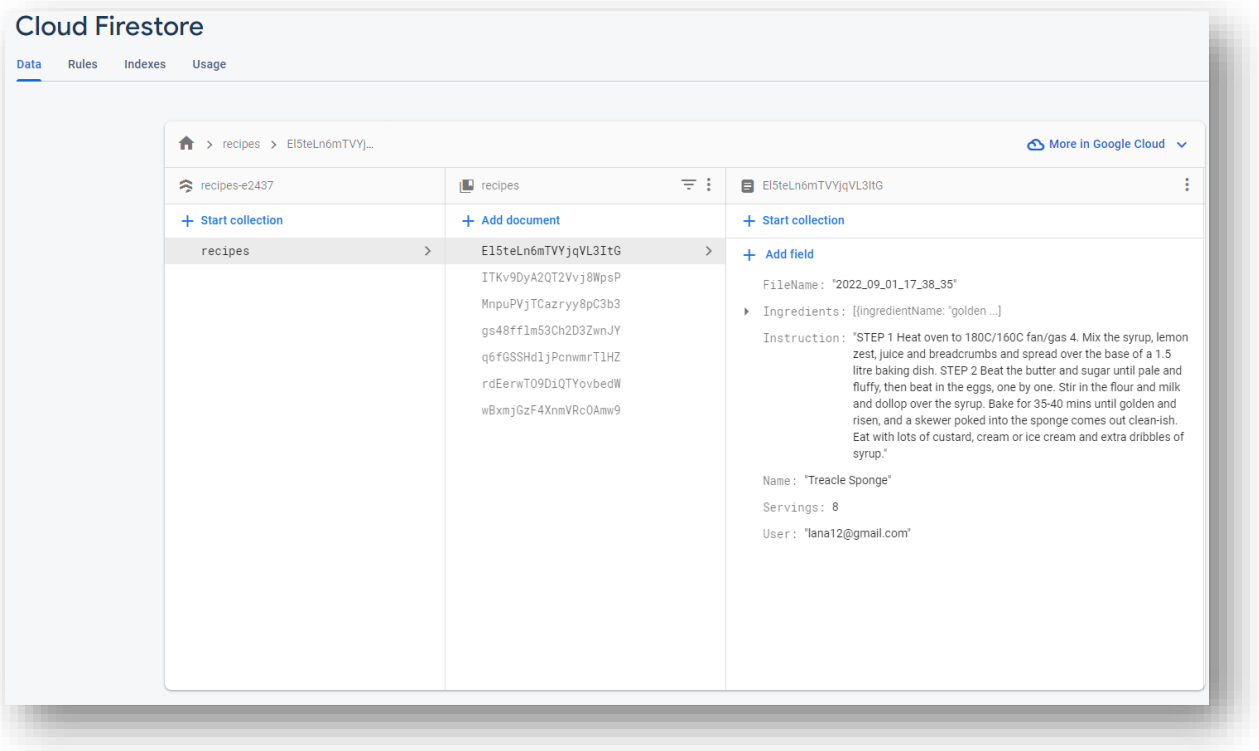
4. FIREBASE

Google-ov Firebase sastoji se od mnogo karakteristika koje ga čine idealnim alatom za mobilne i web aplikacije. Smanjuje radno opterećenje i vrijeme razvoja, idealan je za izradu prototipova, jednostavan, lagan za korištenje te industrijski prepoznat. Osnovan 2011. godine kao API za chat, kupljen od strane Google-a 2014. godine trenutno korišten za izradu dinamičnih web i mobilnih aplikacija. Korištenjem Firebase-a eliminira se potreba za upravljanjem bazom podataka od strane korisnika. Što se tiče sigurnosti, također ima ugrađena sigurnosna pravila koja ga čine pouzdanim rukovateljem podataka i poslužitelja. Neki od nedostataka korištenja Firebase-a jesu ako se njime ne upravlja pravilno, trošak održavanja Firebase-a na usluzi koja se plaća prema korištenju povećava se kako se čitanje i pisanje povećava. Podatke koji su stvoreni u Firebase-u teško je prebaciti u neku drugu bazu podataka. Više se posvećuje Android uređajima nego iOS. Za izradu ovog projekta korištena je Firebase autentifikacija. Firebase autentifikacija pruža metode za stvaranje i upravljanje korisnicima koji koriste email i lozinku za prijavu. Temelji se na tokenu i pruža gotove integracije s najčešćim pružateljima usluga kao što su Google, Facebook, Twitter i drugi.



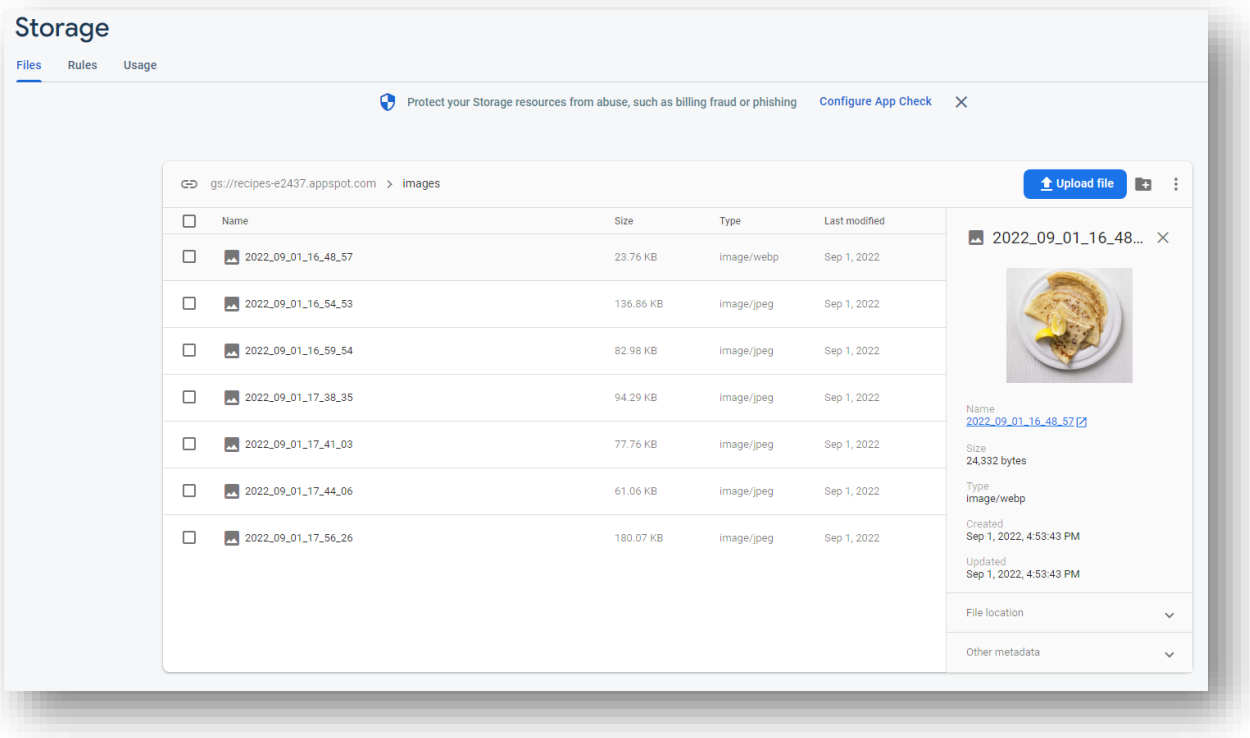
Slika 3: Firebase Authentication

Za pohranu podataka o receptu korišten je Firebase Firestore baza podataka. Firestore je NoSQL baza podataka koja poboljšava iskustvo i razvojni proces programerima. To je učinkovit alat za pohranu podataka. Dizajniran je za rad s bazom podatka u stvarnom vremenu. Firestore ima podatkovni model koji je fleksibilan i podržava hijerarhijsku strukturu podataka koja pohranjuje informacije u dokumentu koji se priprema u zbirku.



Slika 4: Firebase Cloud Firestore

Za pohranu fotografija korišten je Firebase Storage. Pošto je Firestore limitiran na veličinu dokumenta do 1MB za pohranu slika ili videa koristi se Firebase Storage.



Slika 5: Firebase Storage

5. RECYCLERVIEW

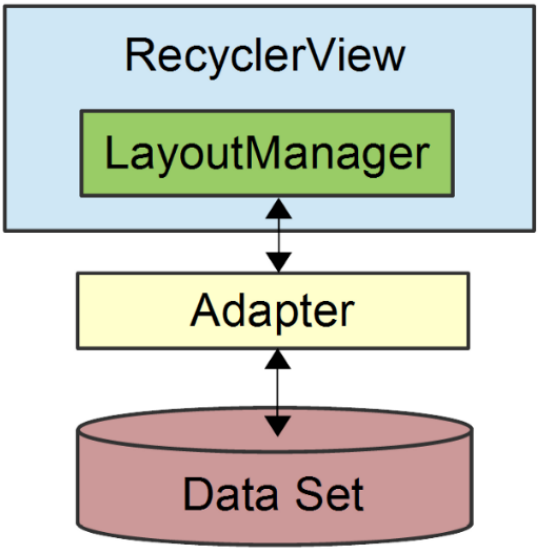
RecyclerView je potreban u skoro svakoj modernoj Android aplikaciji, aplikacije za fotografije, popis kontakata, glazbene aplikacije sa popisom pjesama, popis proizvoda ili u ovom slučaju popis recepata, gotovo svaka aplikacija koju korisnici imaju na svom pametnom telefonu koristi RecyclerView za prikaz nekog većeg skupa podataka ili listu podataka.

RecyclerView nastao je kao zamjena za već postojeći ListView. Problemi kod ListView-a koje RecyclerView rješava su zastajkivanje kod pomicanja liste, ListView kreira onoliko redaka koliko ima podatkovnih stavki u skupu podataka. ListView nema ugrađenu podršku za animaciju te je jedino moguće vertikalno pomicanje, horizontalno pomicanje nije dozvoljeno.

RecyclerView je komponenta korisničkog sučelja koja omogućuje stvaranje lista sa vertikalnim ili horizontalnim pomicanjem. Temelji se na adapteru. Adapter pruža vezanje skupa podataka aplikacije s prikazima stavki koje se prikazuju unutar RecyclerView-a. Adapter zna kako pridružiti svaku poziciju stavke u RecyclerView-u s određenom lokacijom u izvoru podataka.

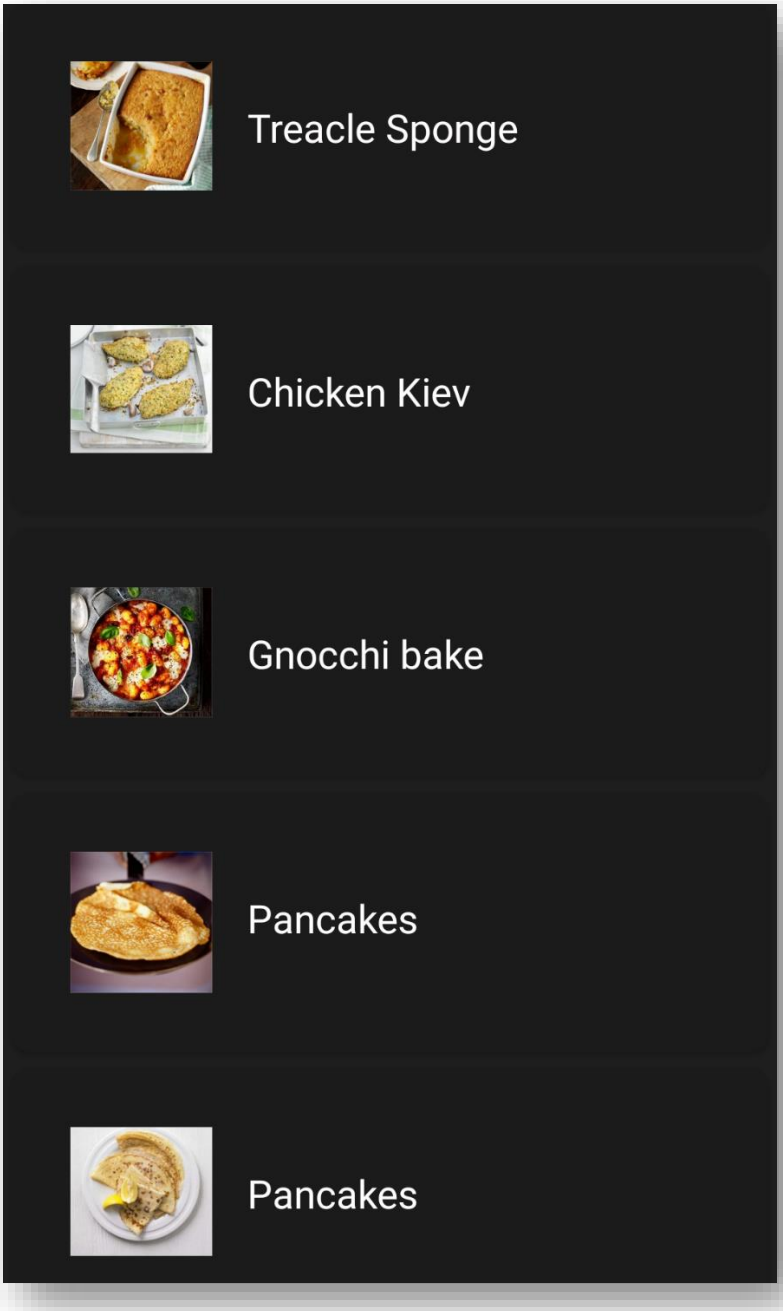
Layout Manager pozicionira stavke unutar RecyclerView-a. Može se koristiti jedan ili nekoliko unaprijed definiranih Layout Manager-a ili implementirati vlastiti.

Uz RecyclerView obavezno je koristiti i View Holder, on pomaže nacrtati korisničko sučelje za svaku pojedinačnu stavku koja će biti prikazana na ekranu.

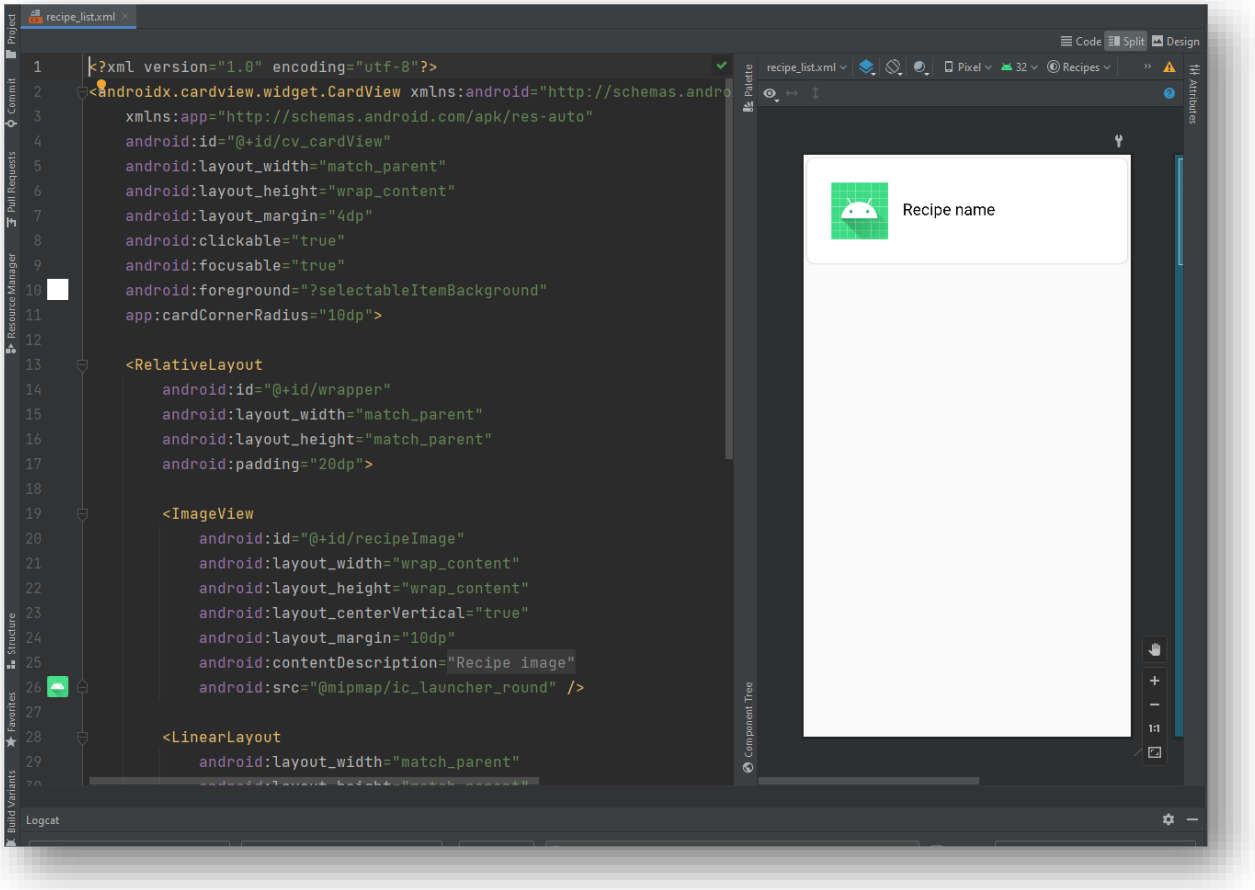


Slika 6: Odnos između RecyclerViewa, LayoutManagera i Adaptera (Microsoft docs, 2021)

U aplikaciji Cook Nook RecyclerView je korišten na više mjesta, na primjeru sa slike 6 RecyclerView se koristi za prikaz recepata. Potrebno je napraviti izgled jednog reda koji se prikazuje u RecyclerView-u koji se onda ponavlja za svaku stavku. Na slici 7 prikazan je primjer izgleda jedne ćelije. U cardview-u prikazana je slika recepta te sa desne strane naziv recepta.



Slika 7: RecyclerView



Slika 8: Izgled jednog reda unutar RecyclerView-a

Glavni dio rada RecyclerView-a odvija se u adapteru. Adapter pristupa izvoru podataka i popunjava stavke sadržajem unutar RecyclerView-a. Adapter učitava svaki red unutar RecyclerView-a s podacima za određenu stavku retka. Na primjer, za poziciju retka P, adapter locira povezane podatke na poziciji P unutar izvora podataka i kopira te podatke u stavku retka na poziciji P u kolekciji RecyclerView.

Kod implementiranja adaptera potrebno je nadjačati (override) metode OnCreateViewHolder koja instancira datoteku izgleda stavke i držač prikaza. OnBindViewHolder učitava podatke na navedenoj poziciji u prikaze čije su reference pohranjene u danom držaču pogleda i ItemCount vraća broj stavki u izvoru podataka.

Slika 9: Adapter

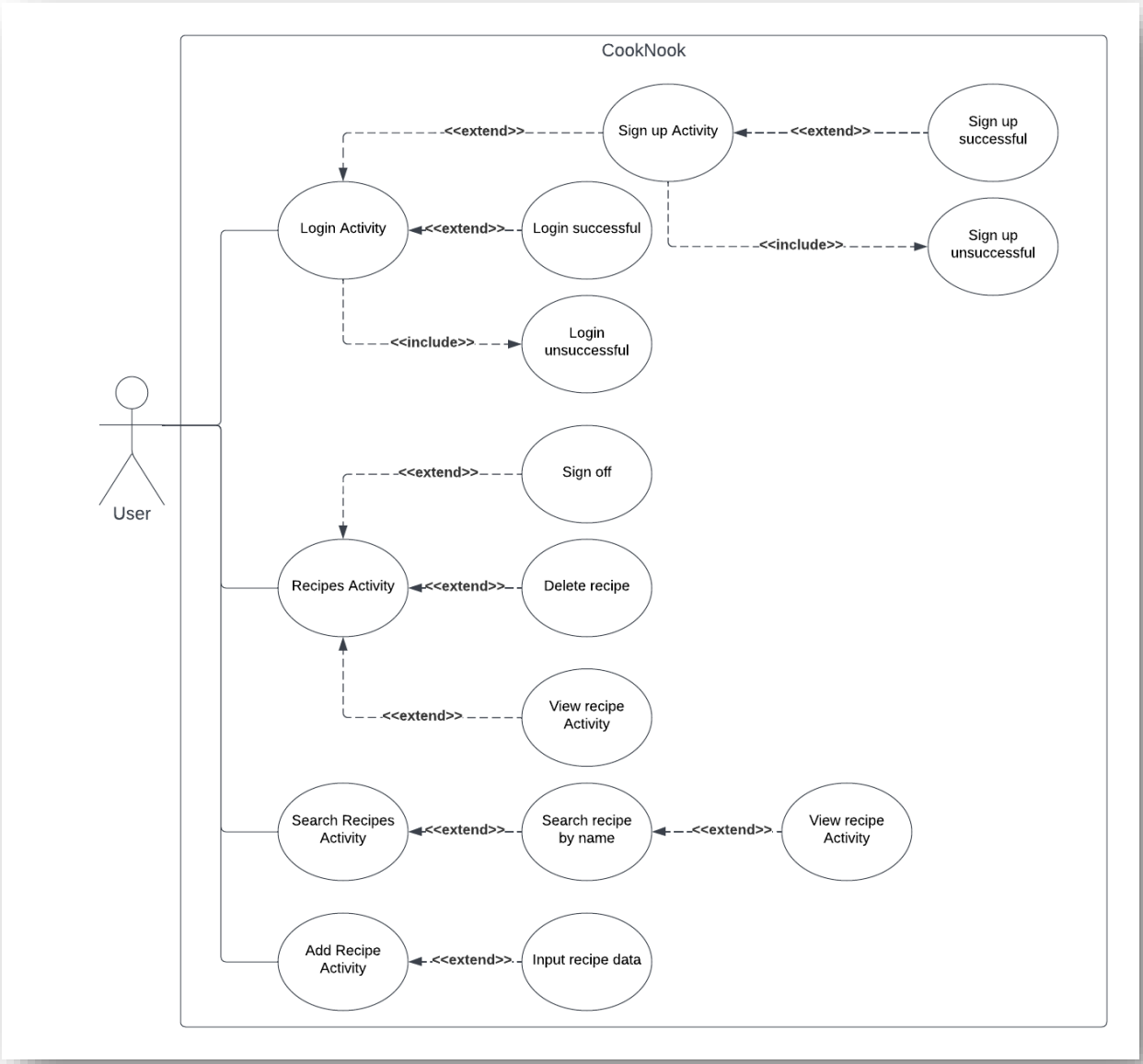
6. DIJAGRAM SLUČAJA

Iz dijagrama slučaja sa slike vide se funkcionalnosti ove aplikacije. Korisnik se može prijaviti na svoj račun te ako upiše ispravan email i lozinku prijava je uspješna. Korisnik koji još nema napravljen račun to može napraviti preko stranice za registraciju.

Nakon prijave postojećeg ili registracije novog korisnika otvara se stranica Recipes Activity na kojoj korisnik može vidjeti vlastite recepte ako ih ima. Na ovoj stranici odnosno aktivnosti korisnik se može odjaviti sa svog računa, obrisati recepte, i otvoriti pregled recepta odnosno View recipe Activity gdje je recept detaljnije opisan.

Korisnik ima opciju pretraživanja svih recepata unesenih od drugih korisnika po nazivu recepta na aktivnosti Search recipes te klikom na željeni recept otvara View recipe Activity.

Korisnik dodaje vlastite recepte preko Add Recipe Activity gdje je potrebno unijeti podatke o receptu kao što su ime recepta, porcije, instrukcije kako se pravi recept, sastojke i sliku recepta.



Slika 10: Dijagram slučaja

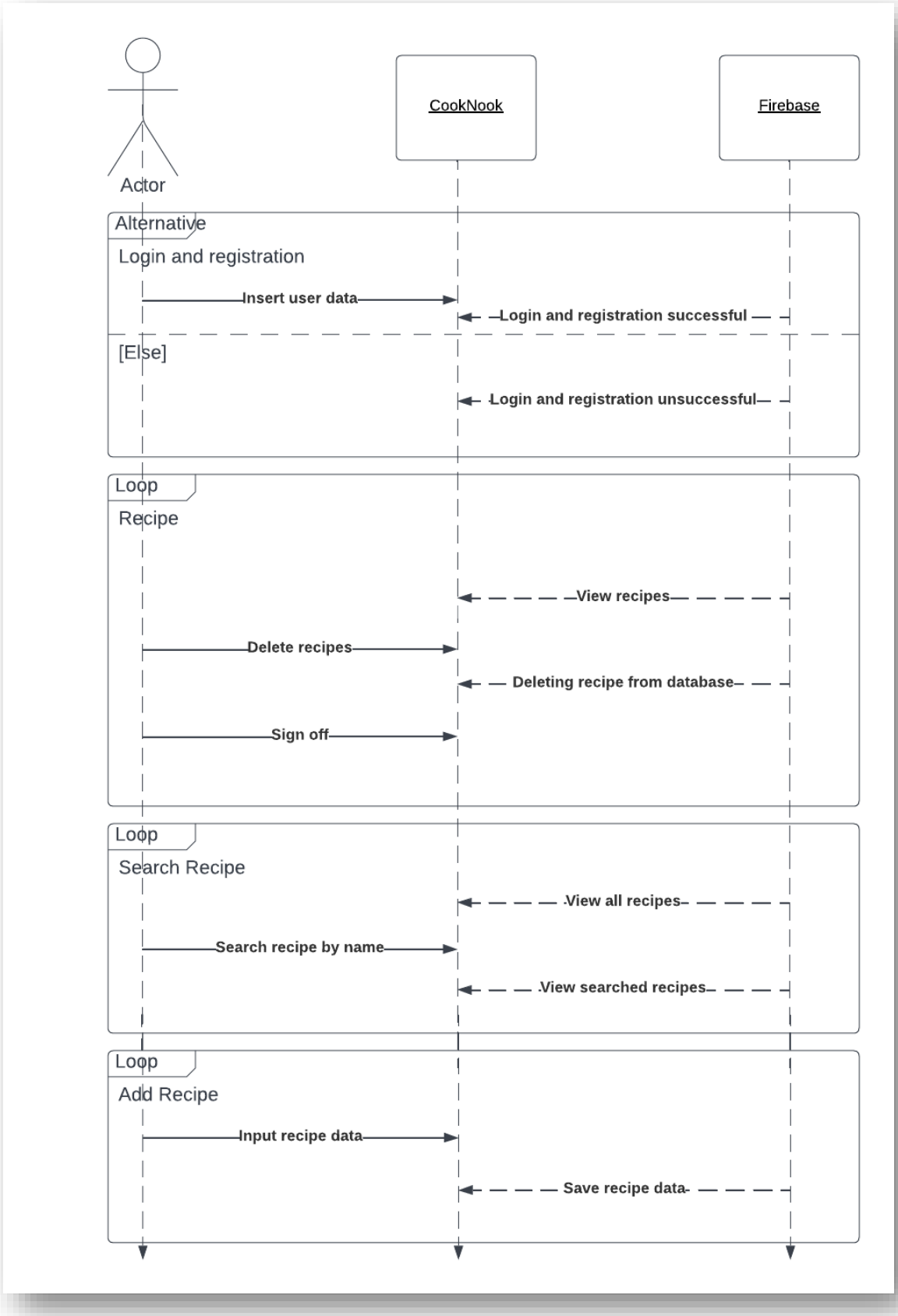
7. DIJAGRAM SLIJEDA

Na dijagramu slijeda prikazan je odnos između aplikacije i baze podataka. Za bazu podataka korišten je Firebase. Nakon što korisnik unese svoje podatke za prijavu ili registraciju podatci se provjeravaju u bazi podataka. Za prijavu i registraciju korišten je Firebase Authentication gdje se spremaju email i lozinka korisnika. Ako su uneseni podatci ispravni korisnik je uspješno prijavljen na svoj račun, ako podatci nisu ispravni korisnik dobiva poruku upozorenja te ponovno treba unijeti podatke.

Pri otvaranju aktivnosti Recipe Activity korisniku se iz baze podataka ispišu njegovi recepti ako ih ima. Korisnik ih zatim može obrisati iz baze podataka. Za spremanje podataka o receptu korišten je Firebase Firestore a za spremanje slike recepta koristi se Firebase Storage. Na ovoj aktivnosti korisnik također ima opciju da se odjavi sa svog računa.

Na Aktivnosti Search Recipe prikazani su svi recepti koji su dostupni u bazi podataka korisnik može pretražiti bazu podataka unoseći naziv recepta kojeg želi pronaći. Nakon što se korisniku ispišu svi postojeći recepti pod traženim nazivom korisnik može otvoriti svaki od njih kako bi detaljnije pogledao upute o receptu.

Kako bi korisnik dodao vlastite recepte na Add Recipe aktivnosti treba unijeti podatke o receptu koji se zatim spremaju u bazu podataka.



Slika 11: Dijagram slijeda

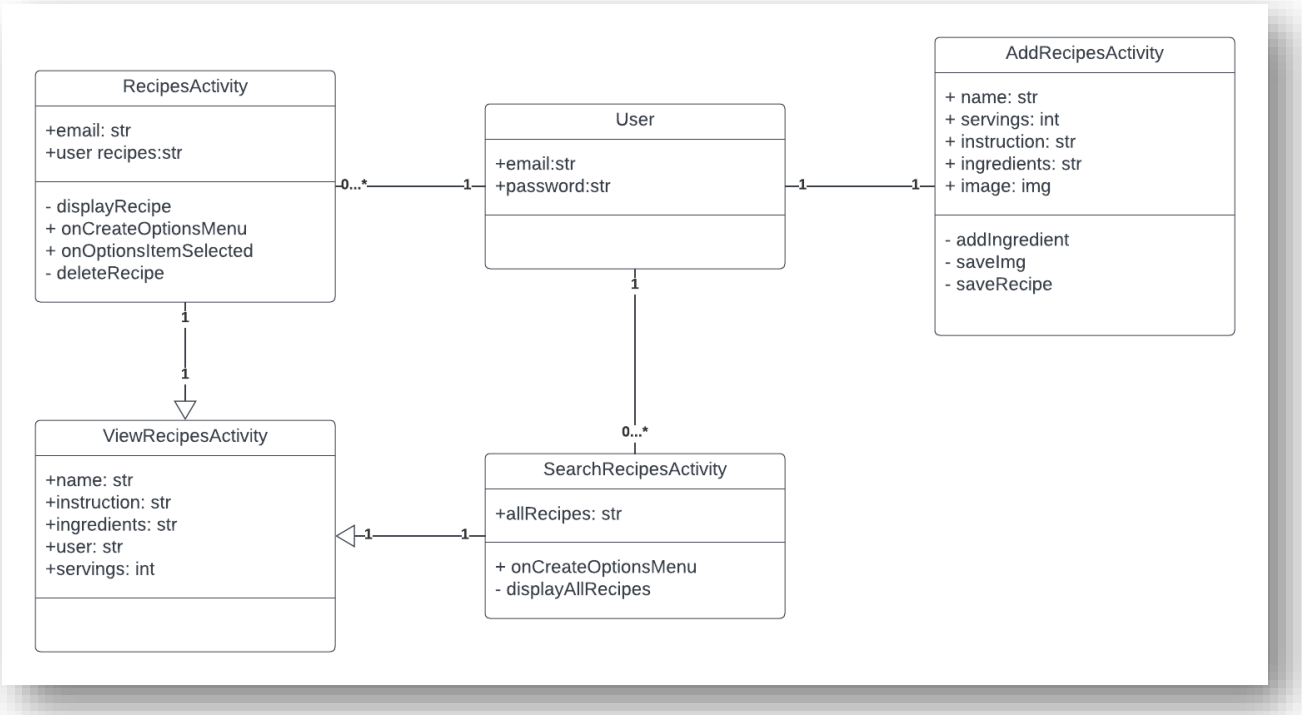
8. KLASNI DIJAGRAM

Iz klasnog dijagrama sa slike 7 mogu se vidjeti odnosi između klasa te funkcionalnosti koje aplikacija ima. Jedan korisnik User ima svoj email i lozinku s kojom otvara svoj profil u aplikaciji.

Na RecipesActivity jedan korisnik može vidjeti svoje recepte, ako korisnik još nema vlastite recepte neće biti prikazani, a ako ima bit će svi prikazani. Na ovoj aktivnosti korisniku se prikažu recepti, te ih može obrisati. Preko funkcija onCreateOptionsMenu i onOptionsItemSelected korisnik preko menija može kliknuti na dugme za odjavu. Klikom na svaki od recepta otvara se pregled odabranog recepta. Klikom na jedan recept otvara se prikaz jednog recepta.

Na SearchRecipesActivity korisnik ima pregled svih recepata od svih korisnika te pomoću onCreateOptionsMenu ima mogućnost pretraživanja recepta po nazivu. Također se klikom na recept otvara pregled istog.

Kako bi korisnik dodao vlastiti recept to može napraviti preko AddRecipesActivity gdje korisnik unosi naziv, broj porcija, instrukcije odnosno metodu pripreme recepta, sastojke te sliku recepta. Jedan korisnik unosi jedan po jedan recept.



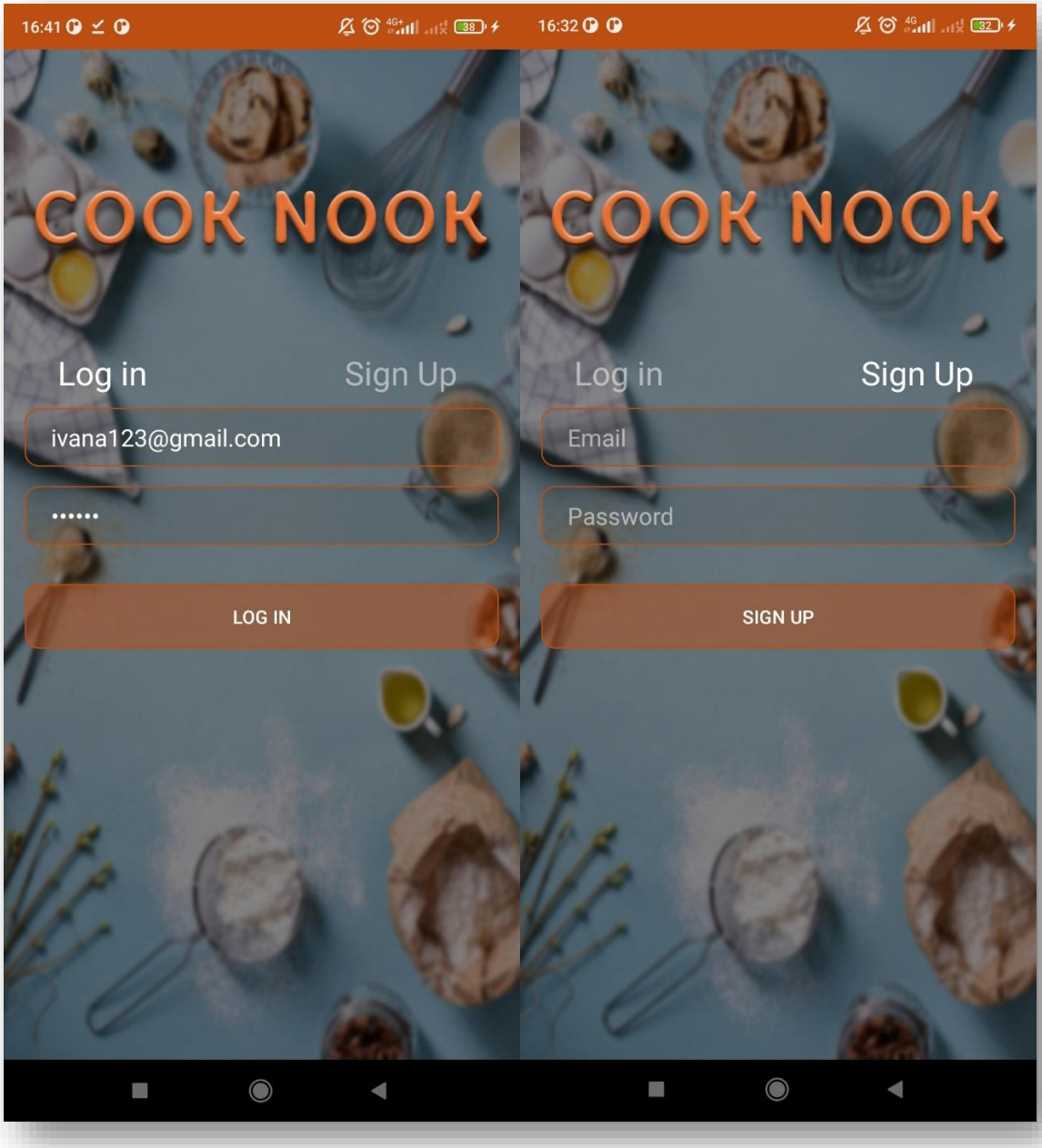
Slika 12: Klasni dijagram

9. IMPLEMENTACIJA

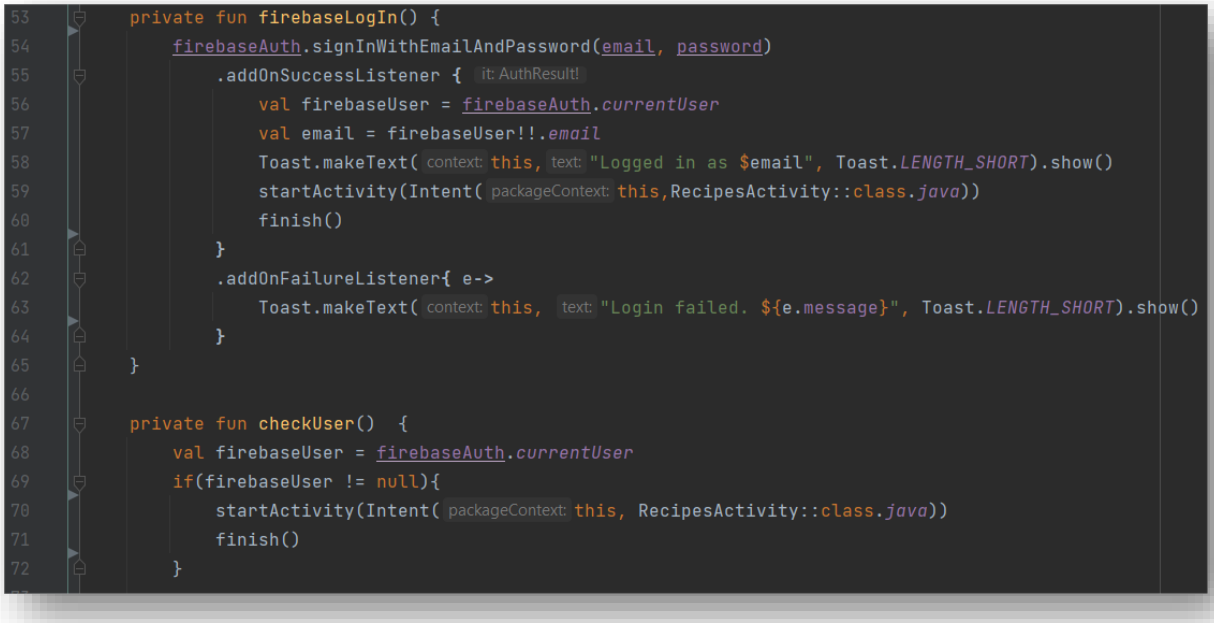
9.1. PRIJAVA I REGISTRACIJA KORISNIKA

Kako bi korisnik mogao koristiti aplikaciju potrebno je da ima napravljen račun. Postojeći korisnici unose svoje podatke na stranici za prijavu a budući korisnici unose podatke na stranici za registraciju. Na vrhu aplikacije stoji logo i naziv aplikacije Cook Nook , naziv znači kutak za kuhanje.

U psihologiji boja poznato je da narančaste i crvene nijanse izazivaju glad kod čovjeka tako da u ovoj aplikaciji za recepte prevladava narančasta boja. Korisnik unosi svoj email i lozinku te ako su uneseni podatci ispravni korisnik se uspješno prijavljuje u aplikaciju. Za prijavu i registraciju korišten je Firebase Authentication preko emaila i lozinke. Na primjeru sa slike 8 korisnik se prijavio u aplikaciju koristeći email ivana123@gmail.com. Email i lozinka moraju odgovarati po pravilima Firebase autentifikacije kako bi bile ispravne te kako bi se prijava ili registracija ostvarile.



Slika 13: Prijava i registracija

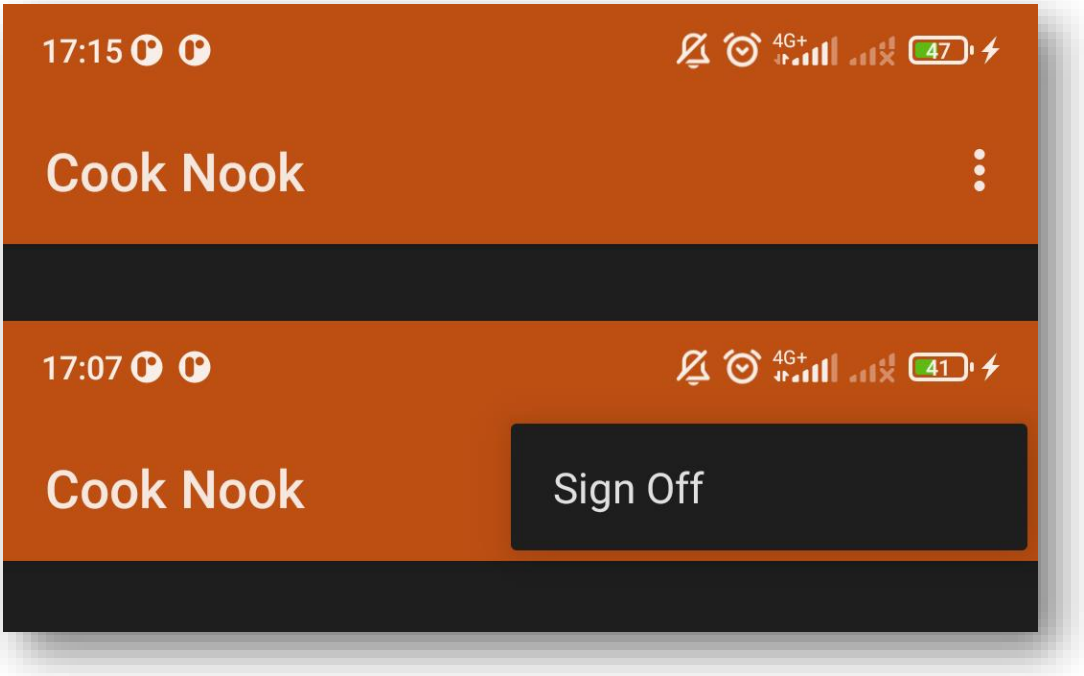


```
53    private fun firebaseLogIn() {
54        firebaseAuth.signInWithEmailAndPassword(email, password)
55            .addOnSuccessListener {  it: AuthResult!
56                val firebaseUser = firebaseAuth.currentUser
57                val email = firebaseUser!!.email
58                Toast.makeText( context: this, text: "Logged in as $email", Toast.LENGTH_SHORT).show()
59                startActivity(Intent( packageContext: this,RecipesActivity::class.java))
60                finish()
61            }
62            .addOnFailureListener{ e->
63                Toast.makeText( context: this,  text: "Login failed. ${e.message}", Toast.LENGTH_SHORT).show()
64            }
65    }
66
67    private fun checkUser()  {
68        val firebaseUser = firebaseAuth.currentUser
69        if(firebaseUser != null){
70            startActivity(Intent( packageContext: this, RecipesActivity::class.java))
71            finish()
72        }
```

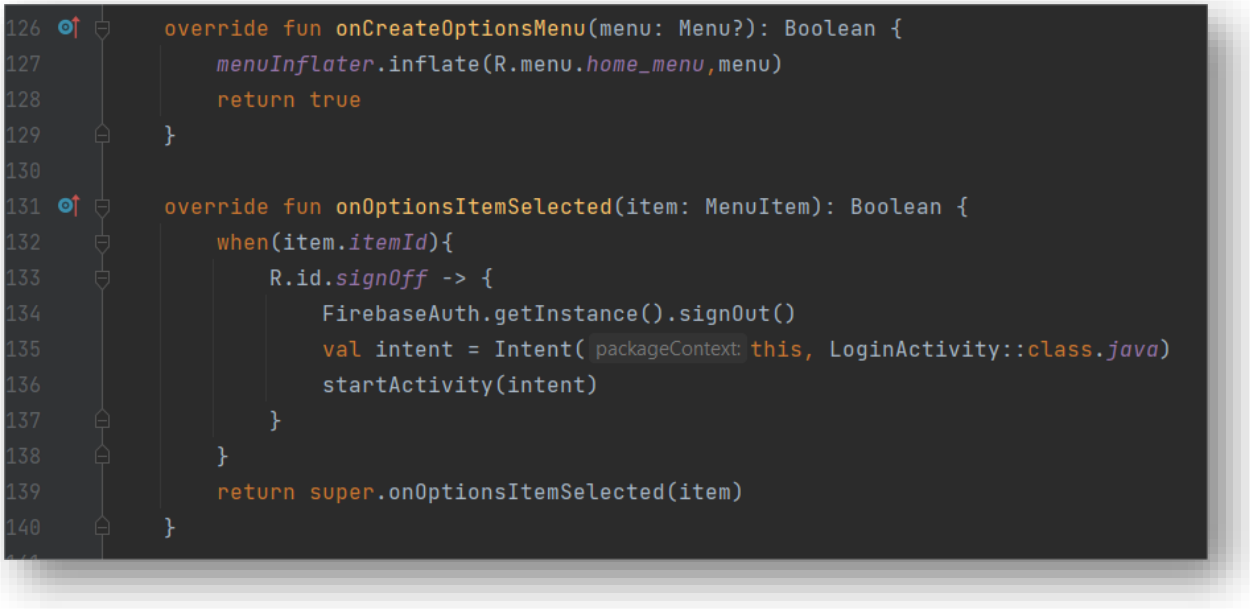
Slika 14: Kod za prijavu

9.2. AKTIVNOST RECIPES (RECEPTI)

Nakon uspješne prijave korisnika otvara se početna aktivnost na kojoj korisnik može vidjeti vlastite recepte. Na ovoj aktivnosti u gornjem lijevom kutu vidimo naziv aplikacije a u desnom kutu korisnik ima opciju odjave s prijavljenog računa klikom na ikonu za meni prikaže se dugme za odjavu. Odjava se radi preko Firebase Autentifikacije. Pomoću funkcija onCreateOptionsMenu i onOptionsItemSelected otvara se meni u ovom slučaju samo se nalazi dugme za odjavu u njemu te se klikom na njega odjavi korisnika pomoću Firebase autentifikacije.

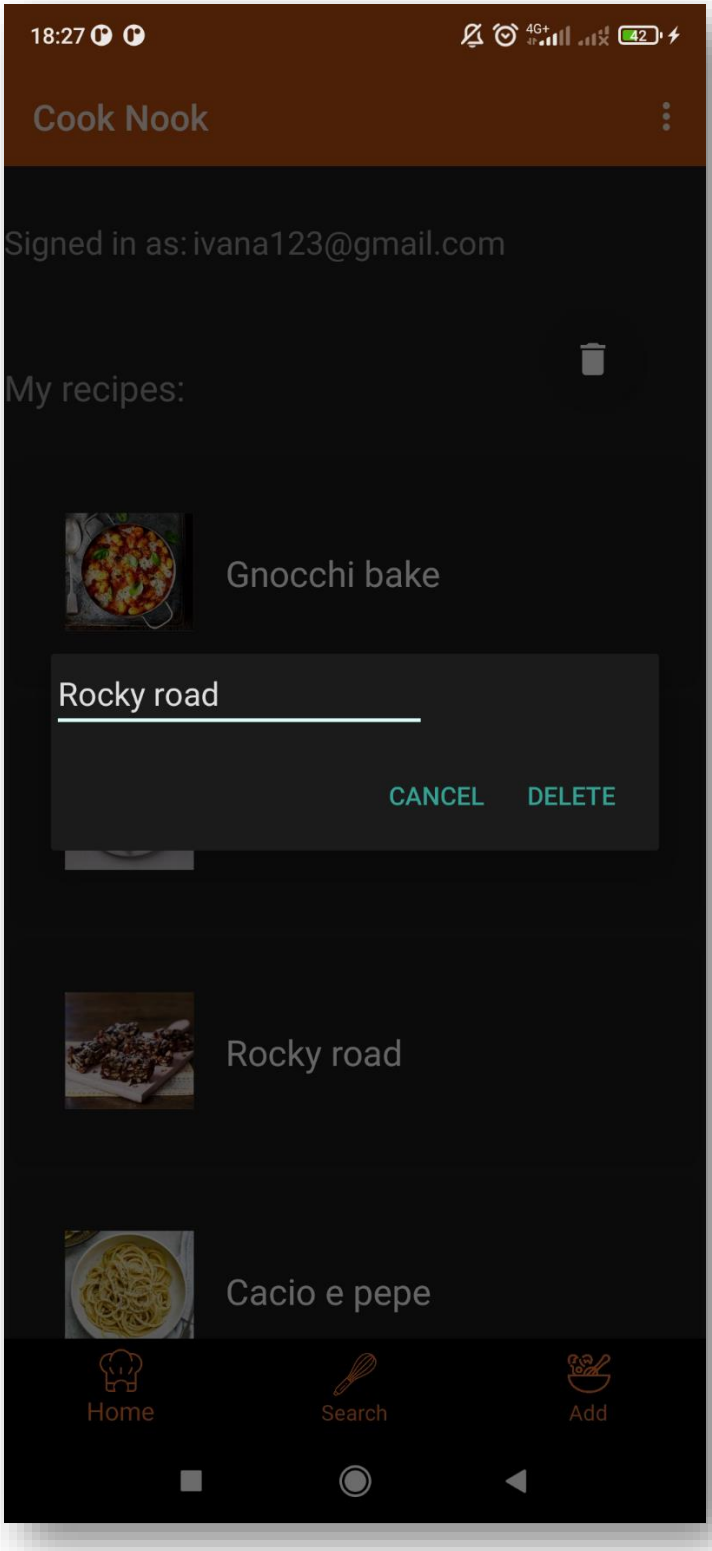


Slika 15: Odjava korisnika

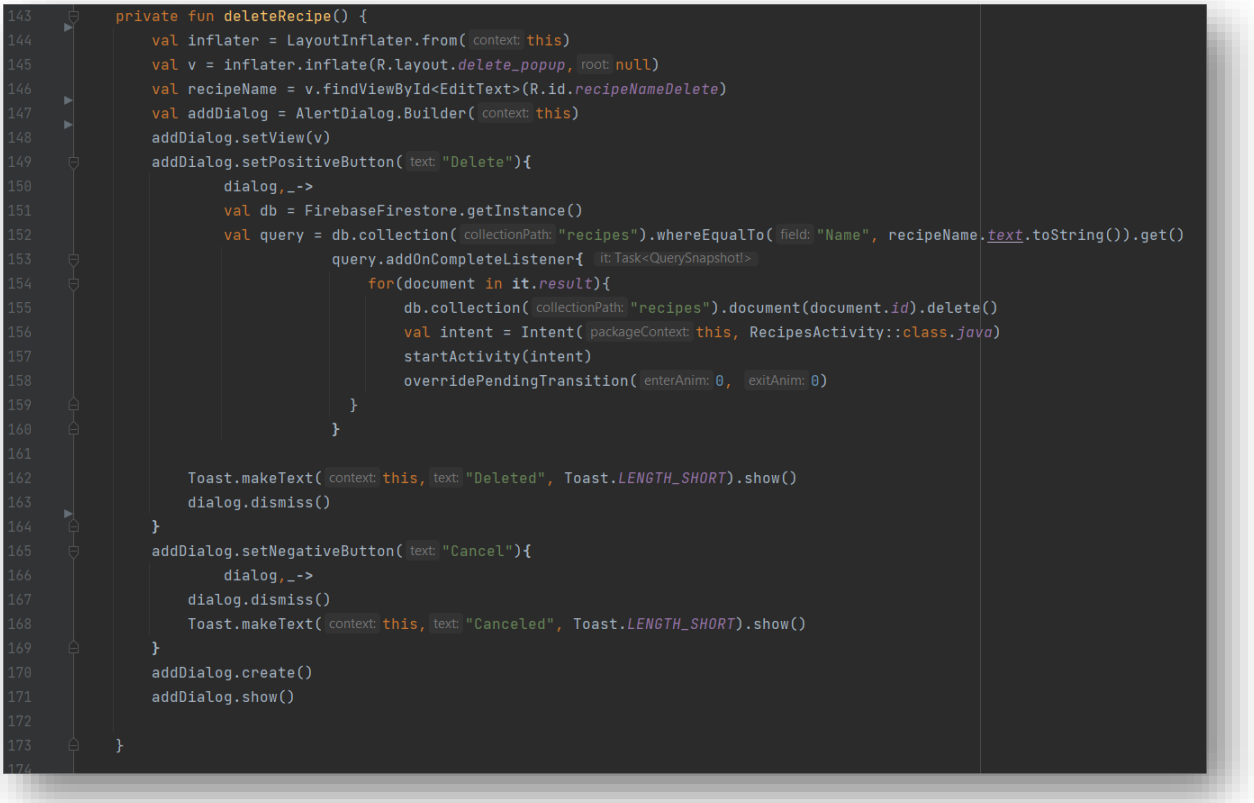


Slika 16: Kod za odjavu korisnika

Ispisan je i email trenutno prijavljenog korisnika te ispod toga pomoću recyclerview-a ispisani su svi recepti trenutno prijavljenog korisnika. Klikom na ikonicu koša za smeće korisnik može upisati ime recepta kojeg želi obrisati te klikom na obriši recept se obriše iz baze podataka. Klikom na koš otvara se skočni prozor layout-a delete_popup, korisnik upiše ime recepta te klikom na delete isti se izbriše iz baze. U primjeru sa slike 12 korisnik upisuje naziv recepta kojeg želi obrisati te tako briše svoj recept pod nazivom Rocky road.



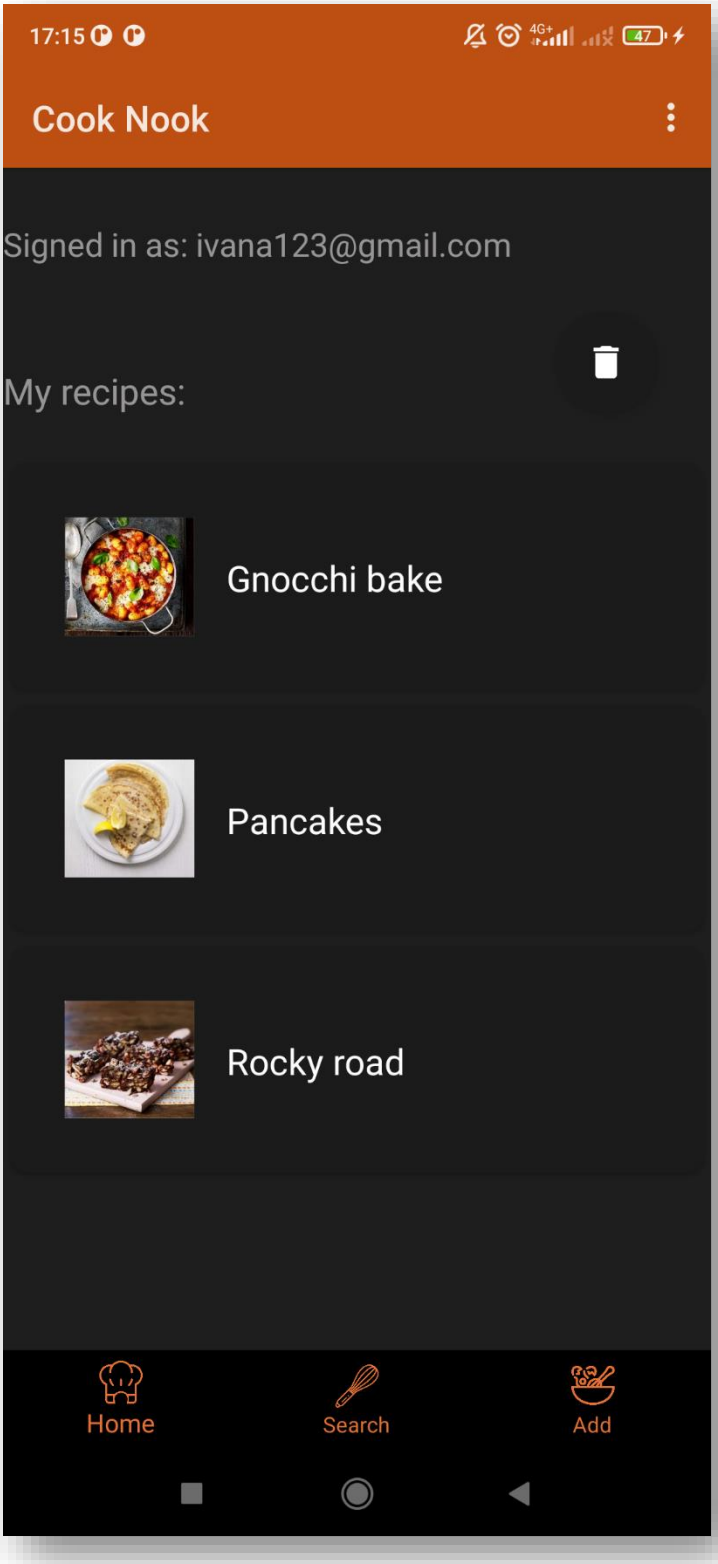
Slika 17: Brisanje recepta



```
143    private fun deleteRecipe() {
144        val inflater = LayoutInflater.from( context: this)
145        val v = inflater.inflate(R.layout.delete_popup, root: null)
146        val recipeName = v.findViewById<EditText>(R.id.recipeNameDelete)
147        val addDialog = AlertDialog.Builder( context: this)
148        addDialog.setView(v)
149        addDialog.setPositiveButton( text: "Delete"){
150                dialog,_->
151                val db = FirebaseFirestore.getInstance()
152                val query = db.collection( collectionPath: "recipes").whereEqualTo( field: "Name", recipeName.text.toString()).get()
153                            query.addOnCompleteListener{  it: Task<QuerySnapshot!>
154                                for(document in it.result){
155                                    db.collection( collectionPath: "recipes").document(document.id).delete()
156                                    val intent = Intent( packageContext: this, RecipesActivity::class.java)
157                                    startActivity(intent)
158                                    overridePendingTransition( enterAnim: 0,  exitAnim: 0)
159                                }
160                            }
161
162            Toast.makeText( context: this, text: "Deleted", Toast.LENGTH_SHORT).show()
163            dialog.dismiss()
164        }
165        addDialog.setNegativeButton( text: "Cancel"){
166                dialog,_->
167            dialog.dismiss()
168            Toast.makeText( context: this, text: "Canceled", Toast.LENGTH_SHORT).show()
169        }
170        addDialog.create()
171        addDialog.show()
172
173    }
174
```

Slika 18: Kod za brisanje recepta

Unutar recyclerview-a prikazana je slika i naziv recepta, korisnik može kliknuti na svaki recept te se klikom na njega otvara se pregled recepta na kojem može pogledati sastojke i metodu pripreme recepta.

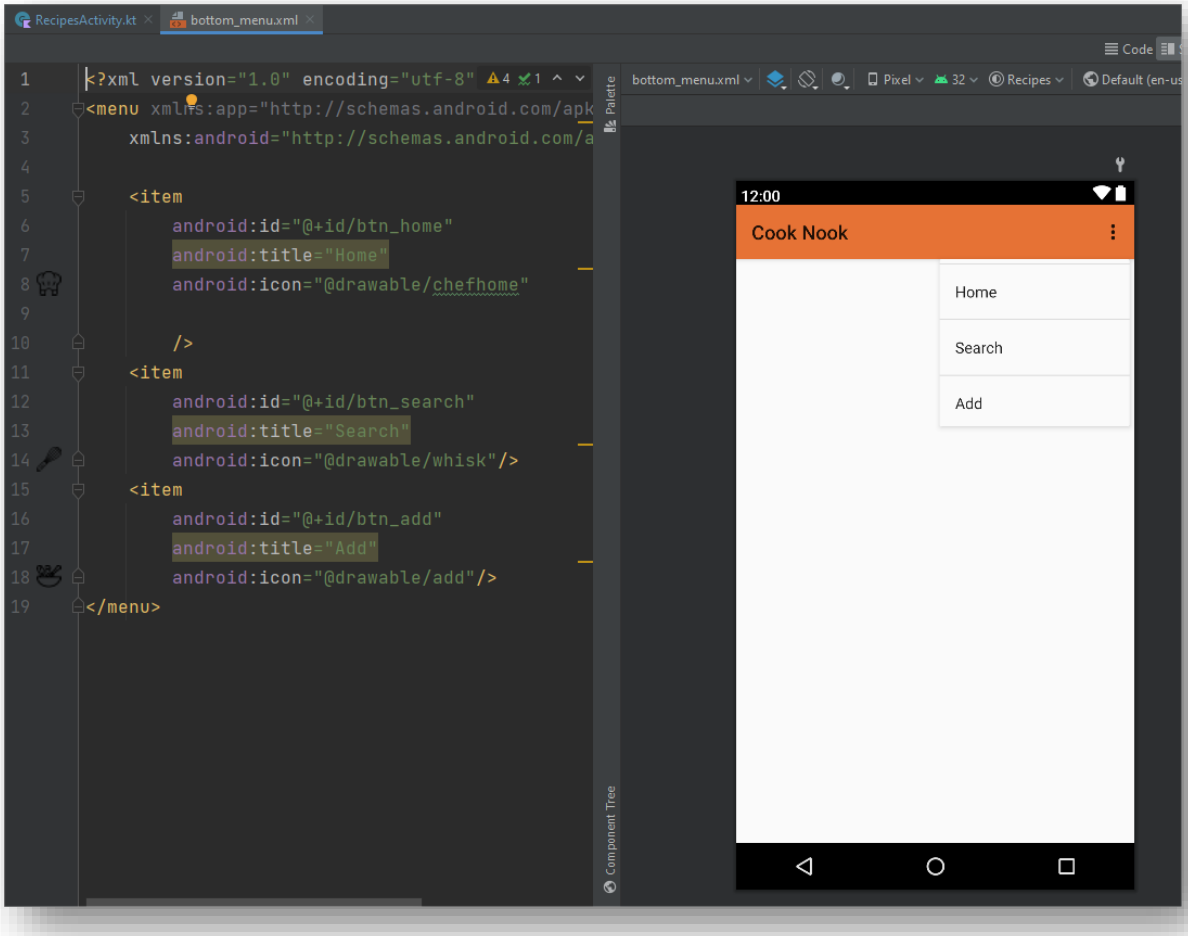


Slika 19: Aktivnost Recipes i donja navigacija

Na dnu aplikacije nalazi se donja navigacija koja korisnika vodi na ostale aktivnosti. Home je trenutna stranica, Search korisnika vodi na aktivnost za pretraživanje svih recepata, a Add korisnika vodi na aktivnost za dodavanje novog recepta u bazu. Donja navigacija je poput odjave napravljena pomoću menija, u ovom slučaju korisnik ima 3 opcije koje vode na različite aktivnosti te svaka ima ikonu i naziv.

```
60          binding.bottomNav.setOnItemSelectedListener { item: MenuItem ->
61              when (item.itemId) {
62                   R.id.btn_home -> {
63                      val intent = Intent( packageContext: this, RecipesActivity::class.java)
64                      startActivity(intent)
65                      overridePendingTransition( enterAnim: 0,  exitAnim: 0)
66                      true  ^setOnItemSelectedListener
67                  }
68                  R.id.btn_search -> {
69                      val intent = Intent( packageContext: this, SearchRecipesActivity::class.java)
70                      startActivity(intent)
71                      overridePendingTransition( enterAnim: 0,  exitAnim: 0)
72                      true  ^setOnItemSelectedListener
73                  }
74                  R.id.btn_add -> {
75                      val intent = Intent( packageContext: this, AddRecipeActivity::class.java)
76                      startActivity(intent)
77                      overridePendingTransition( enterAnim: 0,  exitAnim: 0)
78                      true  ^setOnItemSelectedListener
79                  }
80                  else ->
81                      true  ^setOnItemSelectedListener
82              }
83      }
84      }
```

Slika 20: Kod za donju navigaciju

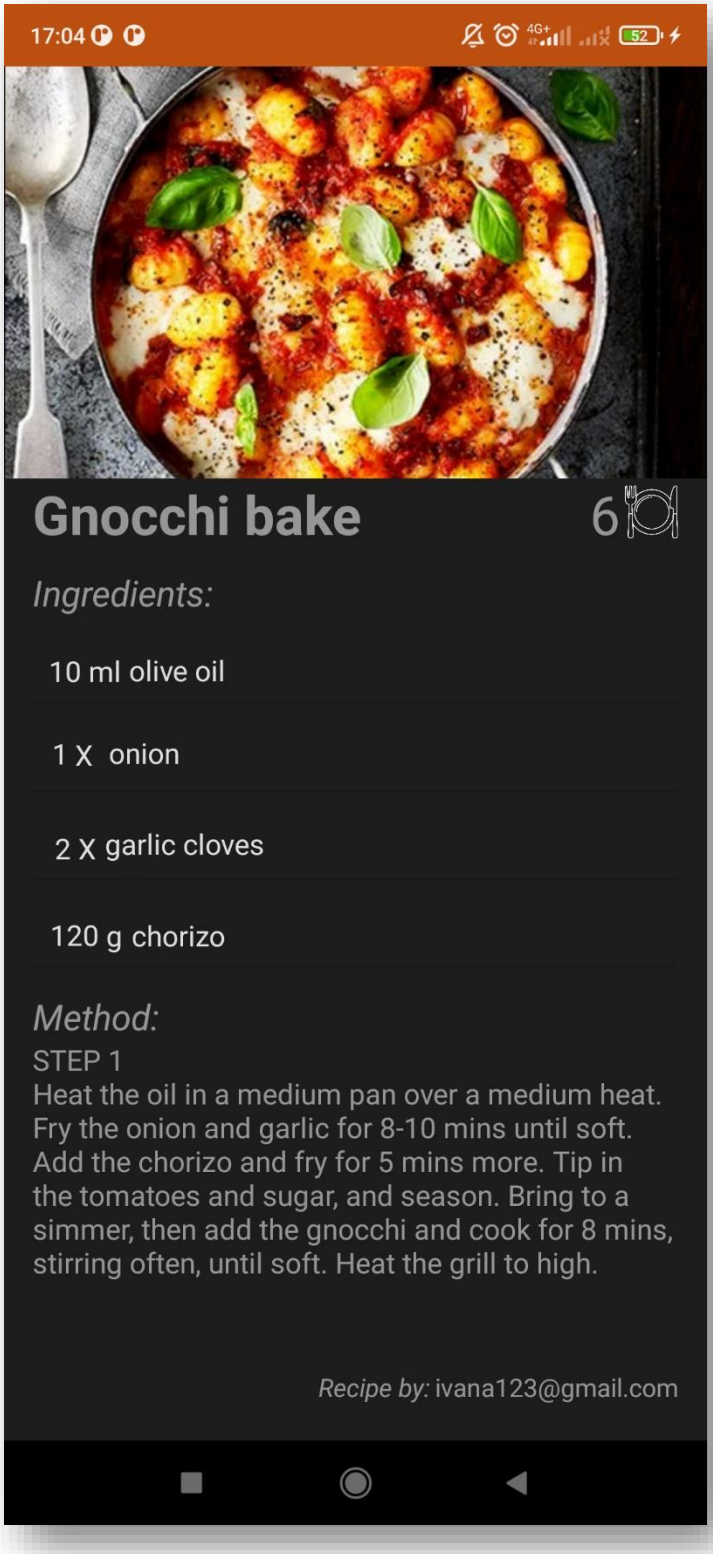


Slika 21: XML kod za donju navigaciju

9.3. AKTIVNOST VIEW RECIPE (PREGLED RECEPTA)

Klikom na recept prikazan u recyclerview-u otvara se aktivnost za pregled recepta, ViewRecipeActivity. Unutar ove aktivnosti korisnik ima detaljan pregled recepta, prikazana je slika recepta, naziv, količina porcija dobivenih iz recepta, sastojci, metoda pravljenja recepta je prikazan email korisnika koji je taj recept dodao u bazu.

Na primjeru sa slike 22 prikazan je prikaz recepta za Gnocchi bake. Na samom vrhu prikazana je slika recepta, ispod slike naziv recepta te desno od naziva broj porcija dobivenih pravljenjem tog recepta, broj porcija u ovom slučaju je 6. Lista sastojaka koje se mogu pomicati vertikalno kao i metoda pripreme koja se također može pomicati vertikalno. Na samom dnu nalazi se email korisnika koji je vlasnik prikazanog recepta.

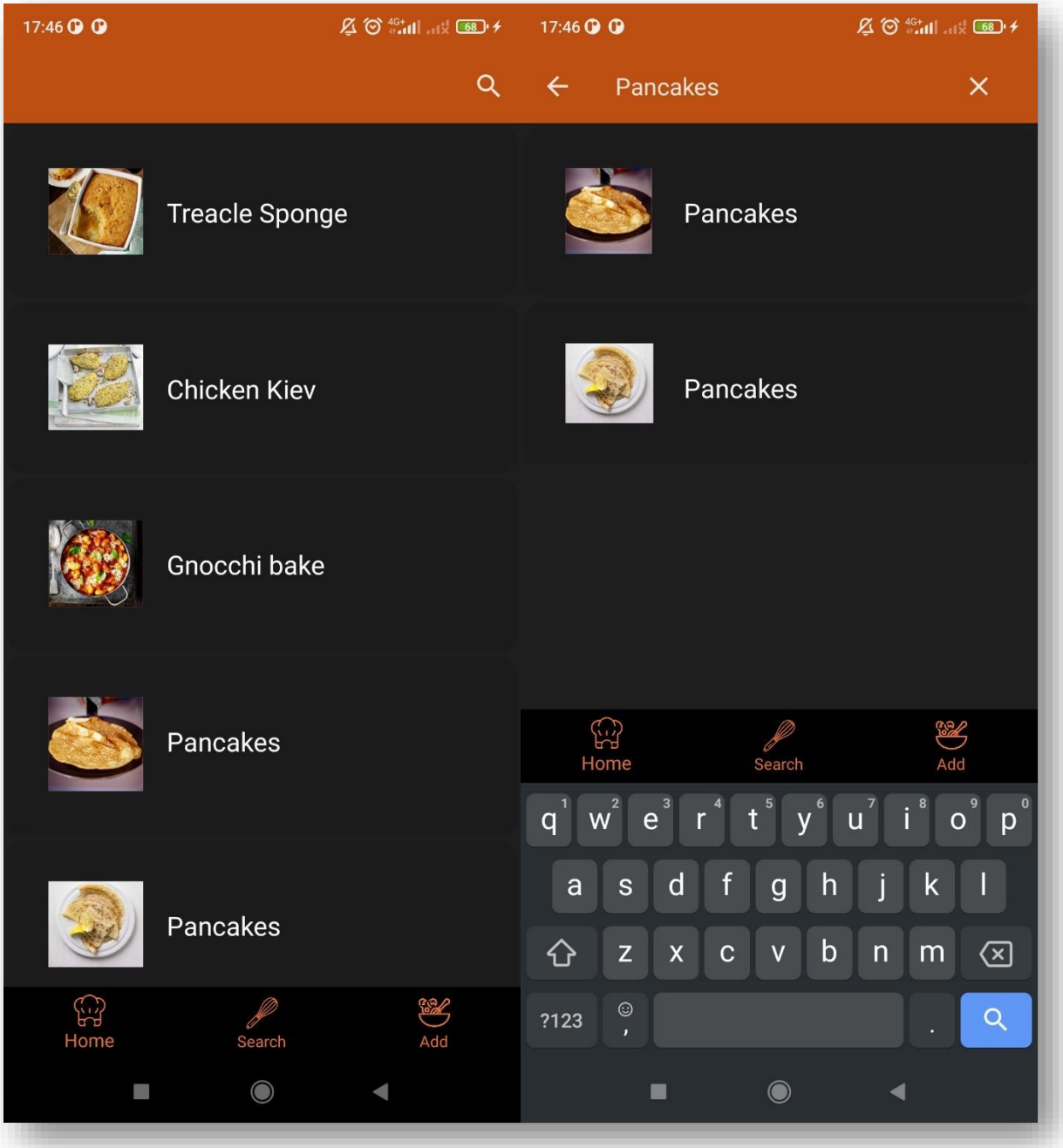


Slika 22: Pregled recepta

9.4. AKTIVNOST SEARCH RECIPE (PRETRAŽIVANJE RECEPTA)

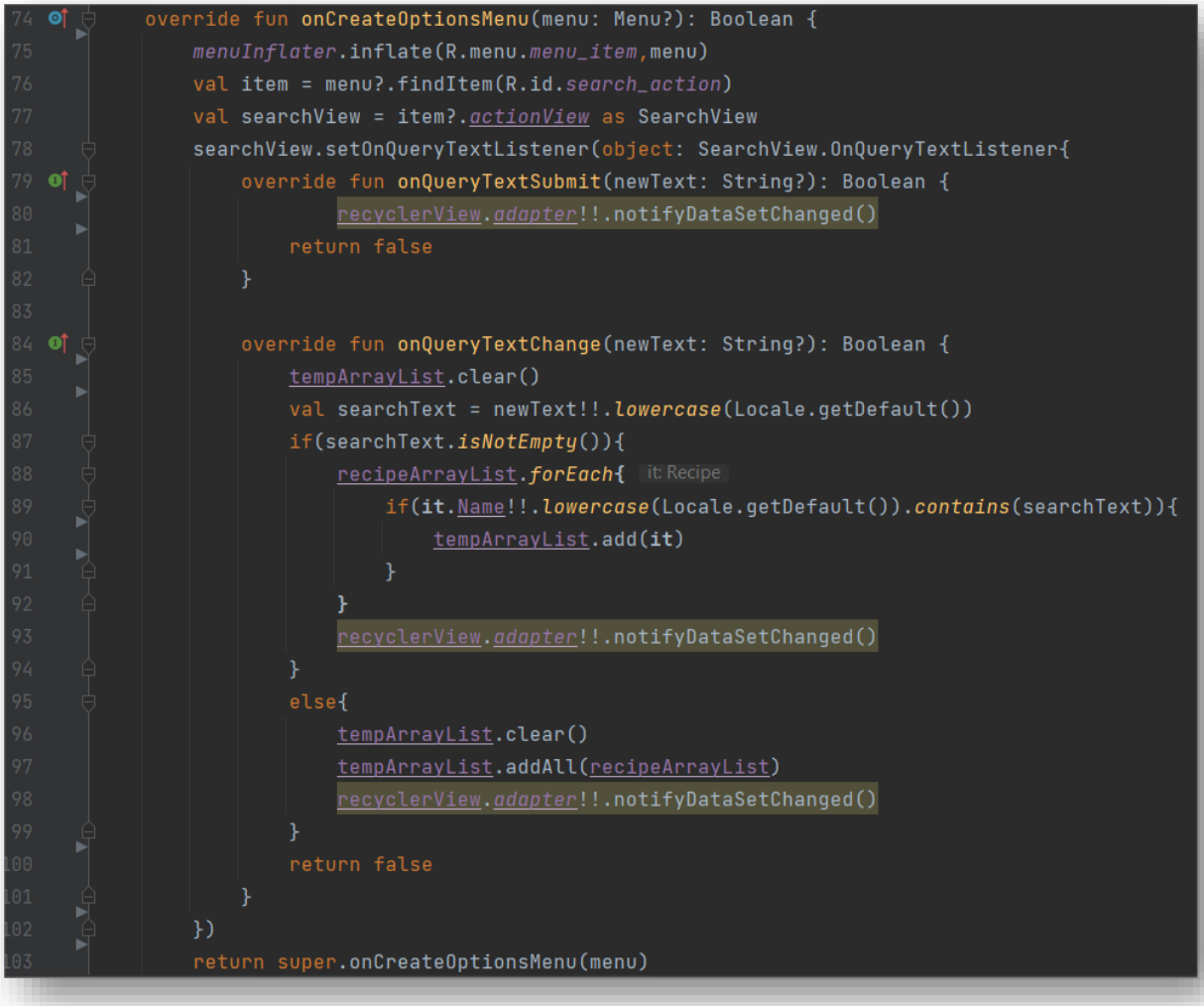
Na ovoj aktivnosti korisnik ima pogled na svoje recepte ali i na recepte unesene od strane drugih korisnika aplikacije. Recepti su ispisani u recyclerview te su vidljivi slika i naziv recepta, ako korisnik želi saznati više o pojedinom receptu to može ostvariti klikom na recept. Klikom na recept otvara se aktivnost za pregled recepta.

U gornjem desnom kutu korisnik ima opciju pretraživanja recepta po imenu klikom na ikonicu. Nakon upisanog naziva recepta ako taj recept postoji u bazi ispišu se svi recepti pod tim imenom. Klikom na njega otvara se pregled recepta.



Slika 23: Pretraživanje recepta

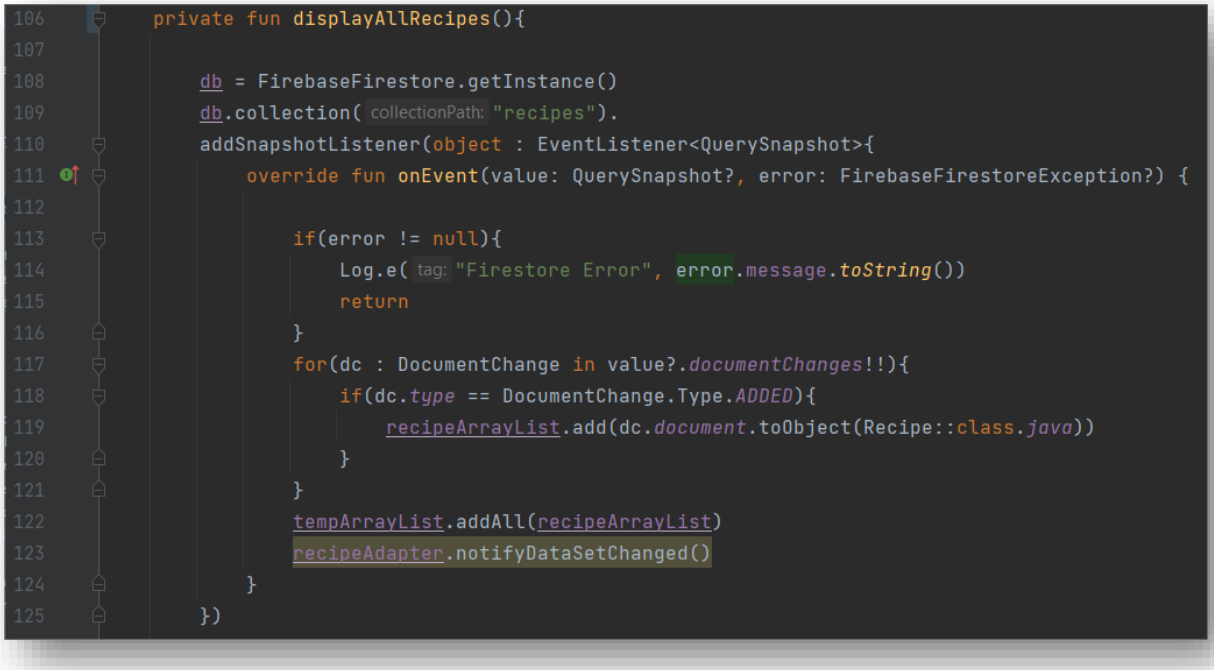
Način na koji pretraživanje recepta radi je pomoću 2 liste. Prva lista koja se ispiše kada se aktivnost otvori je pregled svih recepata, kada korisnik krene s unosom naziva lista iz te liste se traže recepti pod nazivom koji korisnik pretražuje te se ti recepti unose u privremenu listu koja se onda prikaže u recyclerview-u.



```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_item,menu)
    val item = menu?.findItem(R.id.search_action)
    val searchView = item?.actionView as SearchView
    searchView.setOnQueryTextListener(object: SearchView.OnQueryTextListener{
        override fun onQueryTextSubmit(newText: String?): Boolean {
                recyclerView.adapter!!.notifyDataSetChanged()
            return false
        }

        override fun onQueryTextChange(newText: String?): Boolean {
            tempArrayList.clear()
            val searchText = newText!!.lowercase(Locale.getDefault())
            if(searchText.isNotEmpty()){
                recipeArrayList.forEach{ it: Recipe
                    if(it.Name!!.lowercase(Locale.getDefault()).contains(searchText)){
                        tempArrayList.add(it)
                    }
                }
                recyclerView.adapter!!.notifyDataSetChanged()
            }
            else{
                tempArrayList.clear()
                tempArrayList.addAll(recipeArrayList)
                recyclerView.adapter!!.notifyDataSetChanged()
            }
            return false
        }
    })
    return super.onCreateOptionsMenu(menu)
```

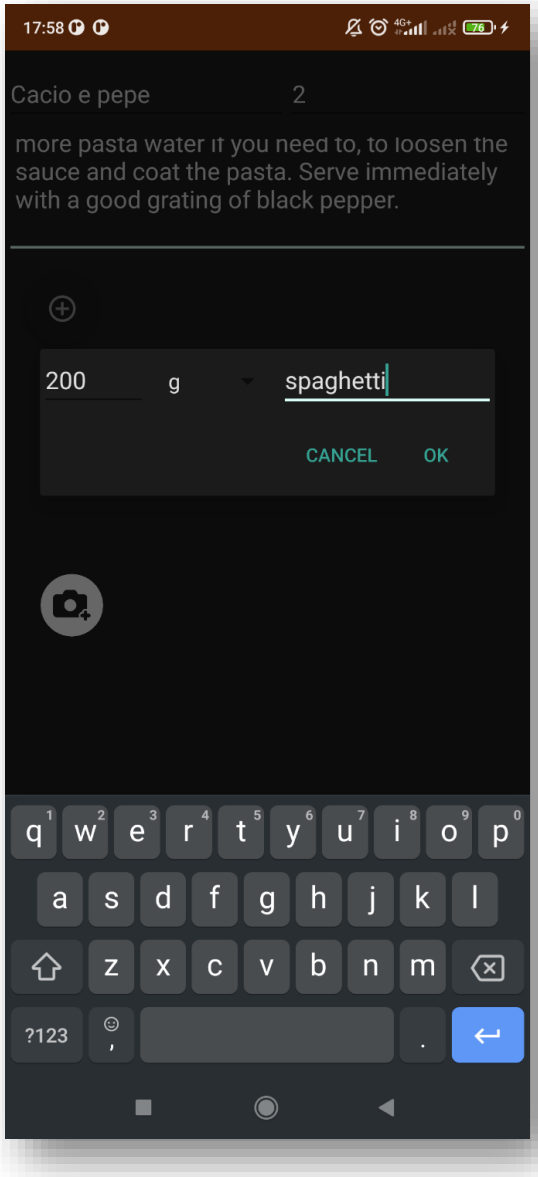
Slika 24: Kod za pretraživanje recepta



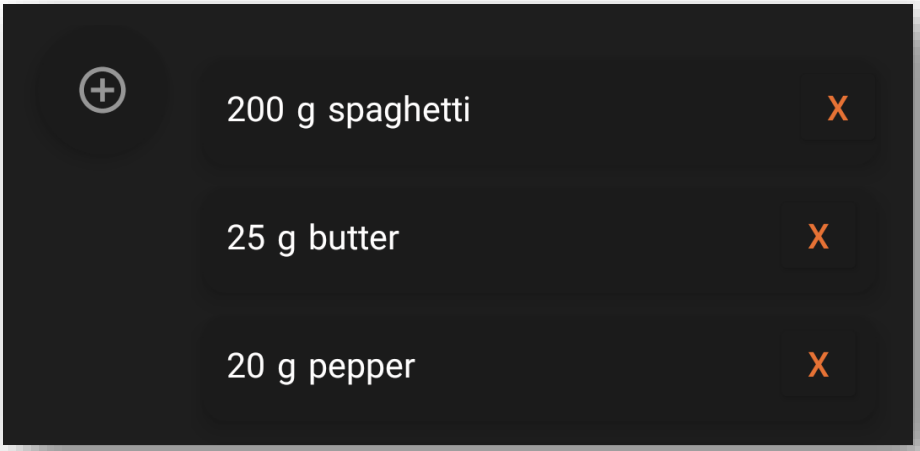
Slika 25: Kod za prikaz recepta

9.5. AKTIVNOST ADD RECIPE (DODAVANJE RECEPTA)

Kako bi korisnik dodao vlastiti recept potrebno je otići na aktivnost Add Recipe. Korisnik treba unijeti naziv recepta, broj porcija koje taj recept daje, metodu pripreme recepta. Korisnik dodaje sastojke jedan po jedan klikom na ikonicu + otvara se novi iskočni prozor u kojem korisnik upisuje količinu svakog sastojka, svaki sastojak ima dugme na sebi da ga korisnik može obrisati ako je pogriješio tijekom unosa. Nakon što su sastojci uneseni spremaju se u listu te prikazuju u recyclerview-u. Mjerne jedinice za količinu sastojka pospremljene su u spinner.



Slika 26: Dodavanje sastojaka

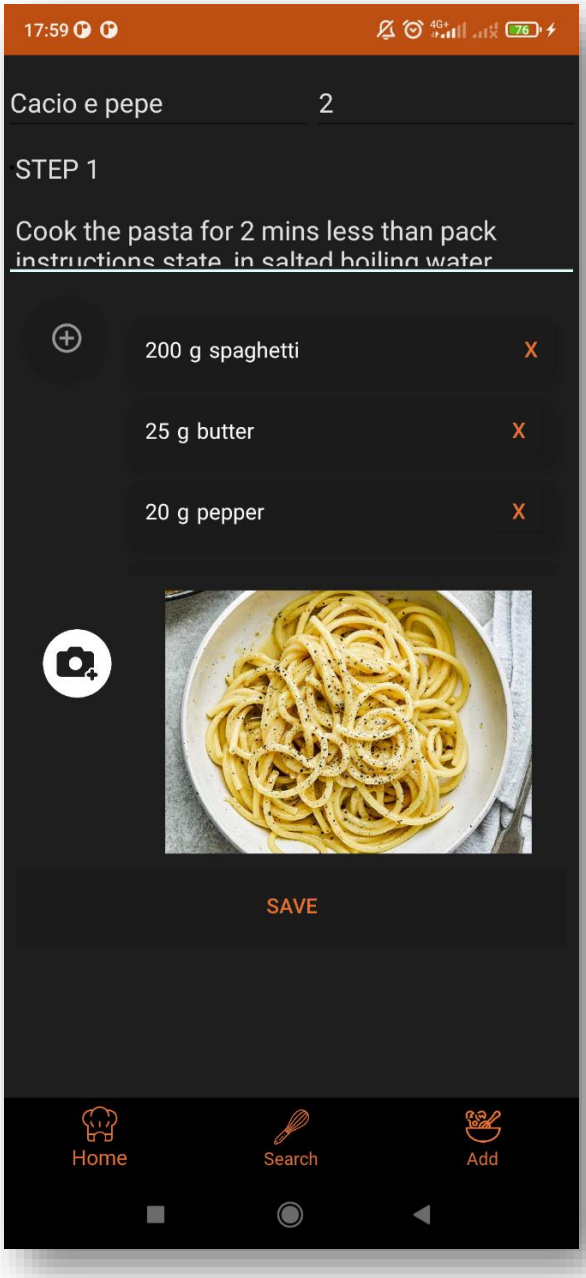


Slika 27: Prikaz sastojaka

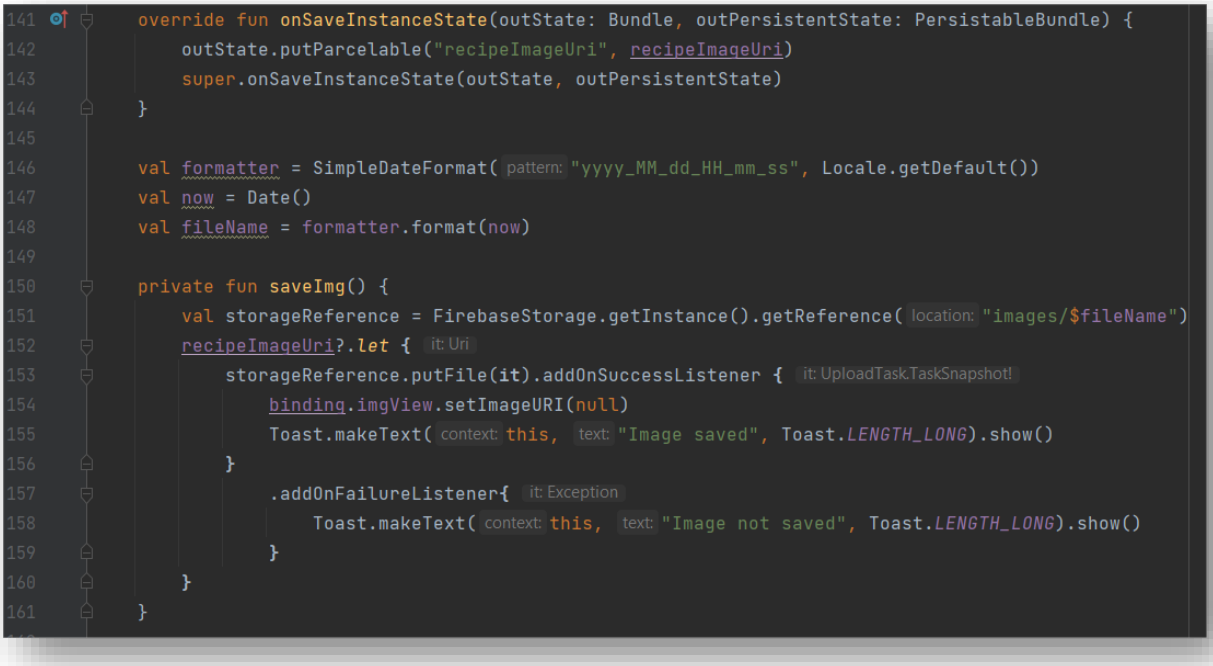
```
104    private fun addIngredient() {
105        val inflater = LayoutInflater.from( context: this)
106        val v = inflater.inflate(R.layout.add_item, root: null)
107        val ingredientName = v.findViewById<EditText>(R.id.etIngredientName)
108        val amount = v.findViewById<EditText>(R.id.etAmount)
109        unit = v.findViewById(R.id.spUnit) as Spinner
110        val pickUnit = resources.getStringArray(R.array.units)
111        unit.adapter = ArrayAdapter<String>( context: this,android.R.layout.simple_list_item_1, pickUnit)
112        unit.onItemSelectedListener = object : AdapterView.OnItemSelectedListener{
113            override fun onItemSelected(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
114            }
115            override fun onNothingSelected(p0: AdapterView<*>?) {
116                Toast.makeText( context: this@AddRecipeActivity,  text: "Please select unit", Toast.LENGTH_LONG).show()
117            }
118        }
119        val addDialog = AlertDialog.Builder( context: this)
120        addDialog.setView(v)
121        addDialog.setPositiveButton( text: "Ok"){
122            dialog,_->
123            val ingredient = ingredientName.text.toString()
124            val amount = amount.text.toString()
125            var selectedUnit = unit.selectedItem.toString()
126
127            ingredientList.add(Ingredient(amount,selectedUnit,ingredient))
128            ingredientAdapter.notifyDataSetChanged()
129            Toast.makeText( context: this, text: "Added", Toast.LENGTH_SHORT).show()
130            dialog.dismiss()
131        }
132        addDialog.setNegativeButton( text: "Cancel"){
```

Slika 28: Kod za dodavanje sastojaka

Nakon što korisnik unese sastojke klikom na ikonicu kamere otvara se galerija te korisnik može odabrati fotografiju recepta. Fotografije se spremaju u Firebase Storage pošto su veće od 1mb što je maksimalna veličina za podatke pospremljene u Firestore. Naziv fotografije namješten je da se spremi po datumu i vremenu u kojem je fotografija pospremljena u bazu. Sve fotografije se spremaju u mapu images u Firebase Storage bazi.

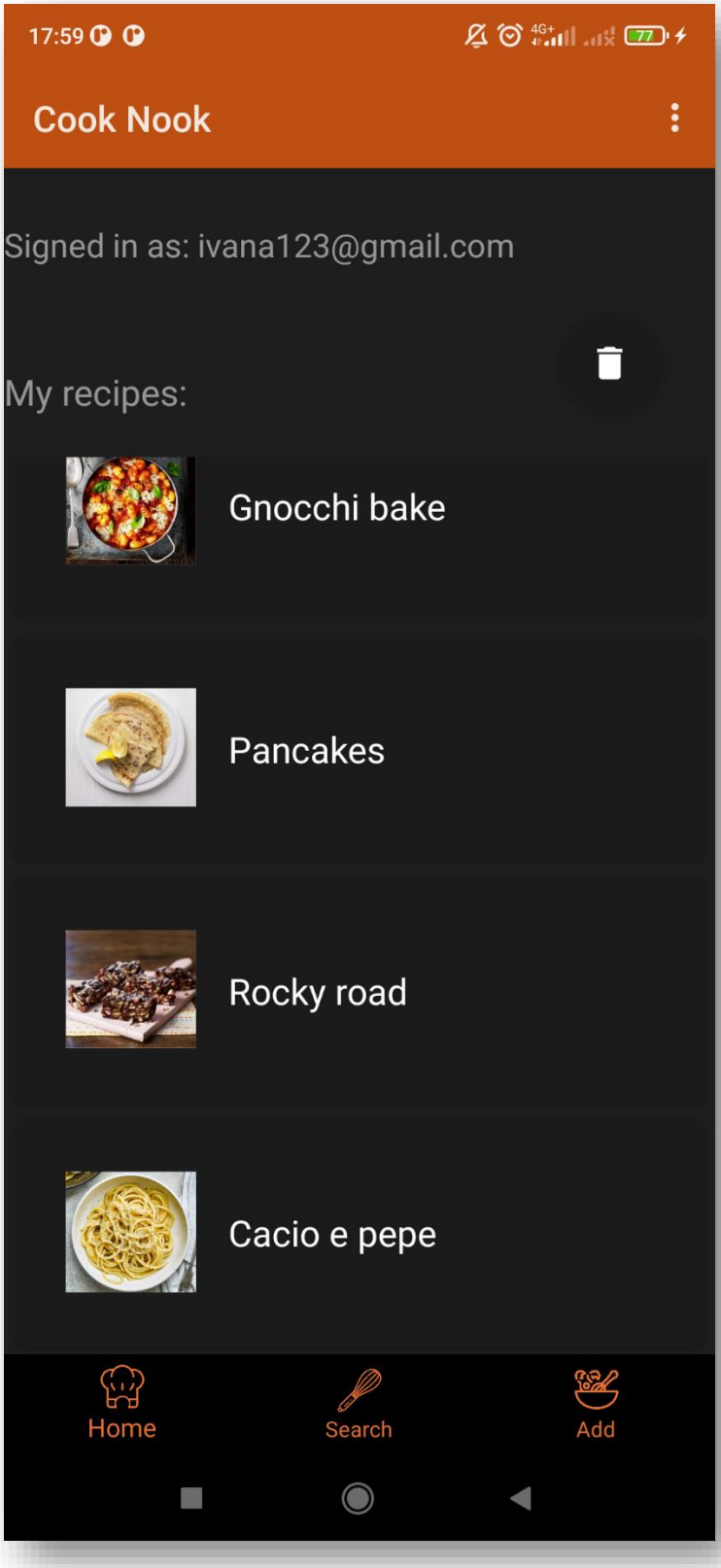


Slika 29: Dodavanje slike

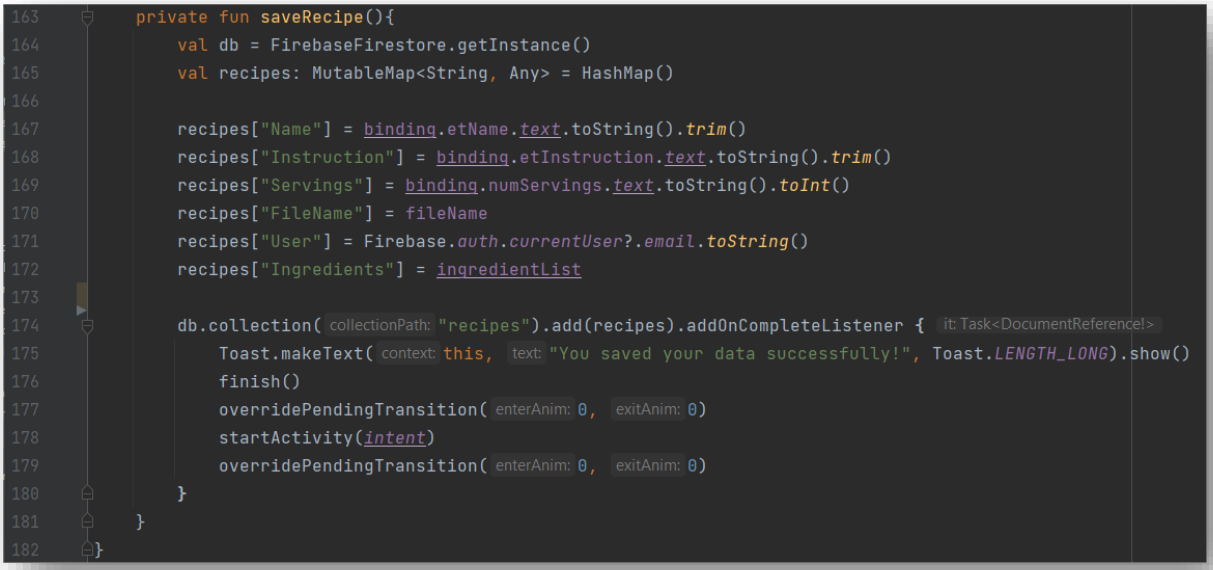


Slika 30: Kod za spremanje slike

Nakon što korisnik završi s upisom svih potrebnih podataka o receptu korisnik može spremiti recept te će se on nalaziti na njegovom profilu odnosno na početnoj stranici aplikacije. Podatci o receptu spremaju se u Firebase Firestore.



Slika 31: Slika spremljenog recepta



```
163    private fun saveRecipe(){
164        val db = FirebaseFirestore.getInstance()
165        val recipes: MutableMap<String, Any> = HashMap()
166
167        recipes["Name"] = binding.etName.text.toString().trim()
168        recipes["Instruction"] = binding.etInstruction.text.toString().trim()
169        recipes["Servings"] = binding.numServings.text.toString().toInt()
170        recipes["FileName"] = fileName
171        recipes["User"] = Firebase.auth.currentUser?.email.toString()
172        recipes["Ingredients"] = ingredientList
173
174        db.collection( collectionPath: "recipes").add(recipes).addOnCompleteListener {  it: Task<DocumentReference!> 
175            Toast.makeText( context: this,  text: "You saved your data successfully!", Toast.LENGTH_LONG).show()
176            finish()
177            overridePendingTransition( enterAnim: 0,  exitAnim: 0)
178            startActivity(intent)
179            overridePendingTransition( enterAnim: 0,  exitAnim: 0)
180        }
181    }
182}
```

Slika 32: Kod za spremanje recepta

10. ZAKLJUČAK

Mobilna aplikacija Cook Nook moderna je zamjena za gomilu ručno pisanih kuharica i knjiga te isječaka recepata iz raznih magazina. Ova aplikacija omogućava pregledno i jednostavno korištenje i pretraživanje recepata koji su uvijek na dohvat ruke. Korisnik može pospremati i pregledavati vlastite recepte ili inspiraciju za kuhanje potražiti u receptima unesenim od strane drugih korisnika aplikacije.

Aplikacija je napravljena koristeći programski jezik Kotlin. Platforma korištena za izradu ove aplikacije je Android Studio. Podatci o receptima i korisnicima pospremljeni su u Firebase bazu podataka.

Otkako je Google proglasio programski jezik Kotlin kao "prvoklasni" programski jezik za razvoj Android uređaja, Kotlin je doživio procvat popularnosti. Redovito je jedan od najomiljenijih jezika u godišnjoj anketi programera Stack Overflowa, a istaknut je i u godišnjem izvješću o vještinama programera HackerRank-a. Upravo zbog tog porasta popularnosti Kotlina ova aplikacija je napravljena u istom.

U današnjem svijetu gdje već više od 83% ukupnog stanovništva koristi pametne telefone vrijeme je da oni zamijene papirnate kuharice i listanje stranica u potrazi za receptom.

LITERATURA

[1] https://youtu.be/BDwwAQsm0KE 3.5.2022.

[2] https://youtu.be/Z-RE1QuUWPg 3.5.2022.

[3] https://youtu.be/Mc0XT58A1Z4 29.5.2022.

[4] https://youtu.be/Az4gXQAP-a4 19.6.2022.

[5] https://youtu.be/idbxxkF1l6k 21.6.2022.

[6] https://youtu.be/Bb8SgfI4Cm4 20.7.2022.

[7] https://youtu.be/SbNQxPDUWaI 22.7.2022.

[8] https://youtu.be/MfCiiTEwt3g 31.7.2022.

[9] http://surl.li/cxoiz 18.8.2022.

[10] https://gs.statcounter.com/os-market-share/mobile/worldwide 27.8.2022.

[11] https://smallbusinessjournals.com/pros-cons-android-studio/ 31.8.2022.

[12] https://www.makeuseof.com/what-is-google-firebase-why-use-it/ 31.8.2022.

[13] https://www.codecademy.com/resources/blog/what-is-kotlin-used-for/ 2.9.2022.

[14]https://docs.microsoft.com/en-us/xamarin/android/user-interface/layouts/recycler-view/parts-and-functionality 3.9.2022

POPIS SLIKA

SAŽETAK

Mobilna aplikacija Cook Nook odnosno kutak za kuhanje donosi korisnicima recepte iz dlana ruke. Korisnik, nakon što se prijavi u aplikaciju, može unijeti vlastite recepte. Kod unosa recepta potrebno je znati naziv, broj porcija, sastojke, metodu pravljenja recepta te dodati fotografiju recepta. Kako korisnici unose recepte tako se baza puni sa velikim odabirom recepata za pretraživanje. Korisnik pretražuje recept po njegovom nazivu te mu se prikažu svi recepti dostupni pod tim nazivom. Za bazu podataka korišten je Firebase, aplikacija je napisana u programskom jeziku Kotlin, te platforma korištena za izradu aplikacije je Android Studio.

Ključne riječi: Diplomski rad, Cook Nook, Recepti, Kotlin, Android Studio, Firebase, Firestore, Firebase Storage, Emulator, Mobilna aplikacija, Mobilna aplikacija za recepte

ABSTRACT

Mobile application Cook Nook brings recipes to users from the palm of their hand. The user, after logging into the application, can enter his own recipes. When entering a recipe, it is important to know its name, serving size, ingredients, method of making the recipe and add a photo of the recipe. As users enter recipes database gets filled with large selection of searchable recipes. The user searches for a recipe by its name and all recipes available under that name are displayed. Firebase is used for storing data for this application, the application is written in programming language Kotlin, and platform used to create this application is Android Studio.

Keywords: Graduation thesis, Cook Nook, Recipes, Kotlin, Android Studio, Firebase, Firestore, Firebase Storage, Emulator, Mobile application, Mobile applications for recipes