Tim: **AIR1816** Model 2

*Božo Čulo*

*Vedran Grbavac*

*Hrvoje Dumančić*

*Luka Garić*

*Darko Gajić*

**Github:**

<https://github.com/dagajic/MyGroceryPal>

**Trello:**

<https://trello.com/b/aB5cMeoT/air-mygrocerypal>

<https://trello.com/b/InvzEIbt/mygrocerypalmanagment>

**Tehnička dokumentacija**

**Specifikacija zahtjeva**

Sadržaj

[1 Uvod 3](#_Toc529967924)

[1.1 Svrha 3](#_Toc529967925)

[1.2 Djelokrug 3](#_Toc529967926)

[1.3 Definicije 4](#_Toc529967927)

[1.4 Reference 4](#_Toc529967928)

[1.5. Pregled 4](#_Toc529967929)

[2 Generalni opis 5](#_Toc529967930)

[2.1 Arhitektura aplikacije 5](#_Toc529967931)

[2.2 Funkcije proizvoda 6](#_Toc529967932)

[2.3 Karakteristike korisnika 7](#_Toc529967933)

[2.4 Ograničenja 7](#_Toc529967934)

[2.5 Pretpostavke i opasnosti 7](#_Toc529967935)

[3 Specifični zahtjevi 8](#_Toc529967936)

[3.1 Zahtjevi performansi 8](#_Toc529967937)

[3.2 Logički zahtjevi nad bazom podataka 8](#_Toc529967938)

[3.3 Ograničenje dizajna 8](#_Toc529967939)

[3.4 Obilježja aplikacije 8](#_Toc529967940)

[3.5 Sučelja 9](#_Toc529967941)

[4 Opis dizajna sustava 10](#_Toc529967942)

[4.1 ERA model 10](#_Toc529967943)

[4.2 Dijagram klasa 12](#_Toc529967944)

[4.3 Web servisi (Firebase) 12](#_Toc529967945)

[4.4 Mockup 15](#_Toc529967946)

[4.5 Use Case 16](#_Toc529967947)

[4.6 Funkcionalnosti sustava 16](#_Toc529967948)

# Uvod

Sljedeća poglavlja predstavljaju detaljnu specifikaciju zahtjeva prema IEEE Std 830-1998 Software Requirements Specifications standardu. U sljedećem dijelu teksta će biti opisani svrha, djelokrug, definicije termina, referenci te opći pregled.

## Svrha

Ovaj dokument predstavlja opis funkcionalnosti aplikacije MyGroceryPal koja je namijenjena posredovanju između korisnika koji žele dostavu namirnica iz dućana na svoju adresu i onih korisnika koji bi pružili takvu vrstu usluge. Biti će opisane tehničke i softverske specifikacije te definirani korisnički zahtjevi. Obuhvatiti će se ograničenja, način korištenja aplikacije od strane korisnika te komunikacija aplikacije s vanjskim servisima. Dokument je namijenjen korisnicima apliakcije kao način prezentiranja aplikacije koja će se izraditi, ali također i samim programerima kako bi se dobila opća slika te svrha aplikacije.

## Djelokrug

Aplikacija se izdaje pod imenom „MyGroceryPal“ i namijenjena je različitim korisnicima. Aplikacija bi vršila posredovanje između fizičkih osoba prilikom nabave namirnica iz dućana u smislu da korisnik može sudjelovati kao naručitelj namirnica i/ili kao dostavljač namirnica na željenu adresu uz proviziju za obavljenu uslugu. Aplikacija će sadržavati modul za pretraživanje popisa proizvoda, webservise koji se odnose na firebase bazu podataka koja sadržava potrebne podatke za rad aplikacije, modul za plaćanje preko PayPala, modul za google mape i dohvat podataka o mjestima te funkcionalnosti poput kreiranja liste namirnica (grocery list) za naručitelja odnosno prihvata takve liste od strane dobavljača namirnica, registracije i prijave korisnika, postavki korisnika i drugih funkcionalnosti koje će biti detaljno objašnjene u ovom dokumentu. Isporučeni sustav će se sastojati od mobilne aplikacije za Android OS koja komunicira sa firebase bazom podataka.

## Definicije

U ovom dijelu se nalaze definicije termina korištenih u ovom dokumentu:

* Naručitelj – korisnik koji kreira listu namirnica i želi da mu se dostave na željenu adresu. Naručitelj odabire dućan za koji mu se prikazuju proizvodi koje može filtrirati i dodavati u listu namirnica (grocery list). Lista se objavljuje sa definiranim vremenom trajanja i može ju prihvatiti drugi korisnik kako bi izvršio uslugu dostave namirnica.
* Dostavljač – korisnik koji između kreiranih listi namirnica (grocery list) odabire one koje želi prihvatiti kako bi izvršio uslugu dostave naručitelju
* Korisnik – registrirani korisnik aplikacije koji može biti u svojstvu naručitelja i/ili dostavljača.
* Grocery list (GL) – lista namirnica za dostavu koju kreira naručitelj.

## Reference

Reference koje su se koristile za izradu ovoga dokumenta:

1. IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications

## Pregled

Ostatak ovog dokumenta sadrži opći opis zahtjeva funkcionalnosti aplikacije. To uključuje:

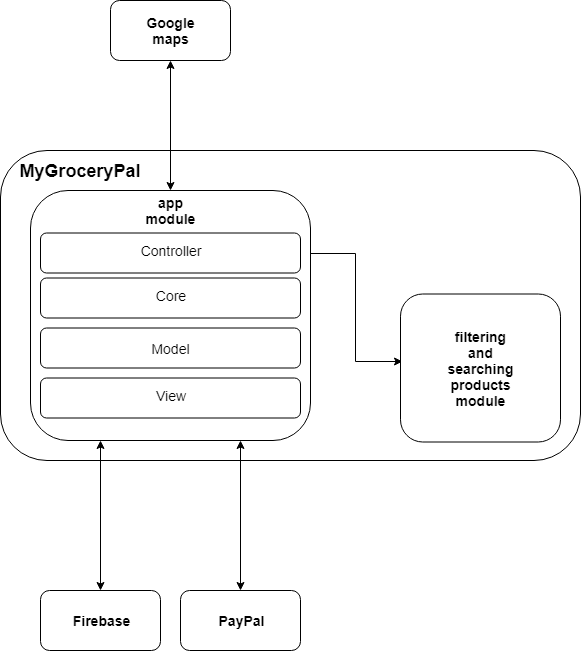
* Opis arhitekture aplikacije
* Pregled osnovnih funkcionalnosti aplikacije
* Karakteristike pojedinih korisnika koji mogu koristiti aplikaciju
* Ograničenja aplikacije koje proizlazi iz dostupnosti servisa i alata u kojima je aplikacija izrađena
* Pretpostavke i ovisnosti koje su nužne za uspješan rad aplikacije
* Dodatni specifični zahtjevi od strane korisnika

# Generalni opis

Sljedeći dio dokumenta predstavlja generalni opis aplikacije te njezinih osnovnih funkcionalnih osobina, odnosno načine komuniciranja s bazom podataka, s vanjskim okruženjem i slično.

## Arhitektura aplikacije

Aplikacija MyGrocePal sastoji se od app modula i modula za filtriranje i pretraživanje proizvoda. U app modulu nalaze se 4 paketa Core, Controller, Model i View (MVC pattern). Od vanjskih servisa, korišten je Firebase za pohranu podataka, PayPal za implementaciju funkcionalnosti plaćanja i Google Maps za implementaciju prikaza narudžbi na temelju lokacije.



## Funkcije proizvoda

1. **Registracija korisnika** – svaki korisnik se registrira kako bi mogao koristiti aplikaciju. Nakon registracije korisniku se šalje aktivacijski link na mail.
2. **Prijava korisnika** – korisnik se prijavljuje kako bi imao pristup aplikaciji. Upisani podaci (korisničko ime i lozinka) se provjeravaju u bazi podataka.
3. **Zaboravljena lozinka** – mogućnost upisa nove lozinke ukoliko je zaboravljena.
4. **Glavni izbornik** – prikaz glavnog izbornika radi lakše navigacije.
5. **Popis aktivnih GL-ova (grocery lista)** – naručitelj vidi svoje aktivne GL-ove. Dobavljač vidi sve aktivne GL-ove koje može prihvatiti.
6. **Popis prošlih GL-ova** – naručitelj vidi popis prošlih GL-ova koji su realizirani ili su istekli. Može kreirati nove iz starih.
7. **Prikaz proizvoda po trgovini** – naručitelj vidi sve proizvode za odabranu trgovinu.
8. **Dodavanje novih GL-ova** – naručitelj kreira novi GL. Može odabrati drugu adresu dostave koja je različita od njegove adrese stanovanja. Odabire vrijeme dostave i trajanje prikaza GL-a.
9. **Dodavanje proizvoda na novi GL** – naručitelj dodaje proizvode na novi GL.
10. **Pretraga i filtriranje proizvoda** – prikaz proizvoda za odabranu trgovinu i drugi načini filtriranja radi lakšeg i bržeg odabira proizvoda koji se dodaje na GL od strane naručitelja.
11. **Odabir lokacije** – korisnik odabire lokaciju dostave.
12. **Podešavanje količine proizvoda i objava GL-a** – naručitelj odabire količinu za svaki proizvod ukoliko je veća od jednog komada i objavljuje kreirani GL kako bi postao vidljiv ostalim korisnicima.
13. **Odabir plaćanja preko PayPal-a** – naručitelj plaća izračunati iznos preko svojeg PayPal računa na račun MyGroceryPal-a. Iznos se nakon obavljne kupnje i izvršene dostave prebacuje na PayPal račun dostavljača.
14. **Prihvaćanje i odbijanje GL-a** – dostavljač može prihvatiti objavljeni GL i time prihvaća izvršenje usluge dobave namirnica za naručitelja. Ukoliko ne želi da mu se neki GL prikazuje može ga ignorirati odnosno premjestiti u listu ignoriranih.
15. **Popis prihvaćenih i odbijenih (ignoriranih) GL-ova** – dostavljač vidi popis prihvaćenih i ignoriranih GL-ova.
16. **Odabir kupljenih proizvoda sa prihvaćenog GL-a i upis iznosa računa** – dostavljač odabire proizvode koje je kupio i upisuje iznos računa koji je dobio u dućanu u aplikaciju.
17. **Statistika dostavljača** – dostavljač može vidjeti neke statističke podatke kao što su ukupan broj dostava, prosječna provizija i sl.
18. **Statistika naručitelja** – statistički podaci koje vidi naručitelj (broj narudžbi, prosjek računa..)
19. **Postavke korisnika (promjena osobnih podataka)** –promijena adrese, grada, maila, lozinke i sl.

## Karakteristike korisnika

Svaki korisnik može biti u ulozi naručitelja usluge i/ili dostavljača odnosno osobe koja izvršava uslugu za naručitelja. Sukladno tome svaki korisnik može koristiti vrstu pogleda nekog od ove dvije uloge ovisno u kojoj ulozi se nalazi u danom trenutku.

## Ograničenja

Za rad aplikacije potrebna je internetska veza budući da se sve promijene korisnika bilježe u centralnoj bazi (firebase). Korisnici moraju imati mobilne uređaje sa android operativnim sustavom kako bi mogli instalirati aplikaciju.

## Pretpostavke i opasnosti

Pretpostavka za ispravan rad aplikacije je internetska veza budući da su podaci na centraliziranoj bazi podataka. Korisnici ne mogu direktno komunicirati te kako bi podaci bili ažurni moraju se sinkronizirati sa podacima koji se nalaze u firebase bazi podataka.

# Specifični zahtjevi

## Zahtjevi performansi

Nema ograničenja performansi za ovu vrstu aplikacije budući da se ne radi sa vrlo velikim količinama podataka. Firebase omogućuje skalabilnost tako da veći broj korisnika ne bi trebao utjecati na performanse.

## Logički zahtjevi nad bazom podataka

U nastavku dokumenta nalazi se ERA model. Radi boljeg razumijevanja korišten je ovaj model iako je firebase noSQL baza podataka.

## Ograničenje dizajna

Aplikacija je namijenjena za uređaje sa Android OS-om te nema ograničenja dizajna za uređaje sa API levelom 21 ili više.

## Obilježja aplikacije

*Dostupnost* – aplikacija je dostupna većinu vremena osim u slučaju ako dođe do problema sa spajanjem na firebase bazu podataka odnosno radi održavanja servera.

*Pouzdanos*t - Prije samog isporučivanja aplikacija, mora se provesti sveobuhvatno testiranje funkcionalnosti kako bi se osigurala ispravnost načina djelovanja i izvršavanja pojedinih operacija za koje će aplikacija biti zadužena. Testiranje se mora obaviti na prikladnom sklopovlju i s prikladnim programima kako bi se što točnije predvidjeli uvjeti korištenja aplikacije.

*Sigurnost* – pristup aplikaciji ima samo registrirani korisnik koji se prijavljuje sa svojim korisničkim imenom ili emailom i lozinkom. Nakon prijave aplikacija pamti trenutnog korisnika. U aplikaciji su vidljivi samo oni podaci kojima korisnik smije pristupiti.

*Održavanje* – funkcionalnost pretraživanja proizvoda će biti implementirana modularno. Sve eventualne promijene u aplikaciji (npr. način preuzimanja podataka o proizvodima) biti će isporučene u novoj verziji aplikacije.

*Portabilnost* – apliakcija se instalira na mobilni uređaj sa Android operativnim sustavom.

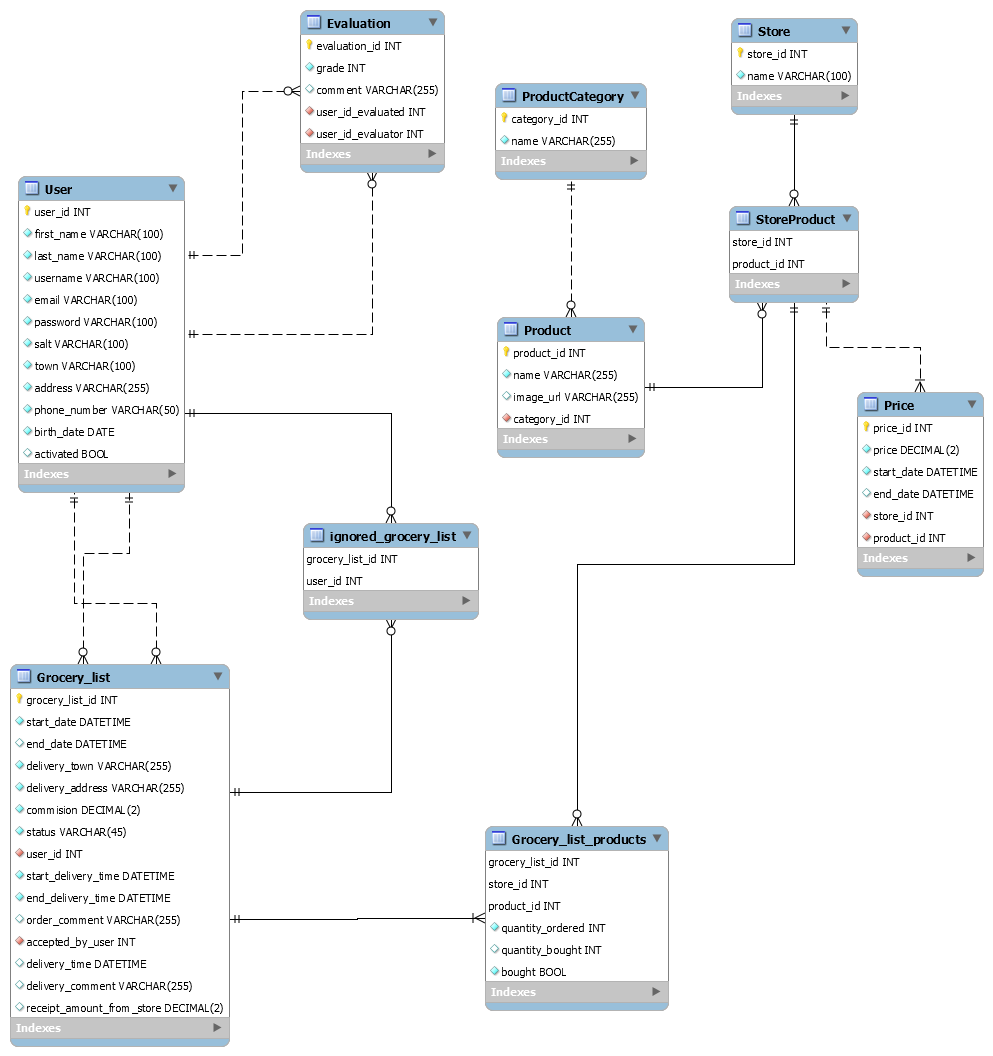
## Sučelja

*NavigationView.OnNavigationItemSelectedListener* – listener za rukovanje događajima koji se odnose na odabir item-a sa navigacijske trake. Prima parametar MenuItem (odabrani item) te vraća boolean tip podatka

*View.OnClickListener* – poziv funkcije onClick kad se klikne na view. Funkcija prima parametar View i ne vraća ništa.

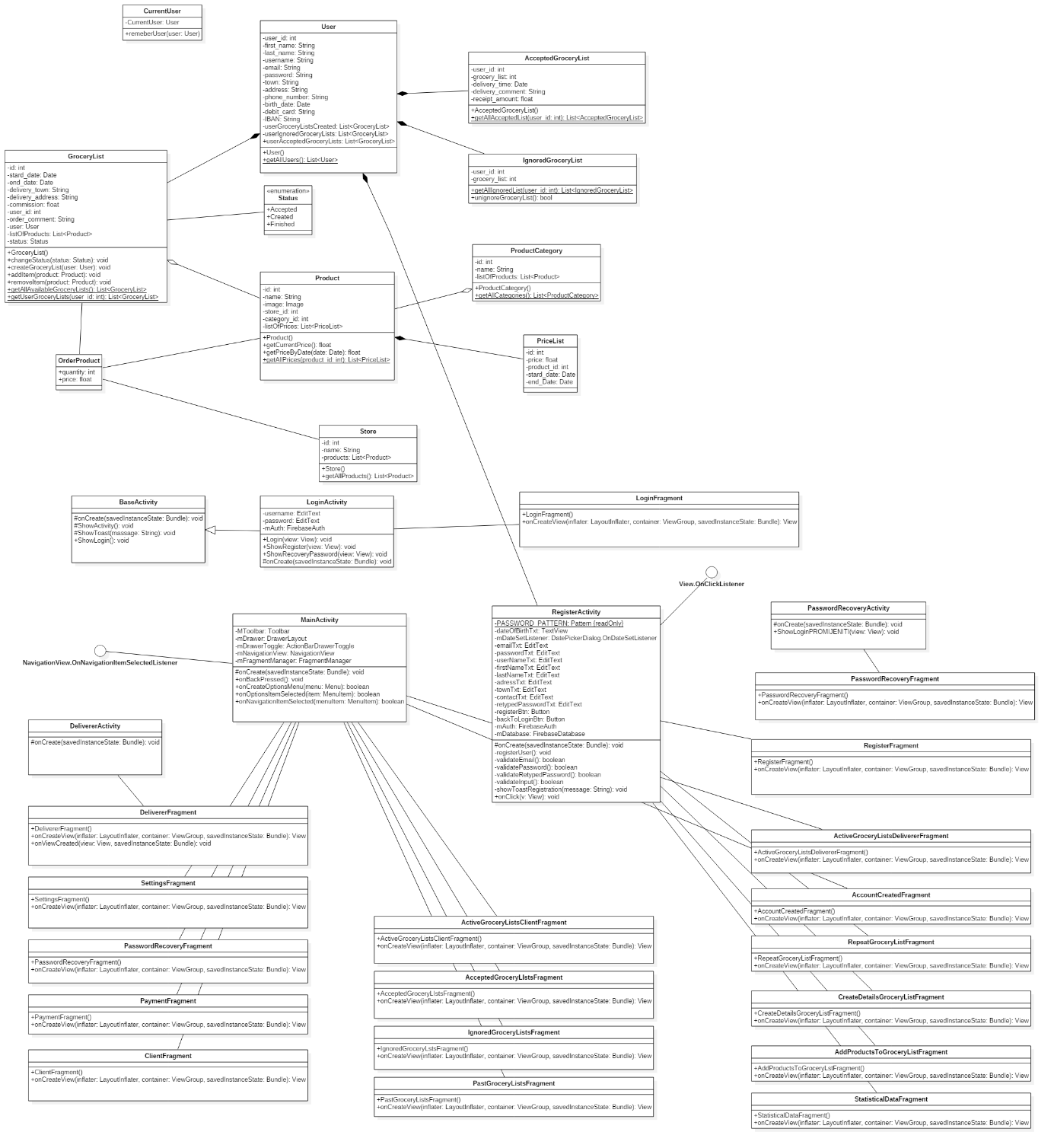
# Opis dizajna sustava

## ERA model



ERA model aplikacije MyGroceryPal sastoji se od 10 tablica. Tablica User sadrži podatke o korisnicima aplikacije. Podaci se unose prilikom registracije korisnika. Korisnik mora aktivirati račun putem upisanog maila. Svaki korisnik ima lozinku i korisničko ime s kojim se prijavljuje u aplikaciju. Product se odnosi na jednu kategoriju proizvoda (tablica Category), a svaka kategorija može imati više proizvoda. Store sadrži podatke o dućanu te svaki proizvod može biti u više dućana i svaki dućan ima više proizvoda (tablica StoreProduct). Cjenik se konkretno odnosi na cijenu proizvoda u određenom dućanu (tablica Price) tako da isti proizvod može imati različitu cijenu ovisno o dućanu u kojem se nalazi. Također, cjenik sadrži podatke o cijenama za proizvod kroz vremensko razdoblje. Ukoliko je došlo do promijene cijene u bazu se sprema vrijeme do kad je ta cijena važila (end\_date) te se upisuje nova cijena za koju znamo datum i vrijeme od kad se primjenjuje (start\_date). Korisnik kreira grocery list (svi podaci vezani uz narudžbu proizvoda iz nekog dućana) koji neki drugi korisnik može prihvatiti u smislu da će obaviti uslugu kupnje za korisnika koji je kreirao grocery list. Korisnik može kreirati više grocery lista, a lista se prihvaća od strane jednog korisnika. Korisnik koji je u ulozi dobavljača (vršitelja usluge) može ignorirati grocery liste tako što se one stavljaju u ignored\_grocery\_list i korisnik te liste više ne vidi kod daljnjeg pregleda listi. Svaka lista može biti ignorirana od strane više korisnika. Grocery list može sadržavati jednu ili više stavki (proizvoda) koji se nalaze u tablici Grocery\_list\_products, dok se proizvodi mogu nalaziti na više grocery lista. Grocery\_list\_products sadrži informacije o količini naručenih proizvoda, količini kupljenih proizvoda (popunjava se nakon kupnje) i da li je proizvod kupljen. Korisnik može evaluirati drugog korisnika bilo da se radi o naručitelju usluge ili onome koji uslugu obavlja. U tablicu Evaluation upisuju se potrebni podaci sa šifrom korisnika koji ocijenjuje i onoga koji je ocijenjen.

## Dijagram klasa



## Web servisi (Firebase)

Web servisi koji su vezani uz bazu podataka su implementirani pomoću Firebase-a. Firebase je razvojna platforma koja omogućava timova brži razvoj mobilnih aplikacija. Za realizaciju same baze korištene su komponente Firebase Authentication i Realtime Database. Pomoću Firebase Authentication implementirane su funkcionalnosti prijave, registracije i zaboravljene lozinke. Kako bi se korisnik uspješno registrirao mora unijeti osnovne podatke, kao što su korisničko ime, e-mail, lozinka itd., nakon čega mu se šalje poveznica na mail koju mora potvrditi za uspješnu registraciju. Kako Firebase Authentication pamti samo e-mail i password ostali podaci vezani uz korisnika spremljeni su u Realtime database. Za svakog korisnika koji se registrira Firebase authentication kreira jedinstveni uid, na temelju kojega možemo prepoznati korisnika i povezati podatke iz Firebase authentication i Realtime database-a. U Realtime database-u nalaze se svi podaci vezani uz artikle, trgovine, liste narudžbi, korisnika itd. Kako bismo u Android Studiju radili s Firebase-om potrebno je u gradle skripte (Module: app) dodati sljedeće linije:

**implementation 'com.google.firebase:firebase-auth:16.0.1'**

**implementation 'com.google.firebase:firebase-database:16.0.1'**

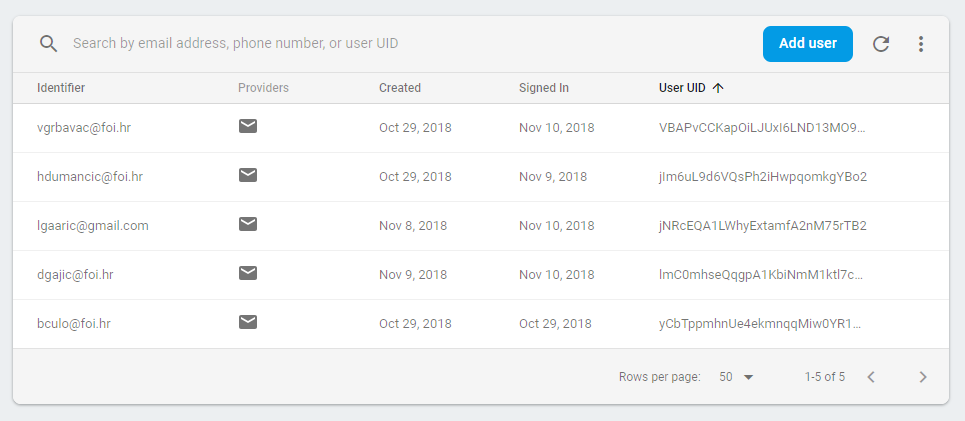
**implementation 'com.google.firebase:firebase-core:16.0.1'**

U projekt je potrebno dodati google-service.json koji je moguće preuzeti sa Firebase-a. Nakon uspješnog spajanja sa Firebase-om koriste se klasa FirebaseAuth za rad s Firebase authentication i klase Query i FirebaseDatabase za rad s Realtime database.

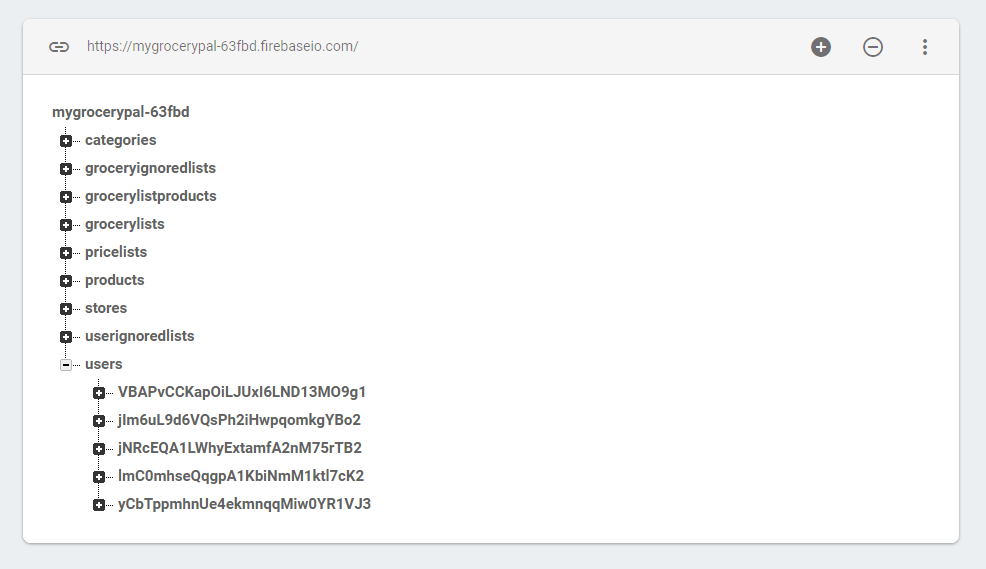
Na sljedećim slikama možemo vidjeti Firebase authentication i Realtime database. Realtime database je NoSQL baza podataka pa se mogu vidjeti sitne razlike u odnosu na era model koji će biti prikazan u ovoj dokumentaciji. Za razliku od relacijskih baza podataka, ovdje ne možemo pisati standardne SQL upite. Čitava baza je zapravo jedno stablo kroz koje se krećemo kako bi dohvatili željene podatke.

Primjer dohvaćanja korisnika.

Naime, ako želimo dohvatiti podatke nekog korisnika, potrebno je prvo dohvatiti korijen stabla (mygrocerypal-63fbd), zatim dohvatiti dijete users i zatim odabrati odgovarajuću id korisnika.



Slika 1. Firebase authentication



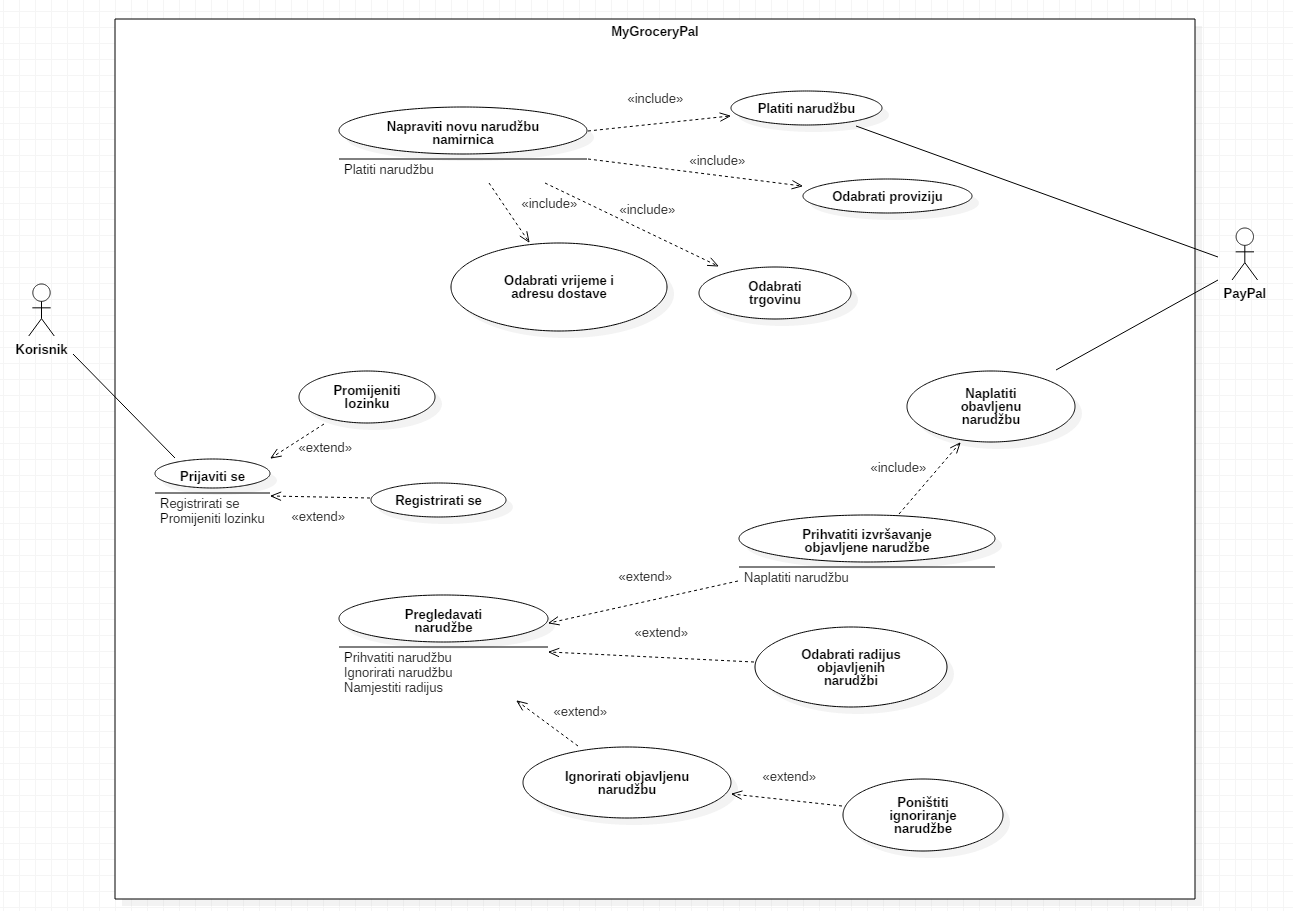
Slika 2. Realtime Database

**Napomena:** atribut password je vidljiv kako bi aplikaciju bilo lakše testirati. U pravilu taj atribut ne bi trebao biti u čitljivom obliku u Realtime database-u.

## Mockup

U izradi i dizajnu, mockup ili modeliranje je skala ili model pune veličine dizajna ili uređaja, koji se koristi za podučavanje, demonstraciju, evaluaciju dizajna, promociju i druge svrhe. Mockup je prototip koji osigurava barem dio funkcionalnosti sustava i omogućuje testiranje dizajna. Dizajneri uglavnom upotrebljavaju modele za prikupljanje povratnih informacija od korisnika. Na slici iznad možemo vidjeti mockup za našu aplikaciju te iz je njega lako vidjeti dizajne pojedinih ekrana te prateći strelice možemo uočiti navigaciju između pojedinih ekrana.

## Use Case



Dijagram korištenja prikazuje sve funkcionalnosti aplikacije korisniku za korištenje. U ovom dijagramu tako vidimo opciju izrade nove narudžbe namirnice prilikom koje je korisnik obvezan odabrati trgovinu iz koje se namirnice moraju kupiti, unijeti vrijeme i adresu dostave, proviziju za dobavljača te platiti iznos kupnje plus proviziju za nabavljača. Nakon napravljene nove narudžbe, korisnik aplikacije u pogledu nabavljača može pregledavati objavljene i aktualne narudžbe. Postoji opcija ignoriranja narudžbi koje mu se ne sviđaju, ali i opcija poništavanja ignoriranja, odnosno vraćanja na aktualni prikaz. Nadalje, narudžbe koje smatra povoljnim može prihvatiti te nakon završetka obavljanja narudžbe mora naplatiti naručitelju iznos kupovine, odnosno dobije točan iznos novca narudžbe s provizijom od iznosa koji je uplatio naručitelj.

## Funkcionalnosti sustava