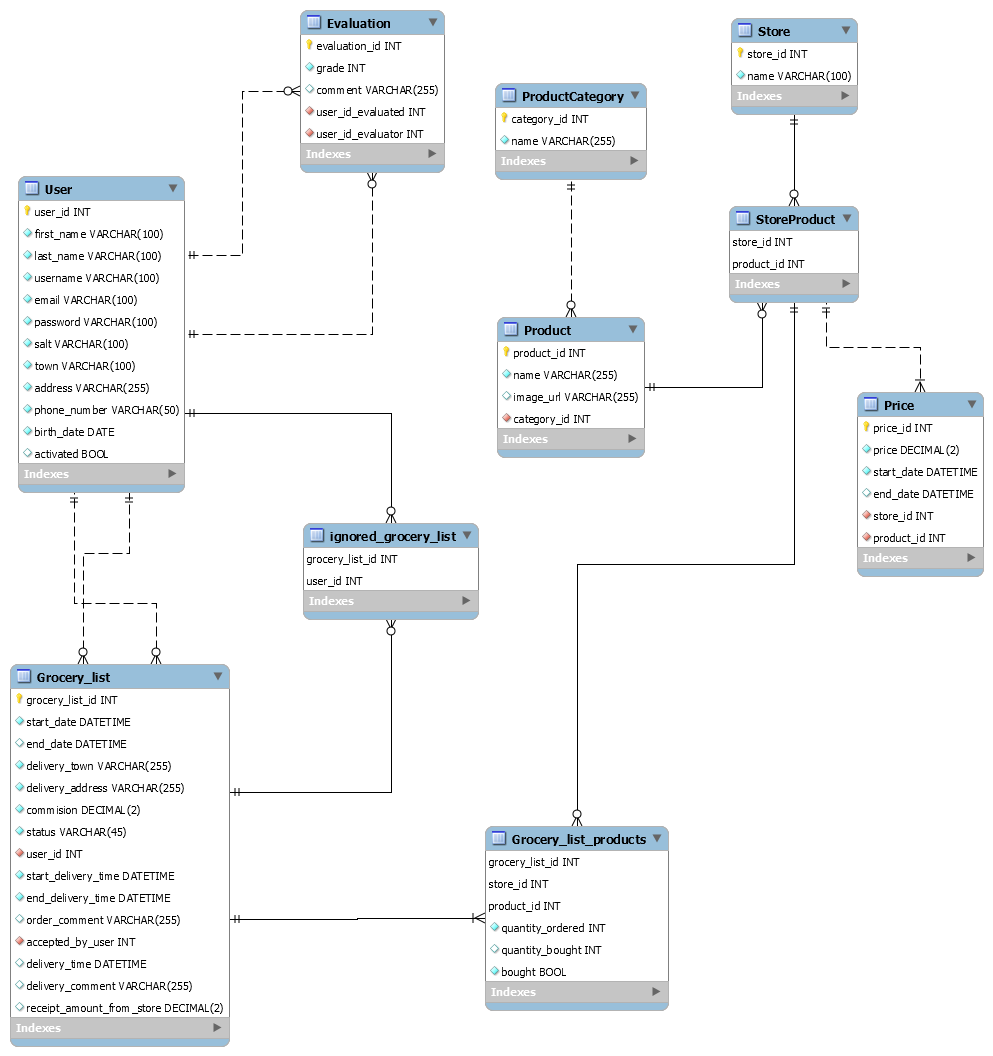
**3. Opis dizajna sustava (opis funkcionalnosti aplikacije uz priložene dijagrame)**

**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.

**4. 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'**

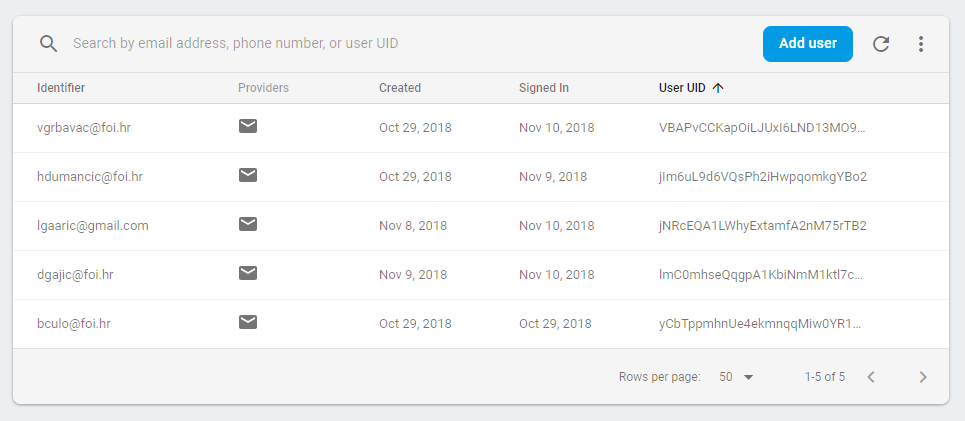
**implementation 'com.google.firebase:firebase-core:16.0.1'**

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.

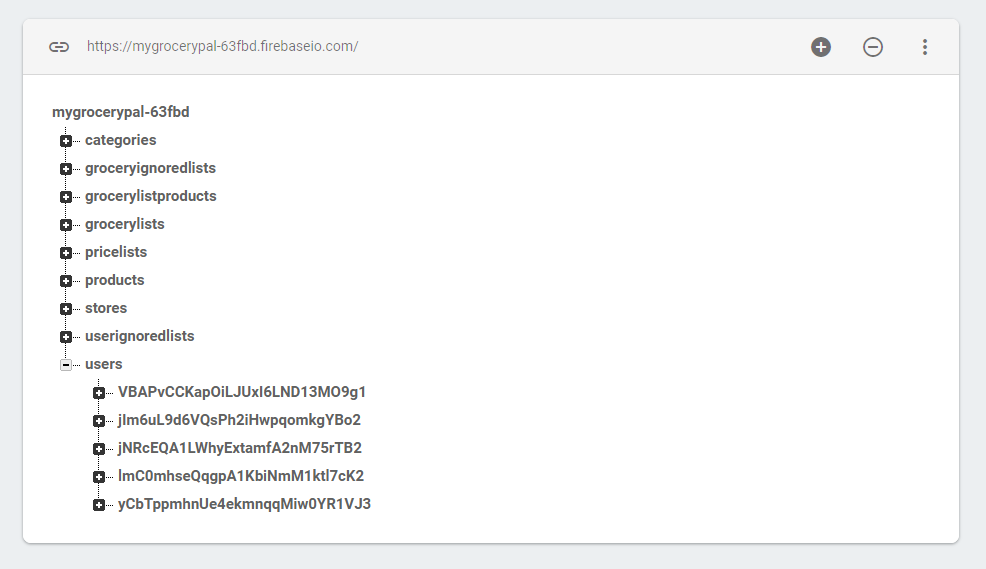
Na sljedećim slikama možemo vidjeti Firebase authentication i Realtime database. Realtime database je NoSQL baza podataka pa se mogu vidjeti sitne razlike u odnosu na era model koji će biti prikazan u ovoj dokumentaciji. Za razliku od relacijskih baza podataka, ovdje ne možemo pisati standardne SQL upite. 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.



*1. Firebase Authentication*



*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.