Model perzistencije

**FoodPin**

|  |
| --- |
| Članovi tima: |
| Tea Mitić 17274 |
| Dimitrije Mitić 17269 |

# 

Contents

[Modeli podataka 3](#_Toc99100333)

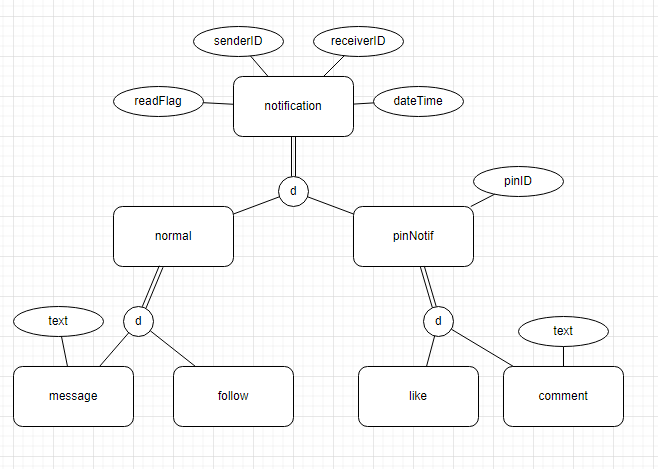
[Mehanizam mapiranja 6](#_Toc99100334)

# Modeli podataka

FoodPin projekat sadrži dve baze podataka, Neo4j, kao primarnu bazu za skladištenje podataka, i MySQL kao log bazu za perzistenciju notifikacija i poruka izmedju klijenata.

Modeli podataka projekta FoodPin prikazani su na slikama ispod:

MySQL EER model:



Model je zamišljen tako da postoji osnovni tip notifikacije koji ima atribute:

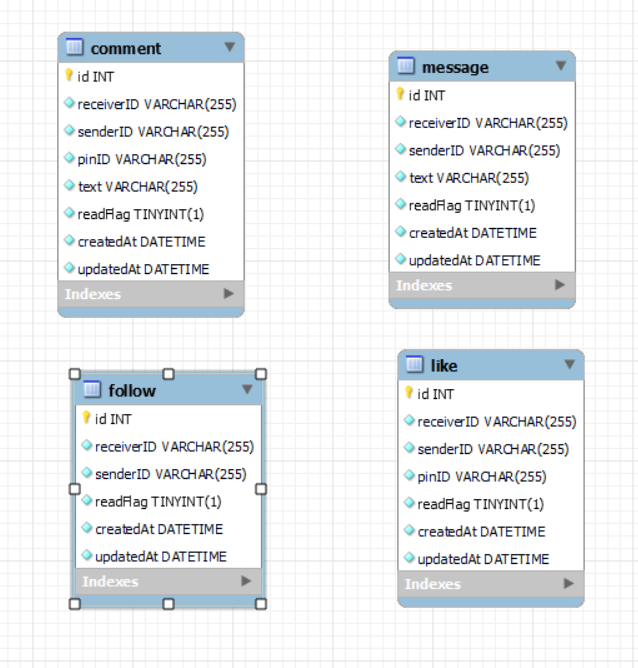
* id osobe čija akcija je kreirala notifikaciju
* id osobe kojoj treba da stigne notifikaija
* dateTime atribut koji se sastoji od dve kolone datum i vreme kreiranja notifikacije i datum i vreme ažuriranja notifiakcije (ovo je automatski dodato od strane ORM mapera Sequelize za node.js)
* flag za pročitanu notifikaciju kako bi se na frontu znalo da li treba prikazati nepročitanu notifikaciju

Iz osnovnog modela se izvode druga dva modela notifikacije, jedan vezan za pin (objavu u aplikaciji) i sadrži id te objave. Drugi model je bez tog pina i zove se normalNotification.

Iz gore navedenih notifikacija se nasledjuju po dva konkretna tipa notifikacija:

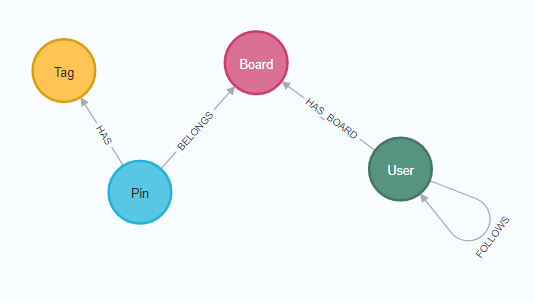
* MessageNotification za logovanje poruka izmedju klijenata, sadrzi atribut tekst poruke
* FollowNotification za logovanje notifikacija o zapraćivanju drugih korisnika aplikacije
* LikeNotification za logovanje notifikacija o lajkovanoj objavi
* CommentNotification za logovanje notifikacija o postavljenom komentaru na objavi

MySql model mapirana baza podataka na relacioni model prikazana je na sledećoj slici:



EER model se razlikuje od implementiranog relacionog modela zbog načina mapiranja EER modela. Izabran je ovaj tip mapiranja zbog preglednosti log podataka u bazi.

Neo4j model podataka je prikazan na sledećoj slici:

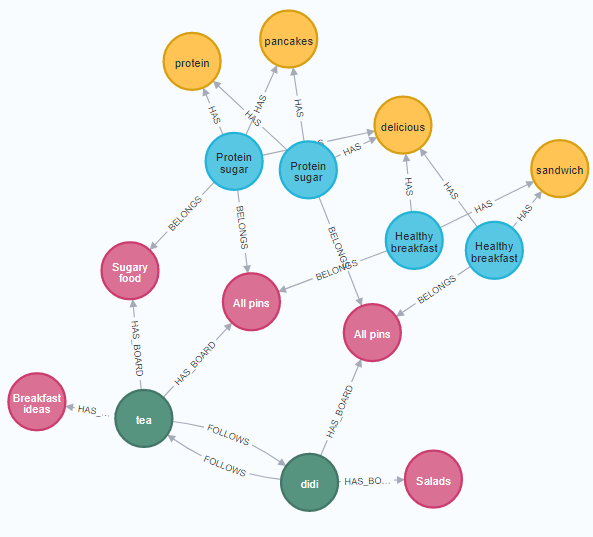


Kao što je već navedeno neo4j graf baza podataka je implementirana da bude primarno skladište podatka. Na slici se vide tipovi čvorova u bazi i njihove medjusobne veze.

Kratak opis neo4j modela:

* User čvor čuva podatke konkretne korisniku, kao što su username, password, ime, prezime i slični atributi.
* Board čvor sadrži objave (Pinove) u sebi i predstavlja medijum izmedju pinova i korisnika. Prednost korišćenja board-ova je organizacija pinova radi lakšeg korišćenja aplikacije.
* Pin čvor sadrži sve neophodne podtke vezane za jednu objavu na drušvenoj mreži (naziv, opis, sliku itd). Jedan pin pripada samo jednom useru
* Tag čvor služi za dodatno opisivanje pinova kako bi se omogućio sistem preporuke sličnih pinova.
* Veza Follows izmedju usera oznacava da jedan user prati drugog usera
* Veza Has-Board izmedju usera i board oznacava da user ima taj board na svom profilu i moze u njemu da skladisti svoje pinove, i čuva tudje.
* Veza Belongs izmedju pina i board-a oznacava da pin mora pripadati nekom board-u.
* Veza Has izmedju pina i taga oznacava da pin mora da ima tagove zbog sistema preporuke.

Primer podataka u neo4j bazi:



# Mehanizam mapiranja

Za neo4j bazu podataka je iskorišćena biblioteka Neode koja omogućava lakše kreiranje šeme graf baze, olakšava pisanje CYPHER upita. Korišćenjem Neode biblioteke implemenitran je Repository pattern kako bi se kreirao sloj perzistencije u projektu i omogucio jedinstven interfejs sloju biznis logike.   
Repository pattern je implementiran instanciranjem jedne instance neo4j servera u projektu u fajlu Persistance/neo4j/config.js odakle se includuje ta instanca u DataProvider klasama gde se izvrsavaju upiti ka neo4j bazi.

Za MySQL bazu iskorišćena je biblioteka Sequelize. To je ORM za node js koja omogucava rad sa mysql i drugim relacionim bazama podatkaka. Iskrošćena je ista logika implementacije repository pattern-a. Sequelize instanca povezana sa mysql bazom se nalazi u config fajlu na putanju Persistance/mySql/config/mySql-config.js. I ovde su implementirane dataProvider klase za izvsavanje upita.