

Poznámka

S vývojom aplikácie som začal príliš neskoro, čo sa podpísalo na výsledku. Snažil som sa najskôr naplniť požiadavky predmetu DBS, a preto nie sú niektoré z plánovaných funkcií aplikácie implementované. Takúto funkcionality v dokumentácii označím (TBI). O grafickom dizajne ani nehovorím (napr. v zobrazení dlaždíc receptov chcem mať obrázky, ale na to potrebujem vyriešiť úložisko obrázkov , ktoré si nechávam nakoniec.)

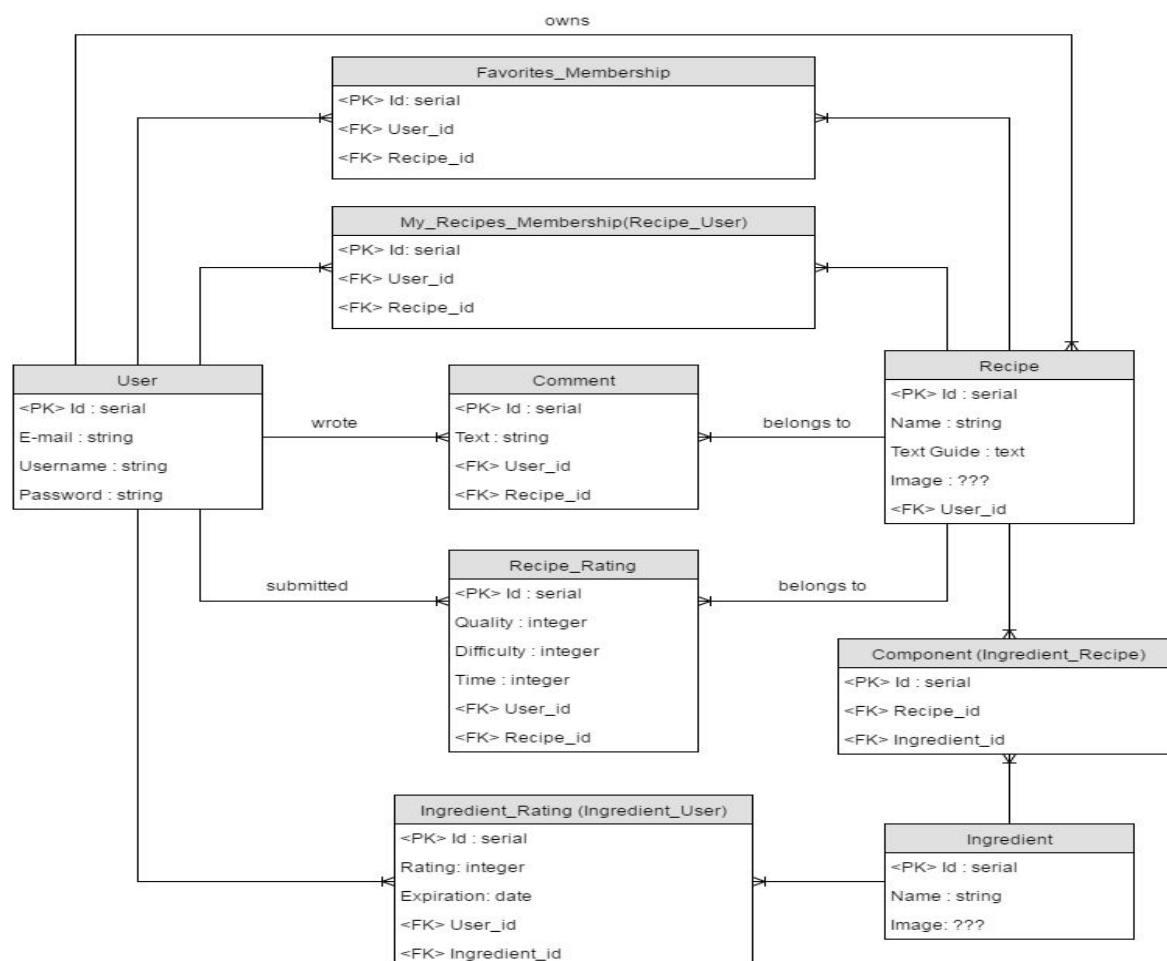
Prehľad

Aplikácia dovoľuje používateľom vytvárať online recepty, navzájom si ich hodnotiť a (TBI)komentovať. Zároveň dokáže (TBI) používateľovi odporučiť recept na základe jeho výberu ingrediencie.

Scenáre

- vytvorenie záznamu - users#create - vytvorí užívateľa, recipes#create - vytvorí recept
- aktualizácia záznamu - users#modify - modifikácia hesla
- vymazanie záznamu - recipes#destroy - vymazanie receptu a jeho spojenia s ingredienciami, používateľmi a (TBI)komentármi
- zobrazenie prehľadu viacerých záznamov - recipes#browse - viacero receptov
- zobrazenie konkrétneho záznamu - recipes#show - jeden konkrétny recept
- filtrovanie záznamov - recipes#browse, recipes#my_recipes (mnou vytvorené) - viacero receptov
- nerelačná databáza - Elasticsearch - fulltext vyhľadávanie receptov podľa názvu, zoznamu, ingrediencií - recipes#browse

Dátový model



Prvotné entity sú User - používateľ, Recipe - recept a Ingredient - surovina.

Recipe patrí Userovi, ktorý ho vytvoril.

Component => Ingrediencia je jednou zo surovín pre recept.

Favorites_Membership => Indikuje, že užívateľ má recept v obľúbených.

My_Recipes_Membership => Indikuje, že užívateľ má recept v zozname na vyskúšanie.

Comment => Komentár pre recept od používateľa.

Recipe_Rating => Hodnotenie receptu používateľom.

Ingredient_Rating => Hodnotenie suroviny používateľom.

Frontend

HTML stránky s prevažne statickým obsahom

- zobrazenie prehľadu receptov
- zobrazenie konkrétneho receptu
- pridanie receptu

Backend

Webová aplikácia v Ruby-on-Rails ,PostgreSQL databáza, databáza Elasticsearch.

Implementácia

Na vykonávanie SQL je použitá metóda `ActiveRecord::Base.connection.execute` a `Class.find_by_sql`, pričom Class je dedič od `ActiveRecord::Base`.

Ďalšie podmienky

- JOIN - napr. `Application#browse`
- agregáčná funkcia - `recipes#show` (priemer používateľských hodnotení)
- GROUP BY - `Application#browse`
- transakcia - `recipes#create`