Dokumentacija za bazu podataka aplikacije „Spiza”

Martina Gaćina, Manuela Pleša, Fran Vojković, Alen Živković

U Visu, 16. svibnja 2020.

Sadržaj

1 Modeliranje **1**

2 Relacijski model baze **2**

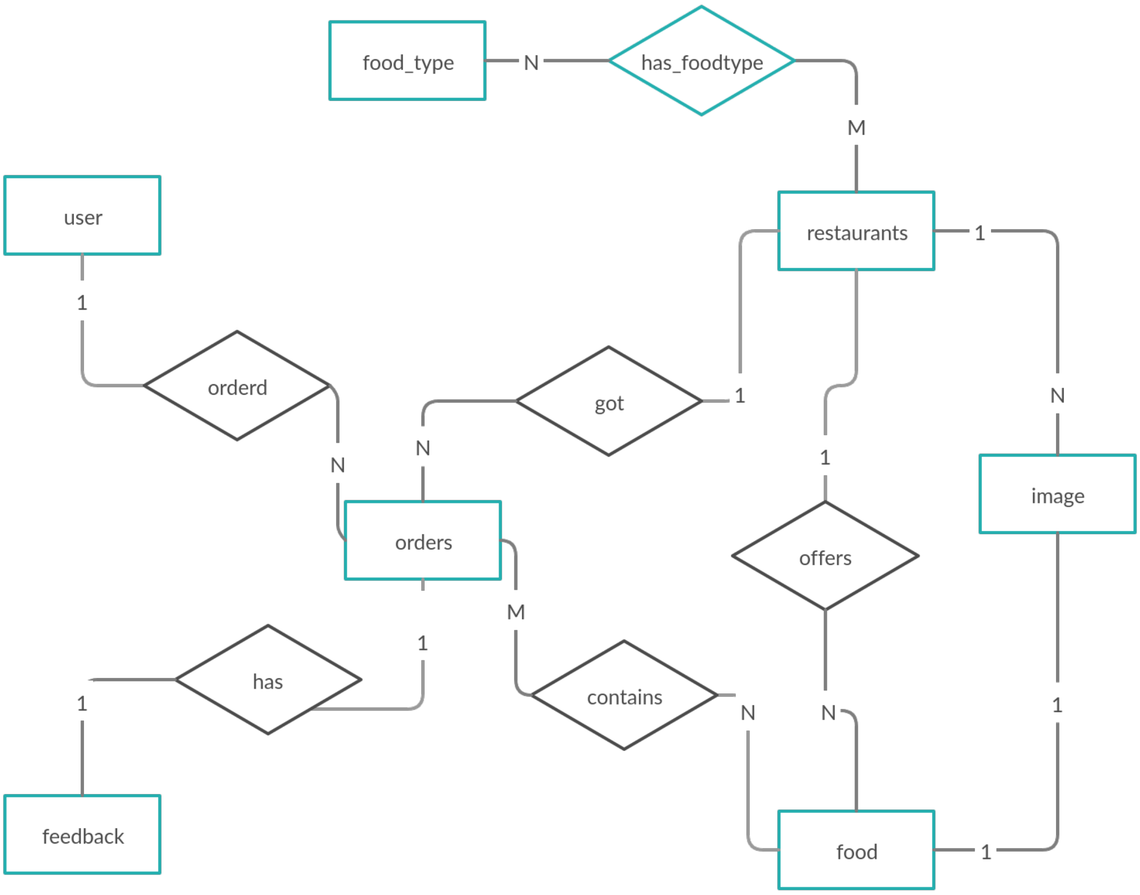
3 Implementacija modela **3**

4 To do list **5**

Napomena 1. Javite ako nesto treba mjenjat u dokumentu. ||

1 Modeliranje

Za potrebe aplikacije uočili smo da nam je potrebno čuvati podatke o korisnicima, restoranima, nerudžbama, hrani koju restorani imaju u ponudi i povratnoj informaciji korisnika o kvaliteti. Koristimo MySQL bazu podataka. Za svakog korisnika imamo sljedeće podatke koje pamtimo: id_user, username, password_hash, email, registration_sequence, has_registered. Navedeni podaci potrebni su nam za registraciju korisnika te *log in* korisnika, primarni ključ predstavlja **id_user**. Svaki restoran ima sljedeće podatke: id_restaurant, password_hash, email, registration_sequence, has_registered, name, address, description. Navedeni podati potrebni su za registraciju novih restorana, *log in* postojećih restorana te prikaza opisa restorana, primarni ključ je **id_restaurant**. Potrebno je pohranit i podatke o jelima: id_food, name, description, waiting_time, price. Također pohranjujemo *feedback* korisnika za svaku narudžbu. Kako će restorani imati mogućnost dodavanja novih jela te njihovih slika, potrebno je pohraniti *upload*-ane slike na server te u bazi zapisati njihovu lokaciju. Alternativno moguće je pohraniti slike direktno u bazu no naved3eno narušava sigurnost same baze na serveru. Trebamo se odlučit za jednu opciju *TBD* . . .



Slika 1: ER shema modela

2 Relacijski model baze

Na slici 1 prikazan je relacijski model naše baze, boldano su označeni primarni ključevi entiteta(tablica), podcrtani su strani ključevi u entitetima(tablicama).

Vezu 1:N *ordered* rješavamo tako da u tablicu orders stavimo ključ *user*-a kao strani ključ. Analogno rješavamo veze 1:N *offers* i *got*. Veza *has* je tipa 1:1 pa ubacujemo ju u orders tablicu kao atribut, a veze tipa N:M realiziramo kao posebne tablice sa primarnim ključem iz pripadajućih tablica.

Slijedi prikaz relacijskog modela:

USERS (**id_user**, username, password_hash, email, registration_sequence, has_registered)

RESTAURANTS (**id_restaurant**, username, password_hash, email, registration_sequence, has_registered, name, address, description)

FOOD (**id_food**, name, description, waiting_time, price, in_offering, id_restaurant, image_path)

FOOD_TYPE (**id_foodType**, name, image_path)

ORDERS (**id_order**, id_user, id_restaurant, active, order_time, delivery_time, lastchange_time, price_total, discount, note, feedback, rating, thumbs_up, thumbs_down)

CONTAINS (**id_order**, **id_food**, quantity)

HAS_FOODTYPE (**id_foodType**, **id_restaurant**)

IMAGE (**id_image**, name, id_restaurant, image - *vjerojatno se neće koristit- za pohranu slike*)

NEIGHBORHOOD (**id_neighborhood**, id_restaurant)

3 Implementacija modela

Pomoću sljedećih naredbi kreiramo bazu.

```
CREATE TABLE IF NOT EXISTS spiza_users(
id_user int NOT NULL PRIMARY KEY AUTO_INCREMENT,
username varchar(50) NOT NULL,
password_hash varchar(255) NOT NULL,
email varchar(50) NOT NULL,
registration_sequence varchar(20) NOT NULL,
has_registered int
)

CREATE TABLE IF NOT EXISTS spiza_restaurants (
id_restaurant int NOT NULL PRIMARY KEY AUTO_INCREMENT,
username varchar(50) NOT NULL,
password_hash varchar(255) NOT NULL,
email varchar(50) NOT NULL,
registration_sequence varchar(20) NOT NULL,
has_registered int,
name varchar(50) NOT NULL,
address varchar(80) NOT NULL,
description varchar(50) NOT NULL
)

CREATE TABLE IF NOT EXISTS spiza_food (
id_food int NOT NULL PRIMARY KEY AUTO_INCREMENT,
name varchar(50) NOT NULL,
description varchar(200) NOT NULL,
waiting_time int NOT NULL,
price decimal(6,2) NOT NULL,
in_offering tinyint NOT NULL,
id_restaurant int NOT NULL,
image_path varchar(200),
FOREIGN KEY (id_restaurant) REFERENCES spiza_restaurants(id_restaurant) )

CREATE TABLE IF NOT EXISTS spiza_food_type (
id_foodType int NOT NULL PRIMARY KEY AUTO_INCREMENT,
name varchar(30) NOT NULL,
image_path varchar(200) )

CREATE TABLE IF NOT EXISTS spiza_orders (
id_order int NOT NULL PRIMARY KEY AUTO_INCREMENT,
id_user int NOT NULL,
id_restaurant int NOT NULL,
active tinyint NOT NULL,
order_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
delivery_time TIMESTAMP,
lastchange_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
price_total float,
discount float,
note varchar(50),
feedback varchar(100),
rating float,
thumbs_up int,
thumbs_down int,
FOREIGN KEY (id_restaurant) REFERENCES spiza_restaurants(id_restaurant),
FOREIGN KEY (id_user) REFERENCES spiza_users(id_user)
)

CREATE TABLE IF NOT EXISTS spiza_contains (
id_order int NOT NULL,
id_food int NOT NULL,
quantity int NOT NULL,
PRIMARY KEY (id_order, id_food),
FOREIGN KEY (id_order) REFERENCES spiza_orders(id_order),
FOREIGN KEY (id_food) REFERENCES spiza_food(id_food)
)

CREATE TABLE IF NOT EXISTS spiza_has_food_type (
id_foodType int NOT NULL,
id_restaurant int NOT NULL,
PRIMARY KEY (id_foodType, id_restaurant),
FOREIGN KEY (id_restaurant) REFERENCES spiza_restaurants(id_restaurant),
FOREIGN KEY (id_foodType) REFERENCES spiza_food_type(id_foodType)
)

CREATE TABLE IF NOT EXISTS spiza_image (
id_image int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
name varchar(200) NOT NULL,
image longtext,
id_restaurant int,
FOREIGN KEY (id_restaurant) REFERENCES spiza_restaurants(id_restaurant),
)

CREATE TABLE IF NOT EXISTS spiza_neighborhood (
id_neighborhood int NOT NULL PRIMARY KEY AUTO_INCREMENT,
id_restaurant int NOT NULL
)
```

4 To do list

Potrebno je još dodat ~~slike restorana i jela u restoranima u bazu~~ te razradit sam sistem čuvanja slika na serveru, dodat za dostavljače i vjv ima još nešto.