

DBMS Project Description and Design

2018 年 11 月 5 日

18:43

Project Description:

Online food ordering web application. There are 5 types of users in our application: Deliverer, Customer, Owner, Admin and Anonymous user.

Anonymous user can browse different restaurants based on different selected cuisines, corresponding reviews, see rating of a certain restaurant. When he/she wants to make an order, he/she must sign up as a customer.

Customer can only add and modify different kinds of dishes with different quantities into his/her shopping cart from a same restaurant. After he/she confirms all goods in the shopping cart, an order will be generated and the shopping cart will be cleared. The current status of this order is New.

Then the restaurant owner can see all orders customers placed and confirm receiving this order. Then the status of this order will become Hold. After finish preparing, he will choose deliverers to deliver dishes based on the rating of them. At this point, the status of this order becomes Shipped.

Once a deliverer is assigned with a delivery, he/she cannot accept other deliveries until he/she finishes the current delivery.

Once the customer received the package, he/she will confirm that this order is delivered. The status of this order will become Delivered. Customer can write reviews for this order and rate the restaurant and the deliverer. Restaurant owner can reply to received reviews. A Restaurant's rating and a customer's rating is the average rating of all previous ratings they received.

Admin can crud all information of all signed up users.

Domain objects:

Restaurant, Cuisine, Dish, Review, Description Picture, Shopping Cart, Order, Order Status

Potential human users:

Restaurant owner, Customer, Deliverer, Administrator

Restaurant owner goals:

1. Post his/her restaurant information online
2. Sell dishes online
3. Have prepared dishes delivered to customers

Customer goals:

1. Order dishes online
2. Confirm receiving ordered dishes
3. Write reviews for orders
4. Follow restaurants
5. View past orders

Deliverer goals:

1. Accomplish a delivery task

2. Mark the current delivery as finished and be ready for the next delivery

The relationship between different users:

The relationship between Restaurant owner and Customer

1. Restaurant owner can reply to customer's reviews

The relationship between Restaurant owner and Deliverer

1. Restaurant owner can assign delivery tasks to deliverers based on the rating of them

The relationship between Customer and Deliverer

1. Customer can rate deliverers' jobs

The relationship between User and Domain Objects:

The relationship between Restaurant owner and Restaurant:

1. A restaurant owner can have zero or multiple restaurants
2. A restaurant owner can crud description pictures of his/or restaurants
3. A restaurant owner can crud restaurants' information such as the location of one restaurant, the cuisine of one restaurant and the description of one restaurant. But he CANNOT crud the rating of one restaurant.

The relationship between Restaurant owner and Dish:

1. A restaurant owner can crud one dish's description picture, name, description, price and available amount.

The relationship between Restaurant owner and Order

1. Once the newly generated order is confirmed by a restaurant owner, the status of order will become **Hold**. If a restaurant owner has goods packaged and designates this package to a deliverer, owner can mark the order status as **Shipped**.
2. A restaurant owner can reply reviews of one order.

The relationship between Customer and Restaurant

1. A customer can follow/like zero or multiple restaurants.
2. A restaurant could be followed by zero or multiple customers.

The relationship between Customer and Order

1. A customer can make zero or multiple new orders, the order status should be **New** until this order is held by a restaurant.
2. A customer can write reviews for an order, rate the restaurant and the deliverer.
3. After customer confirms that he/she has received the package, the order status will become **Delivered**

The relationship between Customer and Shopping Cart

1. A customer can crud different kinds of dish with different amounts in his/her shopping cart.

The relationship between Deliverer and Order

1. Deliverer's status is free until he/she is assigned with an order. Then the status will become busy and he/she cannot be assigned with a new order.
2. After he/she finished this order, the status will become free again.

Relationship between Domain Objects:

The relationship between Order and Order Item

1. An order can have multiple order items

The relationship between Shopping Cart and Shopping Cart Item

1. A shopping cart can have multiple shopping cart items

The relationship between Order Item, Shopping Cart Item and Item

1. An order item is an Item
2. A shopping cart item is an Item

The relationship between Order and Review

1. An order can have zero or multiple reviews

The relationship between Order and Restaurant

1. A restaurant can have zero or multiple orders

The relationship between Item and Dish

1. One Item can only have one kind of dish with multiple quantities.
2. One kind of dish can appear in zero or multiple items in different users' shopping carts and orders.
3. **Constraint:** One kind of dish can only appear once in one shopping cart and one order.

The relationship between Restaurant and Dish

1. A restaurant can have zero or multiple kinds of dish

The relationship between Restaurant and Description Picture

1. A restaurant can have zero or multiple description pictures

The relationship between Dish and Description Picture

1. A kind of dish can have zero or multiple description pictures

API used:

Yelp Fusion

The Yelp Fusion API provides local business contents and user reviews, which is used in our project as the data source. Some data are pre-fetched and loaded to provide real use cases, while other data can be fetched on users' requests (for instance searching for a restaurant). After user choose one certain restaurant, use this API to fetch all information of this restaurant (such as pictures, reviews, rating, etc.) and store parts of those information in our database. Then we can modify these data based on our needs.

