

Spring 2016 CS157B: Programming Assignment 1 (Hibernate ORM)

On-line Pizza Order System Instructor: Dr. Kim

In this individual assignment, you are developing a simple pizza order system in Java. MySQL database is the back-end database and Hibernate is going to be used for ORM. With this system, we assume a customer completely manages his/her orders.

A customer signs up for the system. The customer chooses user name and password and enters address information (street name, city, state and zip code) to sign up for the system. Have Hibernate generate an id for each customer (i.e. `userId` is a surrogate primary key). Also, make the user name uniquely identify a customer (i.e. user name is unique - use `@Column(unique = "true")`.)

In order to make/change/cancel an order, the customer has to login in with the user name. An order consists of pizza size, toppings of choice, delivery date and time (use `TimeStamp` type), payment method and total price. The order is completed after the customer makes a payment. A discounted order is a special order with a discounted rate of 10%. The total price of a discounted order should reflect the discount rate.

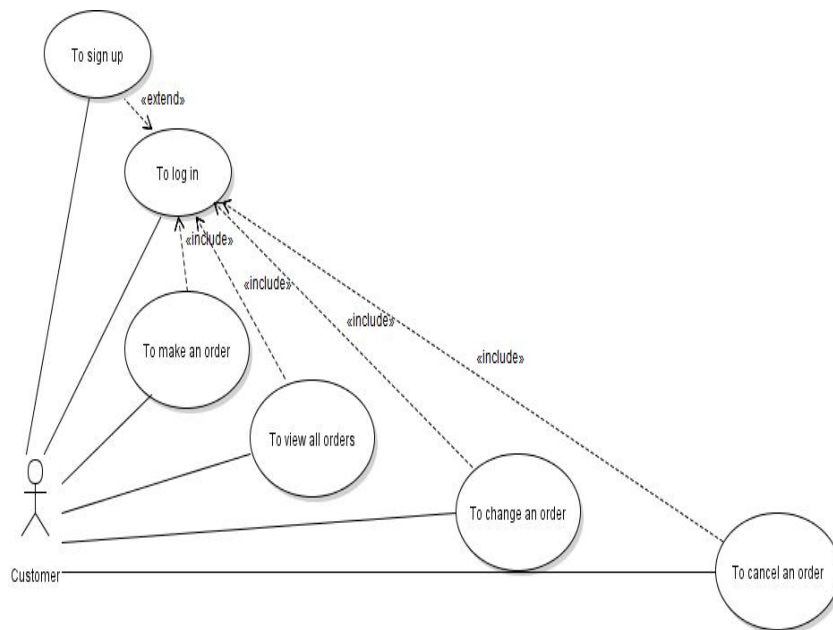
The customer can view all the orders under the user name.

Use Java Enum type to represent `PaymentType` and `PizzaSize`. There are three different payment methods accepted by the system: CASH, VISA and MASTER. Three different pizza sizes available for a customer can choose from: SMALL, MEDIUM, and LARGE. In the enum type `PizzaSize`, define a price of each size as an instance variable and also define the corresponding accessor and mutator. For instance, set a price in a way that a SMALL pizza costs \$3, a MEDIUM pizza costs \$5 and a LARGE pizza costs \$7.

Make the following 10 toppings available to the system.

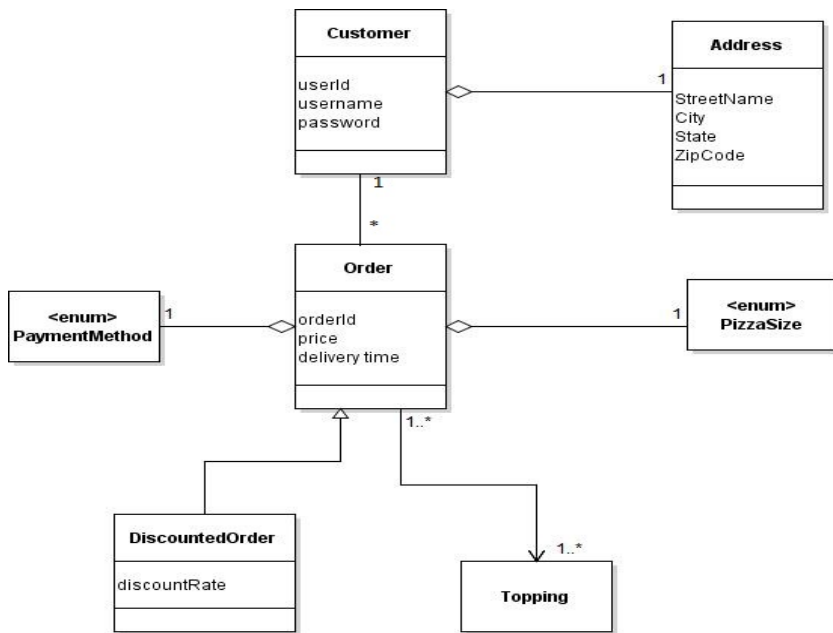
Pepperoni, Mushrooms, Onions, Sausage, Bacon
Extra cheese, Black olives, Green peppers, Pineapple, Spinach

A customer can add upto 3 different or same toppings to the order. Examples of a combination of toppings are (Pepperoni), (Pepperoni and Onions), (Pepperoni, Onions and Green peppers), and (Pineapple, Pineapple, Sausage). The `Topping` class defines an instance variable price for this `Topping` instance. Assign some reasonable price to it. The following UML use case diagram depicts use cases of this pizza order

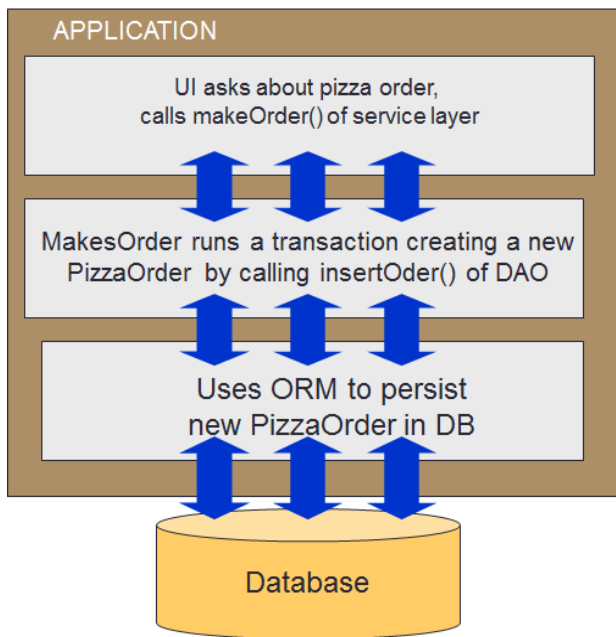


system.

The relationships among required classes are depicted in the following class diagram.



Respect three layers of database application: presentation, service and DAO layers. The expected layers of the application is shown below:



- The presentation layer can be console-based (without using GUI). The presentation layer prompts a menu for a customer to choose one of the 6 functions. indented by the use cases.
- The service later defines 6 methods for the required functions (e.g. `makeOrder`).
- It is your job to identify DAO methods that support functions in the service layer.

Deliverable

1. Soft-copy: Due Sunday, March 13 23:59

Zip all deliverable into `hw1.zip` and submit it through our course web site.

- `PresentationLayer.java`: with `main()` using console interface
- `ServiceLayer.java`: Program that implements the business logic of this medical system. This `ServiceLayer` program will use functions defined in the DAO.
- `PizzaShopDAO.java`: Interface that defines methods of DAO
- `ConcretePizzaShopDAO.java`: Concrete implementation of `PizzaShopDAO`
- All Java classes required to run the application
- `hibernate.cnf.xml` where all required model classes are listed.

2. Hardcopy: Due Tuesday, March 15 in class

7 sets of screenshots. Each set consists of a screenshot of input from the presentation and the table contents to show the result of the function.