Project Part 5: Final Report

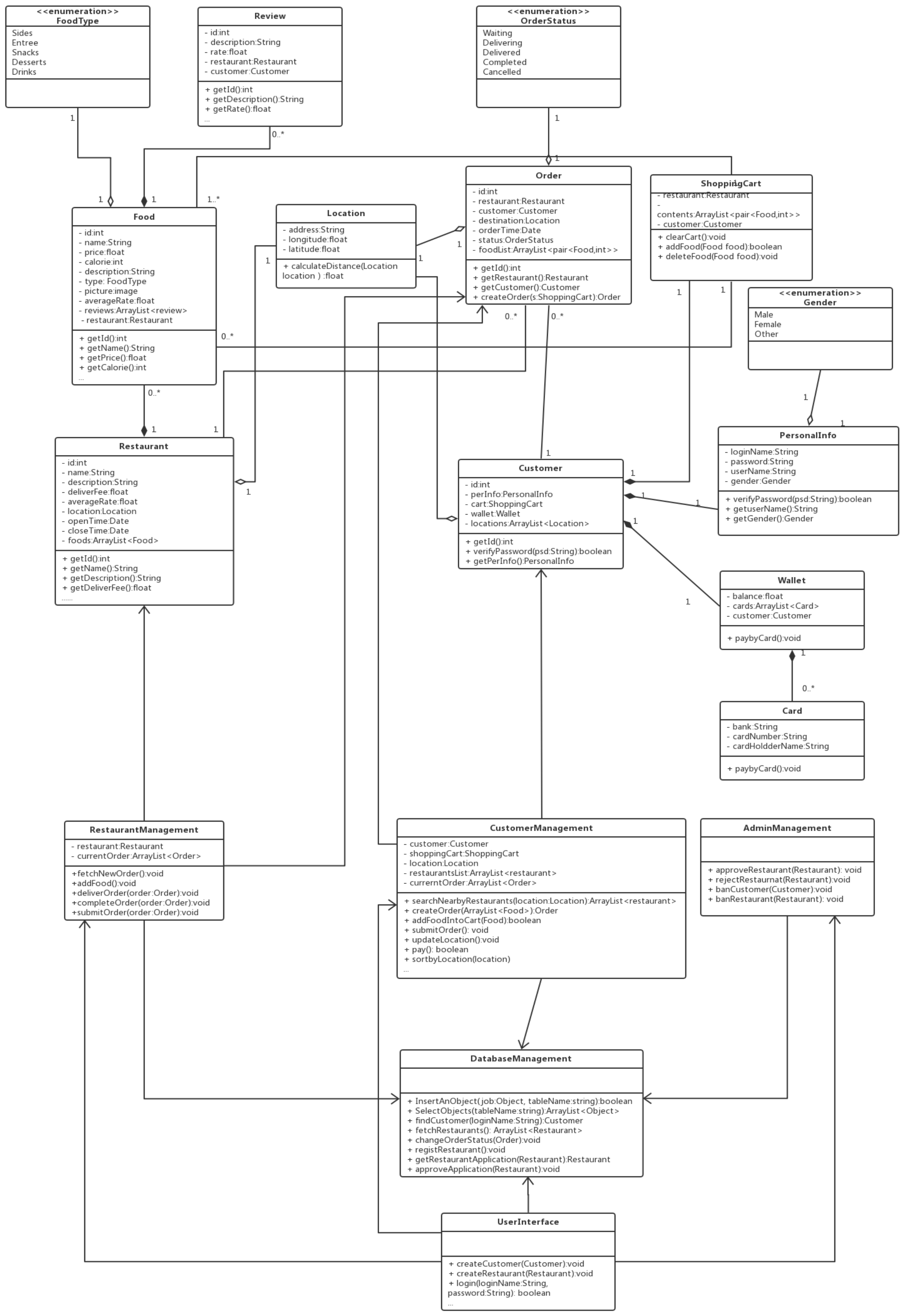
1. The features that were implemented (The ID in the below table is coordinated with the ID in our Part2's user requirements ID and use cases ID)

ID	Title
UR-1, UC-10	Customer Register
UC-11	Customer Log In
UR-2, UC-12	Customer Update Information
UR-3, UR-4, UC-01	Customer View Nearby Restaurants
UR-5, UC-02	Customer View Foods Menu of a Restaurant
UR-6, UC-03	Customer Add Food to Shopping Cart
UR-7, UC-04	Customer Create Order
UR-8	Customer View Order Detail
UR-11, UC-13	Customer Add Debit/Credit Card
UR-13, UC-16	Restaurant Update its Information
UR-14, UC-05	Restaurnat View its Orders
UR-15, UC-05, UC-06	Restaurant Change Status of an Order
UR-16, UC-17	Restaurant Add a Food to its Menu
UR-16, UC-18	Restaurant Update Information of a Food
UR-16, UC-19	Restaurant Delete a Food From its Menu
UR-18, UC-20	Admin Ban a Customer

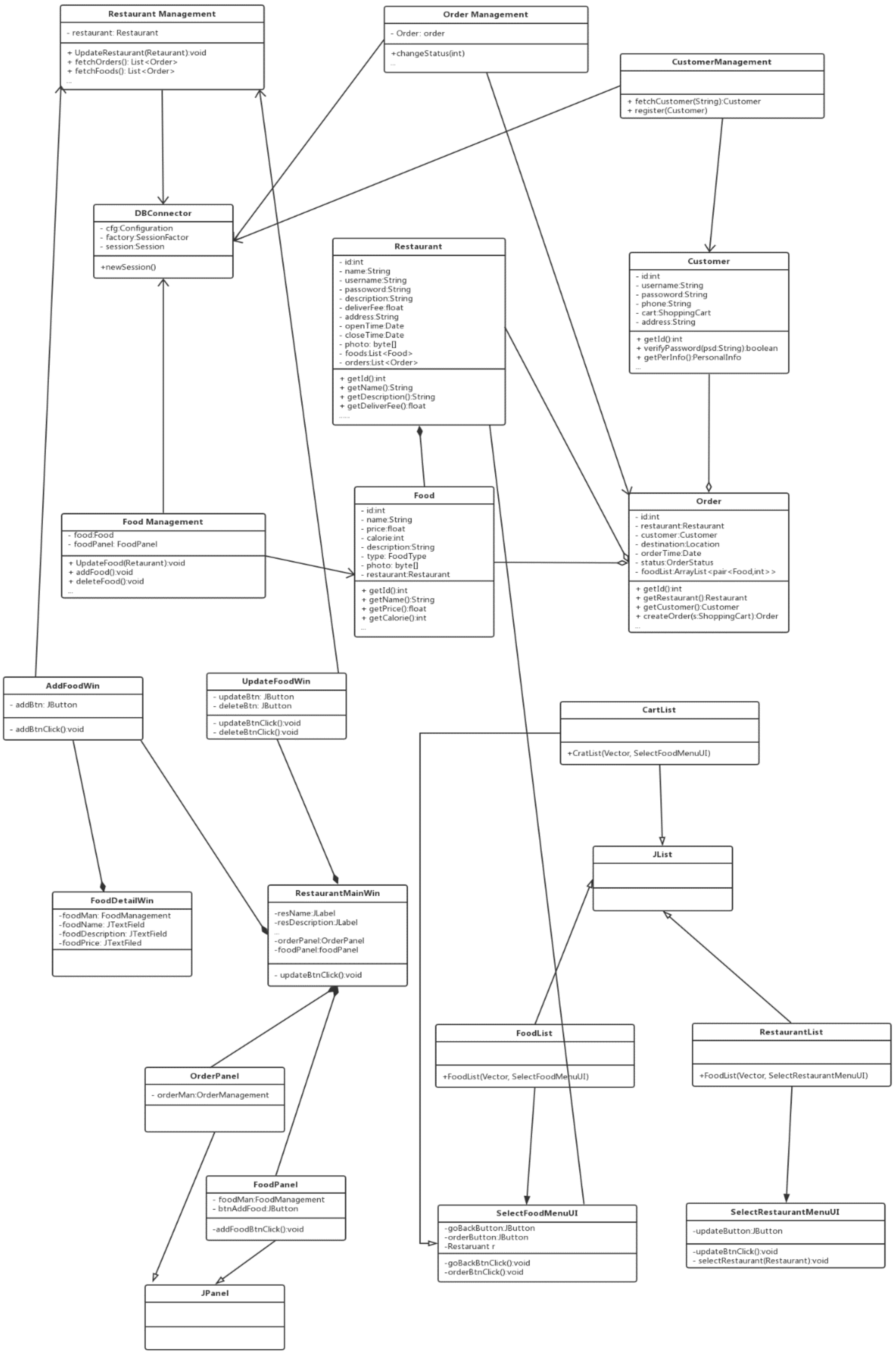
2. The features were not implemented.

ID	Title
UR-9	Customer Give a Review to a Restaurant
UR-10	Customer View Other's Reviews
UR-12, UC-14	Restaurant Register
UR-17,UC-15	Admin Approve a Restaurant's Register Quest
UR-17,UC-21	Admin Ban a Restaurant

3. Class Diagram In Part 2

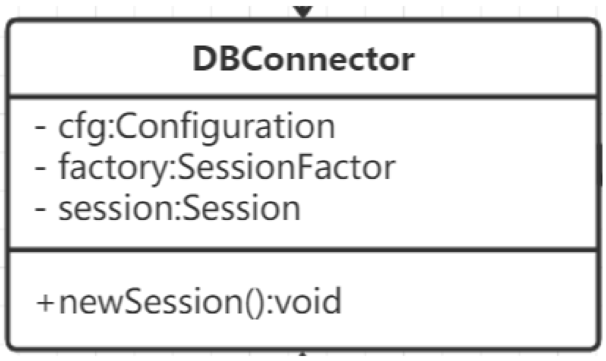


Final Class Diagram:



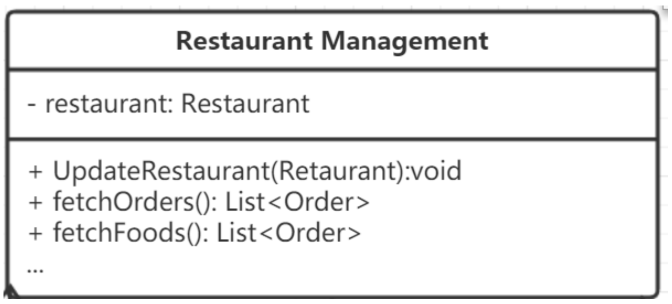
For the controller, we abandoned the DatabaseManagement class because we used Hibernate when we worked on our project but we didn't think of that when we designed it. Also, some method has been adjusted. For the model, we didn't implement some functions, like customers reviewing the restaurants, so we threw away the related classes. And the main difference is the view part, we didn't design this part in details since we haven't decided which tool would be used to implement the GUI. For the reason that it's neither only a single window in customer's side or restaurant's side, so several new classes are needed to implement the whole user interface.

4．Singleton:



This is the class that initialize the configuration of hibernate and open a session connect to database. In order to ensure there's only one session exist in the whole program, we implement the Singleton design pattern.

Proxy:


Restaurant Management
- restaurant: Restaurant
+ UpdateRestaurant(Retaurant):void
+ fetchOrders(): List<Order>
+ fetchFoods(): List<Order>
...

This is the class which provide a series of database writing and database reading methods. So it's applied to Proxy design pattern.

5． The most important thing we found is that the design period is the most significant part for the whole project. It will not only save a bunch of time in the implementing process, but also improve the quality of the final result. We have to admit that we did not do a prefect job at our design period. For example, the view part is lack of consideration. So the view part is totally new. It's actually designed when we implemented it. And some of the controller method are designed unreasonably. So when we came to implement it, we found that some changes of the design have to be made in order to achieve the planned effect. We believe that all our group members are solid programmer, but maybe we are short of design to some extent. But it's a very important ability for future study and works, so we should make effort to improve it. And this 3-month process of creating, designing and implementing this Food Delivery System project is a perfect education for all of us.