1. Features that were implemented
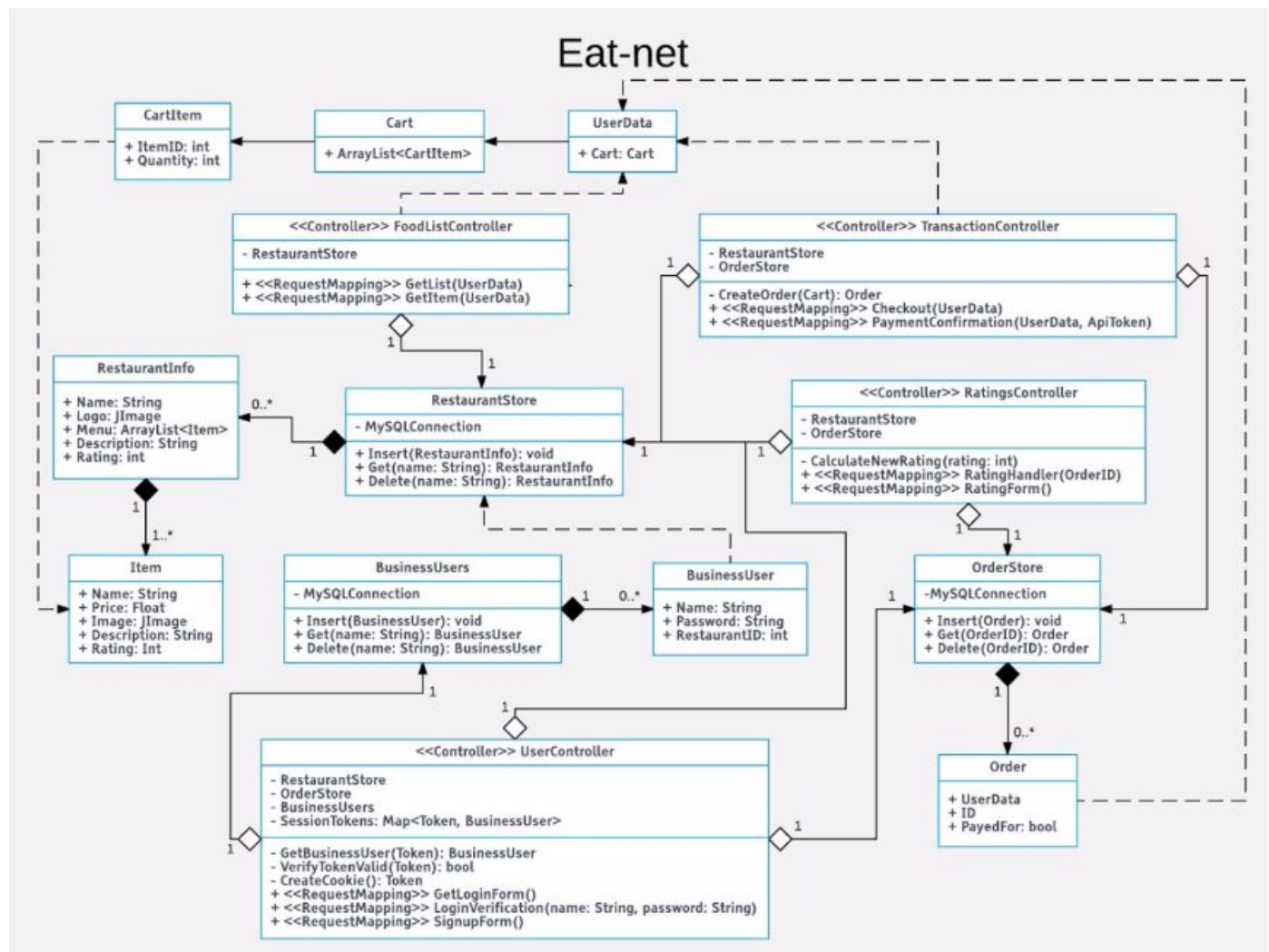
Feature ID	Title
UR-1	Each business user can login to corresponding restaraunt admin page
UR-2	Each business user can sign up
UR-3	Each business user can specify restaurant information (menu, prices, description, address)
UR-4	Add food item to shopping cart
UR-5	Review items in shopping cart
UR-8	Being able to view orders from customers
UR-9	Rate restaurants and items
NFR-2	Rate restaurants and items

2. Features that were not implemented

Feature ID	Title
BR-1	Add advertisements to the site to generate revenue
BR-2	Being able to view site usage
BR-3	Being able to pay fees to Paypal
UR-6	Pay for items in shopping cart
UR-7	Generate monthly report for each business
NFR-1	Payment Encryption

3. Part 2 Class Diagram vs FInal Class Diagram
   a. Part 2 Diagram:

Eat-net

b. Final Diagram:

[Insert Here]

    c. What's different and Why
        i.
4. Design Patterns Used
    a. Observer
        i. Diagram:

[Insert Here]

        ii. How we made use of this pattern
            1.
    b. Repository
        i. Diagram:

Restaurant
- 🖳 Items: Set<item>: undefined
- 🖳 id: long
- 🖳 name: String
- 🖳 Restaurant(): String
- 🖳 Restaurant(): undefined
- 🖳 getId: long
- 🖳 getName(): String

Item
- 🖳 id: long
- 🖳 name: String
- 🖳 restaurant: Restaurant
- 🖳 Item(): undefined
- 🖳 Item(String): undefined
- 🖳 getId(): long
- 🖳 getName(): String

1   1..*

◆1..*   ◆1..*

1   1

RestaurantDao
- 🖳 findByName(): String
- 🖳 findAll(): ArrayList<Restaurant>

ItemDao
- 🖳 findByName(String): undefined
- 🖳 findAllByRestaurant(Restaurant): ArrayList<Item>
- 🖳 findAll(): ArrayList<Item>

◇1   1   ◇1

1

IndexController
- 🖳 restaurants: RestaurantDao2
- 🖳 items: ItemDao2
- 🖳 IndexController(): undefined
- 🖳 index(Model): String

ii.   How we made use of this pattern
         1.
5. What we Learned
    a.