COEN 6312

Model Driven Model Software Engineering

Winter 2016

(Deliverable 3)

Group Name: Neophytes

Project: Online Grocery System

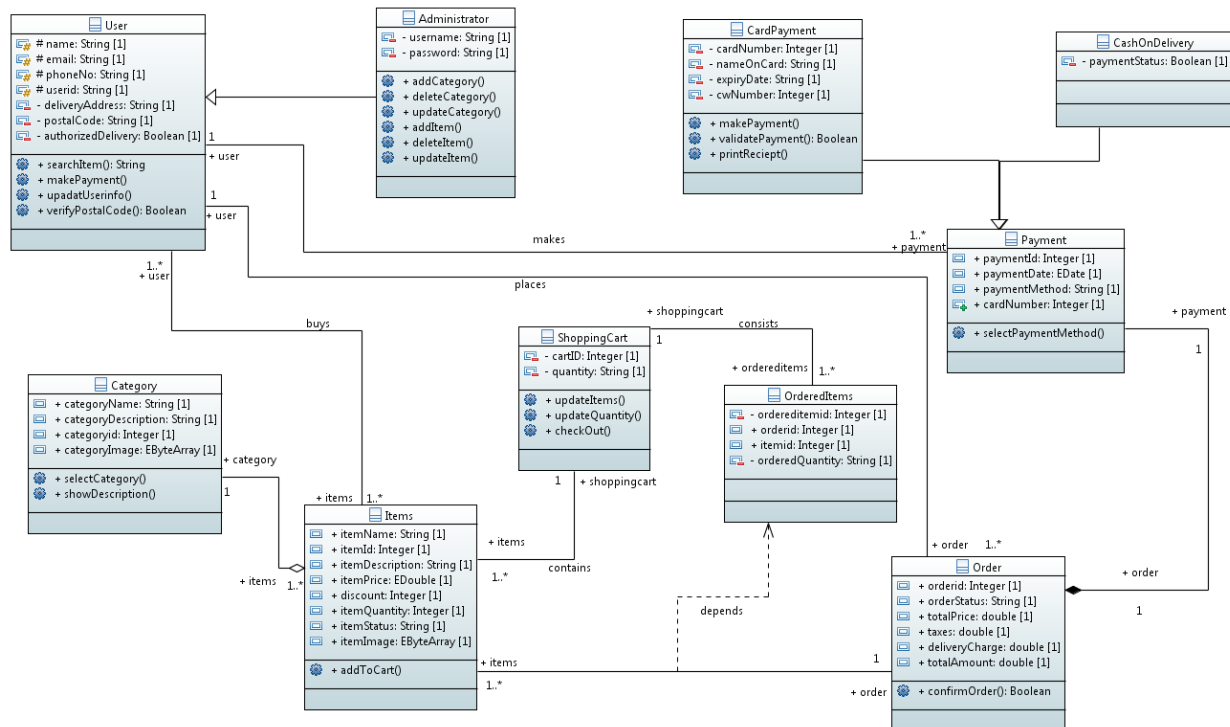Sandeep Kumar (27172486)

Harmeet Singh (27193114)

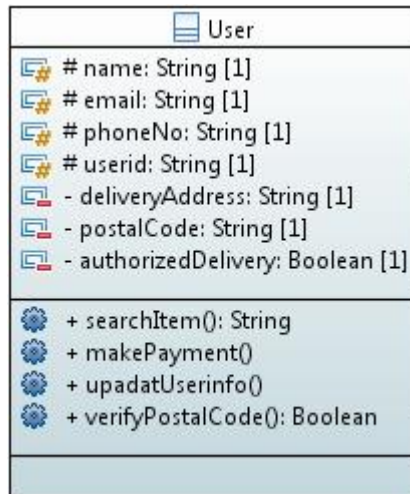Shrijeet Kaduskar (27391773)

Ramandeep Singh Deogan (27394675)

Vikas Yadav (27288700)

Online Grocery System

The online grocery system allows the users to purchase the items online and get it delivered at their door-steps. Depending upon the total bill, there may or may not be delivery charges associated with this type of service. The system allows the users to purchase items listed under various categories. The system allows the user a flexibility while making a payment, there are two payment methods, Card payments and Cash on Delivery. This document contains the class diagram of the system followed by OCL constraints. The detailed class diagram consists of attributes, methods and relationships among the classes such as associations, aggregations, compositions and generalizations.
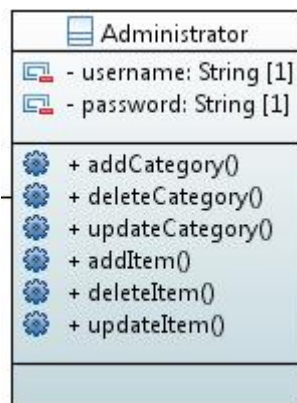
User

| User |
| --- |
| # name: String [1] |
| # email: String [1] |
| # phoneNo: String [1] |
| # userid: String [1] |
| - deliveryAddress: String [1] |
| - postalCode: String [1] |
| - authorizedDelivery: Boolean [1] |
| + searchItem(): String |
| + makePayment() |
| + upadatUserinfo() |
| + verifyPostalCode(): Boolean |

The user class contains specific set of methods which allow users to perform operations such as filling up the user information form and updating it when necessary. User class acts as a superclass for the administrator of the system. User buys the items, places the order and makes the payment.

Administrator

| Administrator |
| --- |
| - username: String [1] |
| - password: String [1] |
| + addCategory() |
| + deleteCategory() |
| + updateCategory() |
| + addItem() |
| + deleteItem() |
| + updateItem() |

The Administrator class is inherited from the user class and is responsible for maintaining the system and perform operations such as adding and deleting categories and items from the system, updating the items for current prices and adding the descriptions.

Category
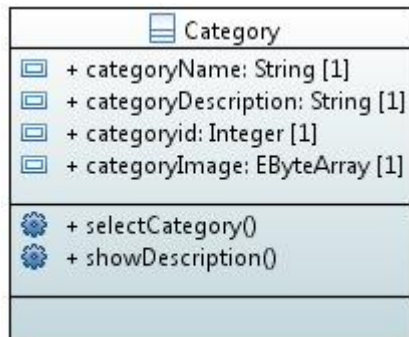
Category class consists of attributes such as name, description and image of the category. It is an aggregation to item class which means that Category class can exist without class Item. The operations associated with the Category class are selection of category and showing the category description. Each category contains 1 or more items.

Items
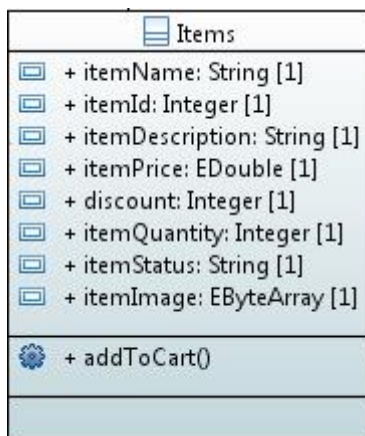
The Item class contains attributes such as name, price, description etc. Item class contains the shopping cart and the items can be added to the cart and each item belong to only one type of Category. Each item has a unique item id associated with the item.

Constraints

1. **Delivery address and postal code cannot be empty.**

   Context User
   inv: self. deliveryAddress ->size()=1 And self.postalcode->size()=1

2. **Phone number is a necessity for the delivery.**

   Context User
   Inv:self.phoneNo ->notEmpty()

3. **Users and user ids are different.**

   Context User
   Inv: self.userid ->forAll(u1,u2:User | u1< >u2 implies u1.userid< >u2.userid)

4. **Minimum order for delivery should be $25.**

   Context Order
   Inv: self.totalAmount >=25

5. **Shopping cart should not be empty.**

   Context ShoppingCart
   Inv: self.quantity>=1

6. **Shopping cart contains all the ordered items.**

   Context ShoppingCart
   Inv: self.consists->includesAll(OrderedItems)

7. **Total Amount must include taxes.**

   Context Order
   Inv: self. totalAmount->includes(self.taxes)

8. **Payment id must be unique.**

   Context User
   Inv: self.makes->(u1,u2:User| u1<>u2 implies u1.paymentId<>u2.paymentId)

9. **If an item is not added to the cart, add item to the cart.**
   Context User :: addToCart(id:itemId):void
   Pre: User.buys->excludes(id)
   Post: User.buys->includes(id)