

## Controller Class

### WholesaleMain

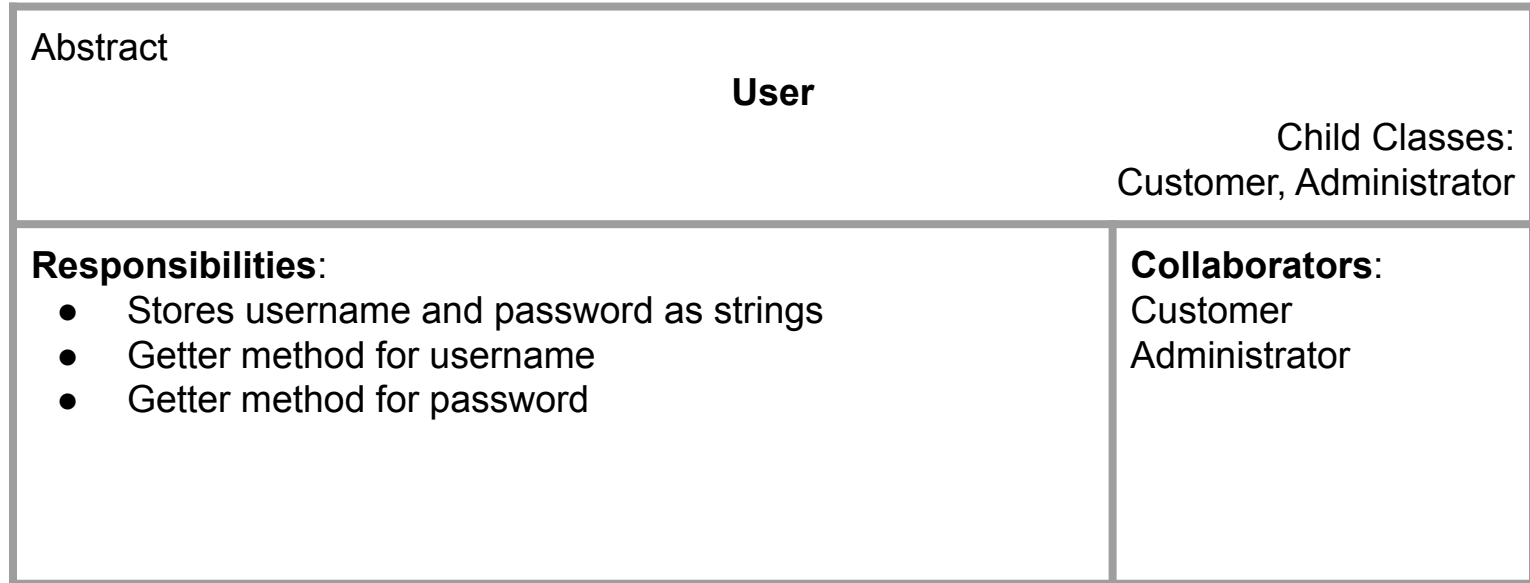
**Responsibilities:**

- Contains the program's main method
- Creates instance of TransactionSystem
- Creates instance of SystemInputOutput
- Calls TransactionSystem's initialize method using the instance of SystemInputOutput as parameter

**Collaborators:**

TransactionSystem  
SystemInputOutput

## Entity Class



## Entity Class

Product	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Stores the name, price, and stock quantity of each product</li><li>• getName, getPrice, getStock methods</li><li>• setStock method for updating stock quantity</li><li>• setPrice method for updating price</li></ul>	<b>Collaborators:</b> Customer Administrator InventorySystem

## Entity Class

<div>Customer</div> <div>Parent Class: User</div>	
<div><b>Responsibilities:</b><ul style="list-style-type: none"><li>• Inherits username and password from User's constructor</li><li>• Each instance of customer has a shopping cart; a hashmap&lt;Product, int&gt;</li><li>• add method which adds an integer quantity of a product to a customer's shopping cart</li><li>• cartTotal method which returns the total cost of all the products in a customer's shopping cart</li></ul></div>	<div><b>Collaborators:</b><div>InventorySystem</div><div>Product</div></div>

## Entity Class

<b>Administrator</b> Parent Class: User	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Inherits username and password from User's constructor</li><li>• Administrators are able to add products to inventory, modify product prices and stock quantities.</li></ul>	<b>Collaborators:</b> InventorySystem Product

## Use Case Class

### InventorySystem

#### Responsibilities:

- Store every product available in an inventory list
- Setter and getter methods for inventory
- createCustomer, createAdministrator, and createProduct methods
- productList method which returns a list of names of products
- availability method which checks if a product name is among the list of product names in the inventory
- findProduct method which returns the product associated with a provided product name
- addToCart method which adds a product of a specified quantity to a customer's cart if that product is available and the quantity specified is less than or equal to the product's stock quantity
- customerTotal method which returns a customer's total cost
- addToInventory method where an administrator adds a specified quantity of a product to the inventory if the inventory does not contain the product; otherwise updates the stock quantity of the product.

#### Collaborators:

Product  
Customer  
Administrator  
TransactionSystem

## Use Case Class

Membership	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Membership number, valid date, expire date, mobile number, payment methods and shipping address.</li><li>• discount method which gives a discount to a customer if the customer has a membership and their shopping cart is nonempty</li></ul>	<b>Collaborators:</b> TransactionSystem

## Controller Class

### TransactionSystem

**Responsibilities:**

- Initialize method which takes InputOutput as parameter
- Decides if a user is a customer or an administrator
- Instantiates a PromptIterator
- Instantiates an InventorySystem
- Takes user inputs and makes InventorySystem create a customer or administrator
- Takes user inputs and makes InventorySystem add products to shopping cart if user is a customer, and adds products to inventory if user is an administrator

**Collaborators:**

InputOutput  
PromptIterator  
InventorySystem



## Interface Adapter (Gateway Class)

### Promptlterator

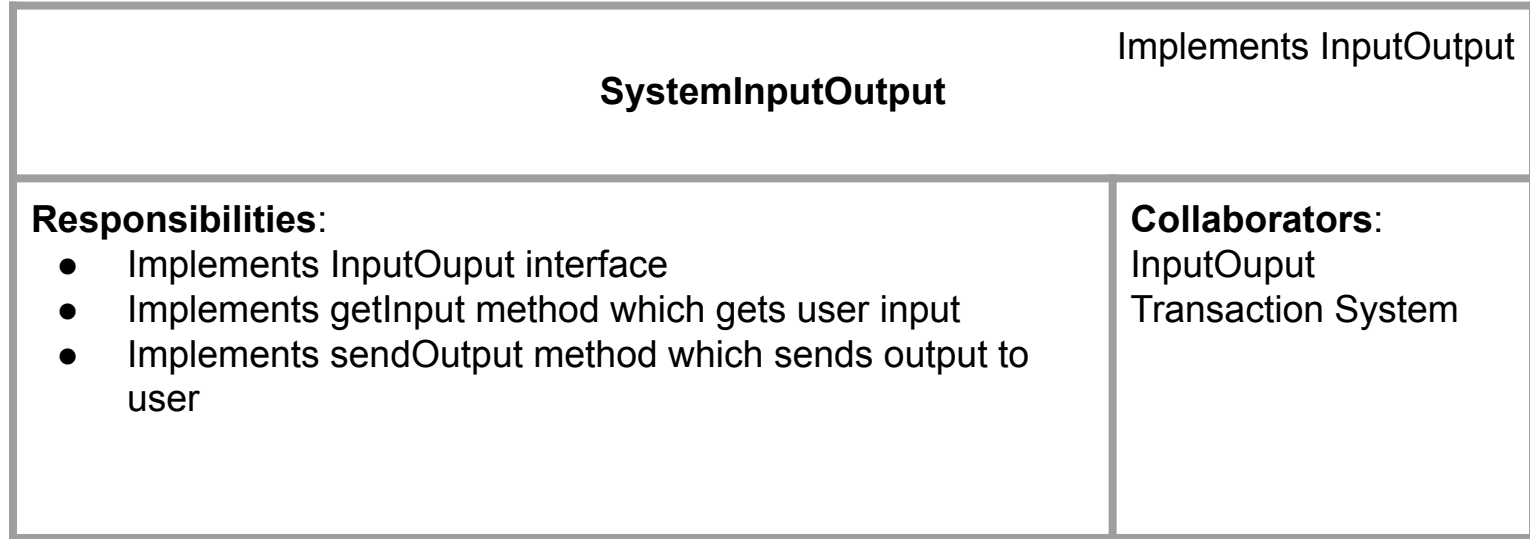
**Responsibilities:**

- Constructor takes a file as parameter
- Reads prompts from the file and adds prompts to a list of strings
- hasNext and next methods to iterate through list of prompts

**Collaborators:**

TransactionSystem

## Interface Adapter



## Interface Adapter

