

Steven Loughran

CMPT 220

Professor Arias

8 May 2019

The classes for this project I have made are Item, Menu, Order, Register, and the Main class named Waiter. All of these work together to give the user which items they are able to choose from and display the prices of each one. I have made the UML for each one which explains exactly the methods and constructors used for each class. I shall talk about what each class will have in them and how they will interact with each other.

My motivation to making this program was due to the up and coming apps integrated with restaurants that customers are able to use when they want to order food.

The customers are able to use an app and order whatever food they want from that restaurant, they then pay with their card and the system will give a time of when their food will be ready.

Using this app would benefit the customers because they can make sure they are ordering the right food and they wouldn't have to deal with calling the restaurant and having to make sure that their order is all correct over the phone.

This system will be an application where a user will be able to order food from a restaurant and by the time they arrive, their food will be ready. They will have many options to choose from for their food including appetizers, entrees, burgers, breakfast sides, and drinks. The user will only be using the class called Waiter. When the program is started, they will go through a list of what they are able to buy, how big the item is, and what the price is that they would have to pay. That main class will call other classes depending on what is called to be used.

In the Item class, we have fields for the double price, string name, string category, and string size. The constructor contains Item(String name, String category, double price, String size). For the getters and setters, we have the price, name, category, and size. This class also contains a toString method where it will display the name and price of the item along with the size if it contains one.

In the Menu class, it contains the fields of the array list for all the item categories on the Item class. This contains appetizers, burgers, breakfast, entrees, sides, and drinks. For each item category, there is a getter and a setter for the size of the item. The getter returns a specific

appetizer, another getter gets the appetizer size, and the final getter returns all the appetizers as a string. This occurs for every item category in the menu.

In the Order class, it contains a getter to get the Item from the user that the user wants and then adds the item to the order. There are also ways to remove the item from the order, to clear the whole order, to get the order number, and to increment the order number depending on how many orders they get. The order will also display what server will get them their food.

In the Register class, it displays the subtotal, the tax of the food, and the total cost of the order depending on the items the customer has wanted. The register class contains getters to actually get the price and then methods to do the calculations for the order it has to calculate. The class will also be able to check the payment to see if the customer has paid the right amount, too much of the total so they have to get change, and to see if they will have to pay more if needed. It also contains a toString class to be able to print the total receipt.

In the Main class named Waiter, it shows the basic of this whole application. It prints out a greeting for the restaurant, gets the menu, and then asks the customer if they would like to read the full menu or just a section of it. It will then check the answer to make sure the user input the correct way. If the customer requests the menu by section, the program will then ask exactly which section they will like to view. Either the appetizers, burgers, breakfast, entrees, sides, or the drinks. The program will then say if they would like to place an order, they will have to enter the item ID and this will be able to add the item to their order.

Item
Double: price String: name String: category String: size
Item(String: name, String: category, double: price, String: size) getPrice: double setPrice(double price): void getName: String setName(String name): void getCategory: string setCategory(string: category): void getSize: String setSize(String: size): void toString: string

Menu
<ul style="list-style-type: none"> <li>ArrayList&lt;Item&gt; appetizers</li> <li>ArrayList&lt;Item&gt; sandwichesANDburgers</li> <li>ArrayList&lt;Item&gt; breakfast</li> <li>ArrayList&lt;Item&gt; entrees</li> <li>ArrayList&lt;Item&gt; sides</li> <li>ArrayList&lt;Item&gt; drinks</li> </ul>
<pre> Menu() populateMenu(): void getAppetizer(int: index) getAppetizerSize(): int getAllAppetizers: string getBurgers(int: index) getBurgersSize(): int getAllBurgers: string getBreakfast(int: index) getBreakfastSize(): int getAllBreakfast: string getEntress(int: index) getEntreesrSize(): int getAllEntrees: string getSides(int: index) getSidesSize(): int getAllSides: string getDrinksr(int: index) getDrinksSize(): int getAllDrinks: string toString() </pre>

Order
<ul style="list-style-type: none"> <li>Int: orderNo</li> <li>ArrayList&lt;Item&gt; order</li> </ul>
<pre> Order() getItem(int: index) addItem(item: Item) removeItem(item: item) getSize() clearOrder() getNumber() incrementNo() getServer() toString() </pre>

Register
static double: TAX double: subtotal double: total NumberFormat: numForm
Register(order: Order) getSubtotal() getTax() getTotal() calcSubtotal() calcTax() calcTotal() clearPayments() checkPayment(double: payment) toString()

Today, many restaurants are getting on the trend of having their customers order online what they would like to have and are able to pick up their food without having to call to order. Online ordering has grown over 16% in the last year. Over calling, orders are more accurate than ever since the restaurant knows exactly what the customers are trying to order. There are minimal complaints with this online ordering, the usage has been increasing every month and both the customer and the restaurant don't have to deal with complicated ordering processes anymore. The only downside of online ordering would have to be the constant upkeep of the app. They have to make sure that everything is running smoothly so that customers are able to use the online ordering to its full potential.

The system will be used by the main class, Waiter, when the program is started. The user will be greeted with some messages saying "Welcome to our restaurant" and will then list all the available menu items a guest may choose from. The user will be able to either see the full menu or a section of the menu. If they put an invalid input, they will then be asked again to input the correct one. Once they choose full menu, they will see what the restaurant has to offer. If they choose section, they will then be asked to input which section of the menu they will want to see. They will be able to type out what they want, one at a time, and will add it to their bill. Finally they will say done when they are finished and their total will be shown. They put their credit/debit card number into the system and if approved they will be able to continue. They will be told how long the food will take and that they should plan their time accordingly.

## Bibliography

Tory. "What Are the Benefits of Online Ordering?" *QSR Magazine*, 7 Jan. 2019, [www.qsrmagazine.com/business-advice/what-are-benefits-online-ordering](http://www.qsrmagazine.com/business-advice/what-are-benefits-online-ordering).