Jonas Williams-Gilchrist, et. al

**Class Diagrams – Comments**

**Account and its Subclasses:**

The purpose of this diagram is to outline the relationships between the different actors involved in using our online API for both ordering meals and maintaining the store. The abstract class *Account* defines many of the general-purpose attributes and methods all users should have access to on the website. This includes basic account information, such as first and last name, email, phone number, account balance, and a password. Additionally, all accounts should be able to update that information if it becomes out of date, so methods to edit names, passwords, etc. are included as well (not all are listed in the UML diagram for clarity). This class gives a good baseline for the structure of our accounts and will be very useful to the database because all its children will be easily listed in the same column of our database.

Next is the abstract class *Employee*. This class doesn’t currently contain any attributes or methods because of our model’s overall simplicity. It could be expanded to include data fields such as salary, time employed, and so forth. However, because all of the specific types of employees will extend this class, it will be easy for someone, such as the manager, to determine the total number of people he is employing.

The class *Customer* directly subclasses *Account*. This is the most basic type of instantiable account in our system. It allows the owner of the account to place and cancel their order, check if they have an outstanding order and the status of that order, and so forth.

A *Manager* is the most specialized type of *Employee*. They are effectively an admin of the website and should be able to exert more influence over the database, such as editing the inventory of products stored in the database, such as the number of plain bagels. Furthermore, they have the ability to create (and remove) accounts from the system as well as to display a report of relevant metadata pertaining to the business.

Bagel *Chef* is another type of employee in our system. The purpose of the class is to allow the chef to efficiently fetch the next order they need to work on and move it along to the next step, as well as update the inventory with the items that they used when fulfilling the order.

Finally, *Cashier* is the *Employee* primarily tasked with handling the monetary transactions between the customer and the company; they have methods to accomplish this. Additionally, it is useful for them to have the ability to modify the inventory of products on hand in case the customer asks for something extra at checkout, for example a beverage.

**Inventory Item and its Subclasses:**

The main purpose of creating the *InventoryItem* class is to provide a framework for storing the company’s inventory in a database. Each menu item subclasses *InventoryItem* and has its own unique set of attributes. These include the price of an individual item, the amount in stock, the name of the item, a relevant description and allergy information, as well as the actual type of the item, as defined in the enumeration *InventoryItemType*. These items can be of four types, as given to us by the client’s menu: Bagel, Beverage, Shmear, and Sandwich.