**Patrick Ian E. Cura**

**2154687**

**Programming 3 – Assignment 1 – Task 2 – Threads**

**Description of the chosen scenario**

The chosen scenario is a Pizza Store. There is a Pizza Making Machine that accepts orders either a Mushroom Pizza or an Anchovy Pizza from customers. The Pizza Making Machine also processes requests to replenish the supplies/ingredients to make the pizzas. Each of the ingredients have a storage limit. So once the storage limit for a particular ingredient is met, that ingredient cannot be replenished until the ingredient’s current amount is less than the storage limit.

The ingredients are as follows:

|  |  |
| --- | --- |
| **Ingredient** | **Storage Limit** |
| Garlic | 6 |
| Olives | 8 |
| Mushrooms | 10 |
| Anchovies | 6 |

It should also be noted that a particular pizza has a recipe and requires certain number of ingredients in order to be made. The recipes are as follows:

|  |  |
| --- | --- |
| **Mushroom Pizza** | |
| **Ingredient** | **Number Required** |
| Garlic | 1 |
| Olives | 0 |
| Mushrooms | 4 |
| Anchovies | 0 |

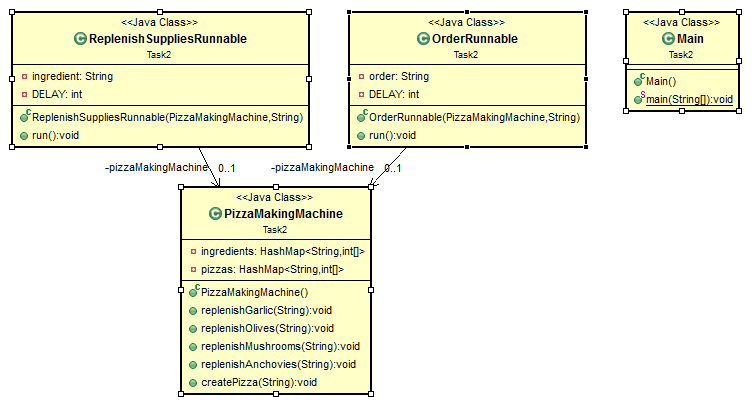
|  |  |
| --- | --- |
| **Anchovy Pizza** | |
| **Ingredient** | **Number Required** |
| Garlic | 2 |
| Olives | 2 |
| Mushrooms | 0 |
| Anchovies | 3 |

If the number of ingredients required to make a pizza is not met, the pizza cannot be made and the creation of other pizzas will have to wait.

The main program is named Main.java. It has two types of Runnable objects that are run: ReplenishSuppliesRunnable and OrderRunnable.

ReplenishSuppliesRunnable can take ingredient names and it will replenish that ingredients supplies in the Pizza Store.

**Name of the classes, methods, and objects that exists in the program**



**Explanation as to how multithreading functionality is achieved**

**Discussion on why you implemented Synchronization to your objects**

**Screenshot illustrating the execution of the program**