UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS
MATHEMATICAL AND COMPUTING SCIENCES UNIT

Computer Science 23: Object-Oriented Programming Paradigms
First Semester A.Y. 2023–2024

MACHINE PROBLEM
Restaurant Inventory System

Overview

Create an inventory system for a small restaurant. They offer a number of items, such as:

- Food
  - Appetizers
  - Main Course
  - Side Dishes / Snacks
  - Desserts
- Beverages
  - Water
  - Carbonated Drinks
  - Alcohol
- Merchandise
  - Store T-Shirts
  - Store Bags

Due to the amount of items they sell, they need to keep a constant list of their inventory, so they would know what items are low in stock and be able to purchase stock in bulk. You are to help them in creating a program to help them keep track of their inventory.

A menu item or food might require multiple ingredients. For example, in a japanese-themed restaurant, their food might consist of the following ingredients:

- Edamame → Salt, Olive Oil, Frozen Edamame
- Shoyu Ramen → Shoyu Base, Pork, Noodles, Oil
- Gyoza → Pork, Cabbage, Garlic, Salt, Soy Sauce

For a menu item/ingredient, the system should allow the user to view the description or characteristics of the item/ingredient stock.
Example:

- Salt → Category: Condiment; Brand: McCormick; Weight: 1kg; Color: White; Description: All purpose iodized salt
- Coke → Category: Beverage; Brand: Coca-Cola; Volume: 330ml; Type: Can; Description: Coke in can
- Smirnoff Mule → Category: Beverage; Brand: Smirnoff; Volume: 330ml; Type: Bottle
- Ground Pork → Category: Meat; Brand: Monterey; Weight: 20kg
- Octopus → Category: Meat; Weight: 10kg

Requirements

The system must have the following features **(50 points):**

1. Have a graphical user interface using JavaFX **(7 points)**
   a. **(4 points)** The GUI has an intuitive design and is easy to use.
   b. **(3 points)** The GUI is professional and presentable.
2. Have a list of inventory **(21 points)**
   a. **(3 points)** View the list of inventory
   b. **(4 points)** Add an item in the inventory with item descriptions
      - **(2 points)** Upon adding an item, the item's category must be specified by the user.
   c. **(4 points)** Delete an item in the inventory
   d. **(8 points)** Add an item usage in the inventory. This is to allow the user to see how much stock is remaining for a specific item.
      - *Example:*
        Initial entry for Ground Pork: 20kg
        Add usage entry for the Ground Pork: 8kg
        When the user views the remaining stock for Ground Pork, it should be: 12kg
3. Import functionality for bulk adding of inventory **(8 points)**
   a. File type for imports are in .csv format
4. SKU **(14 points)**
   a. **(6 points)** Each item that is added must have a unique identifier (SKU) that is automatically generated by your system. When the same item is used in adding another inventory record, it uses the same SKU for reference. The SKU is generated using the following information:
      - The first set of characters of the SKU contains the following:
        - The first three letters of the SKU should be the first three consonants of the category.
        - The next character of the SKU is a forward slash (/).
        - The next three letters of the SKU should be the first three consonants of the item.
        - If in any case that the word contains less than the required consonants, use the vowels instead for the remaining characters.
      - The next character of the SKU is a dash (-).
      - The last four characters of the SKU should be a randomly generated number from 0000 to 9999, unique to each item.
      - *Example:*
        Condiment: Salt → CND/SLT-0918
        Beverage: Coke → BVR/COK-2022
      - If the user failed to add the slash and dash, the system should automatically append them accordingly.

   b. **(8 points)** In adding an item (2.b.), the user would then have an option to just input the SKU and the system will automatically add the existing information in the system
      - The only empty fields that the user would need to fill out manually would be the quantity/amount/count to be restocked, and other fields that you deem necessary to be filled out manually.

As this project would require a form of database storage, it is up to you on how you will implement your storage system. You may use csv formatted files, json formatted files, or if you are familiar with sql, you may use that as well.

The code base for the system must be stored in a git project. Code changes, code contributions per student are expected to be shown via git commits and commit messages.

Bonus Features

These are not required in the project to have a perfect score, but it will allow additional points for your project output

1. Export functionality for current inventory stock as .csv
2. Add and display an image to the item (feature will not be avail able if import bulk items will be used) Groupings

You are to group yourselves to a minimum of 3 members, maximum of 4 members. You may add a member from a different lab class (e.g. 2 members from LAB1A, 1 member from LAB1B).

If there happens to be an existing group with 4 members out of all the groups in the class, bonus feature 1 will be a required feature.

You may add yourself in a group in Canvas. The groupings section should be visible for you under the People section.

Grading Rubrics

The grade for the machine problem will be computed as follows:

$$S + D + P = 100 \text{ points}$$

S: Source Code and Required Functionalities (50 points)

The functionalities specified in the Requirements section must be completed. The lecturer must be added as a "Developer" member on GitLab/GitHub.

D: Documentation (25 points)

Provide a user manual that explains how to use the application from a user's point of view. The manual should explain how to use the system with respect to each of the features.

P: Presentation (25 points)

For the Machine Problem presentation, the team will be presenting their output to their lecturer.

a) Source code

The compilation and running of the program must be seen on the recording.

b) The program

Demonstrate how the system implements each of the functionalities.

Deadline and Presentation for the project is on January 13, 2024.

THERE WILL BE NO EXTENSIONS FOR THE DEADLINE