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. 2022–2023

MACHINE PROBLEM
Inventory Management System

Overview

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

- Appetizers
- Main Course
- Side Dishes / Snacks
- Desserts
- Drinks / Beverages / 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 might require multiple ingredients. For example, in a japanese-themed restaurant, their food might consist of the following ingredients:

- Appetizers
    - Edamame → Salt, Olive Oil, Frozen Edamame
- Main Course
    - Shoyu Ramen → Shoyu Base, Pork, Noodles, Oil
- Side Dishes
    - 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:

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/ingredient in the inventory with item descriptions
        i. (2 points) Upon adding an item, the item's category must be specified by the user.
    c. (4 points) Delete an item/ingredient in the inventory
    d. (8 points) Add an item/ingredient usage in the inventory. This is to allow the user to see how much stock is remaining for a specific item/ingredient.
    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:
        i. The first two letters of the SKU should be the first and last characters of the category, respectively.
        ii. The next two letters of the SKU should be the first and last characters of the item, respectively.

iii. The fifth character of the SKU is a dash (-).
iv. The last four characters of the SKU should be a randomly generated number from 0000 to 9999, unique to each item.

Example:
Condiment: Salt → CTST-0918
Beverage: Coke → BECE-2022

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
   i. 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 must not be available if import bulk items will be used)

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.

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, create a screen recording that shows the following:

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.

The video presentation must be accompanied by either captions or voice over that explains what you are trying to show.

Deadline for the project is on January 11, 2023.