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. 2024–2025

MACHINE PROBLEM
Restaurant Point-Of-Sale System

I. Overview

You are tasked to develop a point-of-sale (POS) system for a restaurant. This system will serve as their menu entry, ordering, and billing system. The system's interface will be based on command-line inputs, hence, no GUIs would need to be developed. As an example for this document, we will be developing the system for a small coffee shop, named *Hell Week Coffee*.

Hell Week Coffee offers a range of items, such as:

- Drinks
  - Espresso Drinks
  - Blended Drinks
  - Tea
  - Others
- Food
  - Pastries
  - Cakes
  - Sandwiches
  - Pastas
  - Others
- Merchandise
  - T-Shirts
  - Bags
  - Mugs
  - Others

To be able to ring-in a customer's order, the item itself should be available (encoded) in the system. A user can log on as *Manager* in the system to be able to add, edit, or delete currently encoded items in the system. Then, users logged on as *Cashier* would be able select the item. When the cashier completes the transaction, a "receipt" containing the purchased items would be printed on the screen.

II.Interface

The system would entirely be a command-line-based interface. For menu selection, each option should have a number assigned to them. Hence, selecting an option would just require the user to type the associated character.

A. Sample Menu - Main Menu

```
Select user:
(1) Cashier
(2) Manager
(3) Quit

2

Welcome, Cashier!
Select option:
(1) Place order
(2) Exit
```

III.Data Definitions

Data	Description/Definition	Example

Item	• Any standalone (with or without customizations) product that a customer can order is defined as an item.
	• All items have the following characteristics:
	  • Item Code
	  • Name
	  • Sizes and their corresponding prices
	  • Category
	• An item can only be one of three types (item type):
	  • Food
	  • Drink
	  • Merchandise
	• An Item Code is an automatically generated identifier to be able to easily search an item. Its format is the following: *XX-YYYY-ZZZ*
	  • XX - The first and last characters of the category name
	  • YYYY - The first four characters of the name
	  • ZZZ - An auto-incrementing number, based on the number of items already in the category.
		• Latte
		• Americano
		• Croissant
		• Bagel
		• Hell Week Coffee Halloween 2024 Mug

Category	• A categorization of an item based on its item type.	• Item type: Drink Category: Espresso Drinks Item name: Americano • Item type: Food Category: Pastas Item name: Spaghetti Bolognese • Item type: Merchandise Category: Bags Item name: Hell Week Coffee Canvas Tote
Customization	• A selection of customizations that an item can have. • Only food and drinks can have customizations. • When customers place an order, adding customizations are not required.	• Item type: Drink Customization: Milk • Item type: Drink Customization: Sauce Add-On • Item type: Food Customization: Rice

Data	Description/Definition	Example
Customization Options	• The options and its corresponding price for a customization. • We will limit each customization to have a maximum of five (5) options only.	• Customization: Milk Options and Prices: Soy Milk - 35 Oat Milk - 40 • Customization: Sauce Add-On Options and Prices: Caramel - 30 Mocha - 30 • Customization: Rice Options and Prices: Garlic Rice - 20 Yang Chow Rice - 30

IV. Features

User - Feature	Pts	Description
Manager - Encode Items	25	The manager should be able to add an item in the system. The manager should be able to define the following: **(5 pts)** Basic item definitions • Item type • Item name • Sizes and prices • Category **(5 pts)** Customizations of an item **(5 pts)** Customization options of an item **(5 pts)** Auto-generated item code **(5 pts)** The user experience of encoding an item should be relatively easy to use
Manager - Modify Encoded Items	10	The manager should be able to modify an existing item in the system. To modify an item, an item code is required. The following actions can be done: **(5 pts)** Modify size and price definitions **(5 pts)** Modify customizations of an item
Manager - Delete Items	5	The manager should be able to delete an item in the system. To delete an item, an item code is required.
Manager - Encode from a File	10	A manger should be able to encode an item by importing a file. Implementation and format of the file is up to your team's discretion.
Cashier - Create a Transaction	25	The cashier should be able to add an item via the following options flow: **(20 pts)** *Create new transaction* > *Select an item type* > *Select a category* > *Select an item* > *Select customizations (if any)* > *Back to item type selection* **(5 pts)** The user experience of the entire transaction flow should be relatively easy to use
Cashier - Complete a Transaction	10	During item type selection, the cashier should have an option to complete the transaction. A bill representation (similar to a receipt) should be displayed, marking the completion of a customer's transaction. **(2 pts)** The list of ordered items are displayed **(2 pts)** The customizations per item are displayed **(2 pts)** The price per item is properly displayed, factoring in the price per customization, and quantity **(2 pts)** If there is more than one of the same item (with the same customizations), the quantity and price should adjust accordingly. **(2 pts)** The total transaction price is displayed
Error handling	10	Invalid inputs and error scenarios should be properly handled by the system.

User - Feature	Pts	Description
Option Selection, Exit Selection	5	The menu options selection and exit option selection should be properly implemented.
Total points	*100*	

V. Database Storage

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 relational databases (e.g. SQL databases), you may use that as well.

VI. Assumptions and Limitations

A. All items encoded in the system are assumed to always be in stock.
B. If a user makes a mistake while placing an order, the only option should be to cancel the order, and start again.
C. The “exit” option varies in behavior, depending on the context of the menu:

Context	Option Text	Behavior
User Selection	Quit	Quit application
Manager Main Menu; Cashier Main Menu	Logout	Return to user selection
Transaction Item Type Selection	Cancel Transaction	Return to cashier main menu

D. If an item has only one size, you can define the size an its corresponding price as: NS (No Size) and its price.

VII.Grading Rubrics

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

$$S + C + D + P = 100\,\%$$

S: Source Code and Required Functionalities (60%)

The functionalities specified in the Requirements section must be completed. The entire project should be on GitHub. Commit history will be reviewed. The lecturer must be added as a "Developer" member on GitLab/GitHub.

C: Code Cleanliness / Coding Practices (20%)

Good coding practices should be observed for the entire source code of your resulting project.

D: Documentation (10%)

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 (10%)

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 December 7, 2024.

THERE WILL BE NO EXTENSIONS FOR THE DEADLINE

REVISION AND UPDATE HISTORY

DATE	DESCRIPTION
10/16/2024	Added initial MP requirements.