Food Product Storage Systems

Group 14

Parth Popli
B00930712

Kanav Tyagi
B00921752

Afif Katrib
B00922055

Outline and Problem Statement

This project proposes building an effective food preservation and expiry management system in response to the demands facing modern food supply chains and stores. This system is designed to enhance food safety, minimize food waste, and simplifying inventory management. This is affectively achieved through utilizing an effective data storage system with multiple classes linked to each other.

Objectives of the Proposed Solution

Keeping the problem statement in mind, the following objectives are proposed:

- Class based Design: Creating a Class based architecture in C++. The parent class will have various attributes keeping track of the storage type, and expiry of the stored food items.
- Hierarchical Structuring: Utilize inheritance to establish hierarchical relationships between class. This will facilitate the categorization of food by the storage type they require.
- Facilitate Reusable Components: Emphasizing on class composition to facilitate the integration of reusable class date for the system architecture.

Architecture

As explained above, the system architecture uses the integration of class composition and inheritance within the C++ programming. The architecture of the proposed design is described in detail below:

- User Interface: A menu-based interface will be created for the user to input, retrieve, modify and view data as specified by the project requirements.
- Base Class “Food’: introducing a base class Food. It will contain common attributes like Item ID, Quantity, Price etc. Will also use the composition to establish the various dates associated with food items and inheritance to distinguish food items based on the storage type they require.
- Class Composition: Using class composition to represent the information related to the expiration Date, Date the food item is shelved and the Date of expiration. This will be done using the class Date as an object member in the parent composite class Food.
- Hierarchical Structuring: Leveraging the inheritance features offered by C++ to establish class pertaining to the storage conditions of various food i.e. Frozen, Refrigerated, Canned, and Dry Storage.
- Inheritance: The Classes Dry storage, Canned, Frozen, and Refrigerated are derived from the parent base class Food. As a result, each of these classes also have the attributes that the base class Food has.

UML diagram

A Unified Modeling Language (UML) diagram showing how various classes are related to each other is presented in Figure 1:

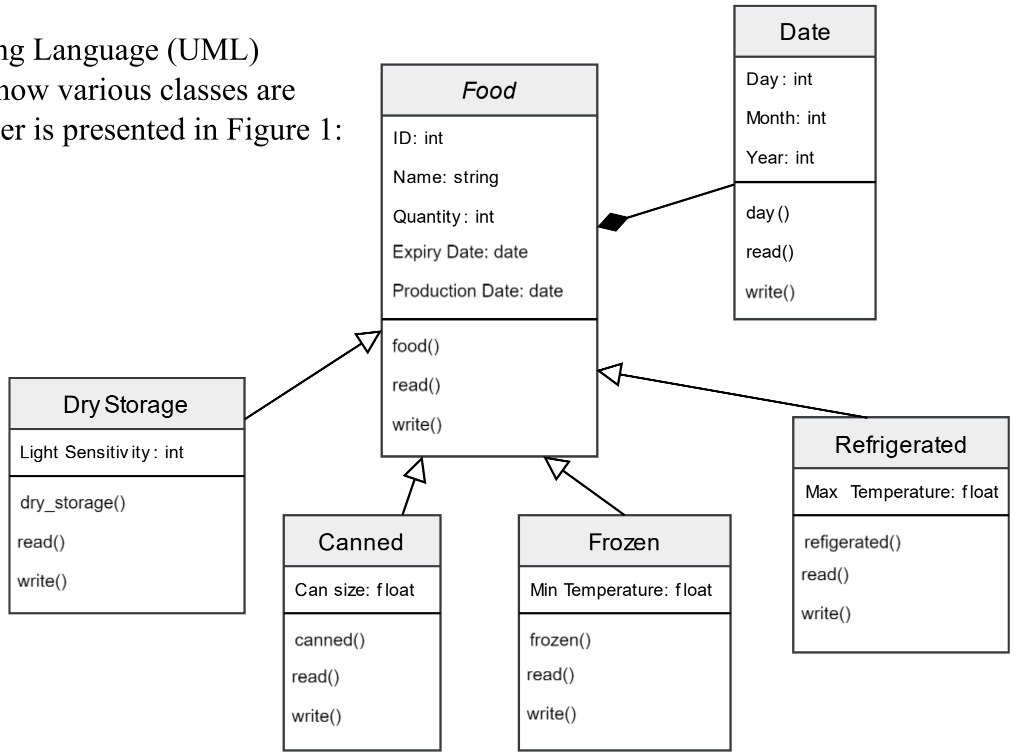

Figure 1: UML map

Attributes Of Classes

The attributes of each class are presented in the form of a table in Table 1 below:

Class Name	Attributes
Food	[ID, Name, Quantity, Expiry Date, Production Date]
Dry Storage	[Light Sensitivity]
Canned	[Can Size]
Frozen	[Minimum Temperature]
Refrigerated	[Maximum Temperature]
Date	[Day, Month, Year]

Table 1: Class Attributes

Conclusion

To conclude, the Food Preservation and Expiry Management System is a carefully thought-out architecture. It integrates inheritance, class composition, and object-oriented design concepts of C++ programming language.

The solution utilizes the inheritance hierarchies, anchored by the parent class "Food", which serves as a unified representation of food items. The software design also accommodates various preservation methodologies through specialized classes.

Strategically using the class composition, and the use of class Date makes the code more reusable and streamlined.