|  |  |  |
| --- | --- | --- |
| **LAB 211 Assignment** | **Type:** | **Long Assignment** |
| **Code:** | **J1.L.P0018** |
| **LOC:** | **500** |
| **Slot(s):** | **N/A** |

**Title**

Store Management at Convenience Store

**Background**

A convenience store sells 2 groups of products: products for daily use (bread, dumplings, sticky rice, ...) and products with long shelf life such as cookies, soft drinks, filtered water, .... The product.dat file is used to store information of all products in the Store. The Warehouse.dat file has stored import/export information. **Each import/export receipt can contain many different items with different quantities.**

**Program Specifications**

Build a management program. With the following basic functions

1. Build your data structure
2. Manage products
   1. Add a product
   2. Update product information.
   3. Delete product.
   4. Show all product.
3. Manage Warehouse
   1. Create an import receipt.
   2. Create an export receipt.
4. Report
   1. Products that have expired
   2. The products that the store is selling.
   3. Products that are running out of stock (sorted in ascending order).
   4. Import/export receipt of a product.
5. Store data to files
6. Exit.

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

**Features:**

***This system contains the following functions:***

Display a menu and ask users to select an option.

1. **Build your data structure (80 LOCs)**

* Product information: product code, product name, manufacturing date, expiration date, and other attributes.
* Warehouse Information: the import/export code is a self-incrementing 7-digit number, the time to create the import/export slip is taken from the system time. Each input/output code can contain many different items with different quantities.
* Classes, abstract classes, Interfaces.
* Must implement the polymorphism properties of object-oriented programming.

1. **Manage products/items of the store:**
   1. **Add a product. (30 LOCs)**

* Create a submenu.
* Remember that the constraints must be checked: product code can not duplicate
* Add the new product to collection.
* Ask to continuous create new product or go back to the main menu.
  1. **Update product information. (50 LOCs)**
* Require enter the product code
* If product code does not exist, the notification “Product does not exist”. Otherwise, we can start input new information of product and update.
* If new information is blank, then not change old information.
* Remember new information must be validated.
* System must print out the result of the updating.
* After updating, the program returns to the main screen.

**When changing product information, the corresponding information must be adjusted on the import/export receipts.**

* 1. **Delete product. (20 LOCs)**
* Require enter the product code
* Before the delete system must show confirm message.
* Show the result of the delete: success or fail.
* After delete, the program returns to the main screen.

**Note, only remove the product from the store's list when the import / export information for this product has not been generated.**

* 1. **Show all product. (20 LOCs).**
* Show all data in the product.dat or product’s collection file into the screen.

1. **Warehouse management** 
   1. **Create an import receipt. (50 LOCs)**

* Create a submenu.
* Remember that the constraints must be checked: productCode
* Add the new product to the import receipt.
* Ask to continuous add new product or show confirm message.
* After confirm, the result of the import receipt: success or fail.
* After confirm, the program returns to the main screen.
  1. **Create an export receipt. (100 LOCs)**
* Create a submenu.
* Remember that the constraints must be checked: productCode, quantity within expiry date
* Add the new product to the export receipt.
* Ask to continuous add new product or show confirm message.
* After confirm, the result of the export receipt: success or fail.
* After confirm, the program returns to the main screen.

1. **Report** 
   1. **Products that have expired. (25 LOCs)**

* The system searches the product list, and returns all product that has manufacturing date greater than expiration date.
* Show result list: product code, product name, production date, expiration date, quantity
* The program returns to the main screen.
  1. **The products that the store is selling. (25 LOCs)**
* The system searches the product list, and returns all product that has manufacturing date less than or equal than expiration date.
* Show result list: product code, product name, production date, expiration date, quantity
* The program returns to the main screen.
  1. **Products that are running out of stock (sorted in ascending order). (25 LOCs)**
* The system searches the product list, and returns all product that has manufacturing date less than or equal than expiration date.
* Show result list: product code, product name, production date, expiration date, quantity
* Sorted in ascending order byquantity
* The program returns to the main screen.
  1. **Import/export receipt of a product. (25 LOCs)**
* Require enter the product code
* If product code does not exist, the notification “Product does not exist”.
* Show product’s data in the warehouse.dat file or warehouse’s collection into the screen.
* The program returns to the main screen.

1. **Store data to files. (50 LOCs)**

* Store list product to product.dat file.
* Store list warehouse information to warehouse.dat file.
* The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
* The lecturer will explain the requirement only once on the first slot of the assignment.