**UNIVERSITY OF CAPE COAST**

****

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**COURSE TITLE:**

**DATA STRUCTURES**

**COURSE CODE:**

**CSC 319**

**ASSIGNMENT**

**REPORT ON INVENTORY MANAGEMENT**

**INTRODUCTION**

Inventory management systems are used by businesses to manage their inventory levels, sales, and orders. These systems need to be efficient and effective to ensure that the business runs smoothly. One way to achieve this is by implementing data structures such as stacks and queues. In this report, we will discuss how these data structures can be implemented in an inventory management system.

**Stacks in Inventory Management**

A stack is a last-in, first-out (LIFO) data structure. In our inventory management system, we used stack to track the movement of items such as **beverages**, **bread/bakery**, **canned/jarred** **goods** and **diary** in and out of our warehouse. When an item is received, it can be added to the top of the stack. When an item is purchase, it can be removed from the top of the stack. This ensures that the most recently received or purchased, items are always at the top of the stack, and the oldest items are at the bottom.

Also, we used stack to keep track of the data structure (queue, stack, etc.) that will be implemented in moving items.

**Queues in Inventory Management**

A queue is a first-in, first-out (FIFO) data structure. In our inventory management system, we used queues to keep track of the movement of items such as **Dry/Baking Goods**, **Frozen Foods** and **Meat** in and out of our warehouse. When a new order is received, it can be added to the back of the queue. When an order is fulfilled, it can be removed from the front of the queue. This ensures that the oldest orders are always at the front of the queue, and the newest orders are at the back.

Also, we implemented queues with regards to *Issue Goods* and *Viewing Goods and Bills*

**ArrayLists in Inventory Management System**

ArrayLists are used to store a fixed number of items in a specific order. In our inventory management system, we used ArrayLists to keep track of the movement of items such as **Produce**, **Cleaners**, **Paper Goods** and **Personal Care** in and out of our warehouse. For example, you can use an ArrayLists to store the list of products in the inventory system. Each element of the array can represent a particular product and its associated attributes, such as the product name, price, and quantity.

**HashMaps in Inventory Management System**

HashMaps are useful for storing and accessing data quickly based on a unique key. In our Inventory Management System, we used HashMaps to store information about vendors where the vendor ID becomes the key of the HashMap and the information about the vendor becomes its corresponding value. Additionally, we used HashMaps to keep track of product sales with goods ID as key and quantity sold as its corresponding value.

**Report On the Performance of Algorithm**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Method** | **Big O Notation** |
| Popping item from stack | (DataAccess.java)allGoodsInfo  (AddGoodsController.java)save | O(n) |
| Polling item from queue | (DataAccess.java)allGoodsInfo  (AddGoodsController.java)save  (IssuedGoodsController.java)process | O(n) |
| Removing item from list | (DataAccess.java)allGoodsInfo  (AddGoodsController.java)save | O(n2) |
| Getting item from database | (DataAccess.java)allGoodsInfo | O(n) |
| Traversing through vendor info using HashMaps | (DataAccess.java)allVendorInfo | O(n) |
| Keeping track of product sales | (AddGoodsController.java)proceed | O(n) |

**GROUP MEMBERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **INDEX NUMBERS** | **Activities** | **%Contribution** |
| Jessica Dziedzorm Gogovi | ET/CSC/20/0018 | Take documentation on our meeting also assist in the database creation. | 9.7 |
| Nana Adjoa Bentum | ET/CSC/20/0004 | Take documentation on our meeting also assist in the database creation. | 9.7 |
| Emmanuel Adjei | ET/CSC/20/0007 | Coordinate group meetings and assist in database creation and normalization. | 9.8 |
| Edmund Ahiekpor | ET/CSC/20/0011 | Research on data structures and their implementation. | 9.8 |
| Christian Owusu Yaw | ET/CSC/20/0010 | Code analyses | 9.8 |
| Stephen Commodore | ET/CSC/20/0022 | Helped in the UI development. | 9.7 |
| Alhassan, Mikdad | ET/CSC/20/0032 | UI development and analyse codes. | 9.9 |
| Jomphia Essuman Asante | ET/CSC/20/0029 | Research on data structures and their implementation. | 9.9 |
| John Berewono Kyenaatuo | ET/CSC/20/0031 | UI development and analyse codes. | 9.9 |
| Solomon Eshun | ET/CSC/20/0027 | Wrote the logic, data structures implementation and database connection for the project. | 11.8 |

**VIEW REPORTS ON THE ON THE INVENTORY MANAGEMET**

The interface consists of a vertical menu on the left side of the application, which contains eight (8) buttons: "Profile," "Add Goods," "View Goods," "Issue Goods," "View Vendors," "View Bills," "View Issued Goods," and "Register Vendor."

The field part of the interface has been divided into three sections: "Add New Goods to Inventory," "Add New Category to Inventory," and "Add Goods Item."

In the "Add Goods Item" section, there are several fields that allow you to input information about a new item to be added to your inventory, including "Quantity," "Selling Price," "Profile," "Buying Price," "Category Type," "Add Goods" button, "Remove Goods" button, and "Top Up Good Quantity" button. In the "Add New Goods to Inventory" section, you can enter the name of a new item and select a category from a drop-down list, and then click the "Add New Good to Inventory" button to add it to your inventory. In the "Add New Category to Inventory" section, you can enter the name of a new category and click the "Add New Category to Inventory" button to add it to your inventory.

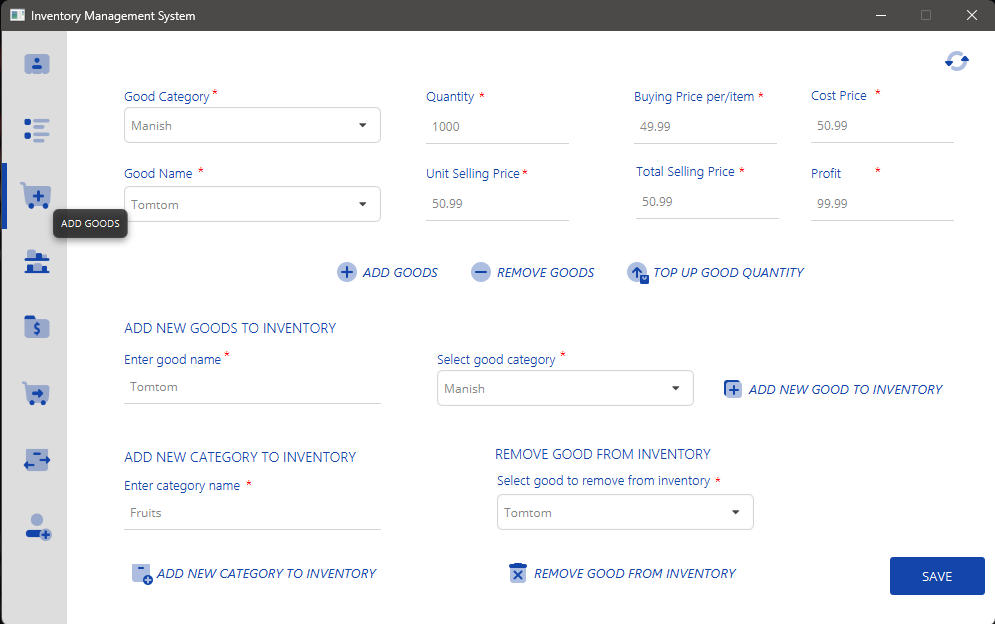
The user interface includes a comprehensive goods categorization system, encompassing a wide range of categories such as beverages, bread/bakery, cleaners, personal care and more.

Each category is accompanied by its respective prices. The user interface is equipped with a dropdown menu that is linked to each category, providing a distinct list of items in the corresponding dropdown for easy selection of goods by their name.

The administrator is empowered to create a new category in the event it is not present among the existing categories. Additionally, the administrator can add items to the category that are not yet defined in the category.

The user interface enables the administrator to input the quantity of goods, as well as their corresponding selling and buying price, facilitating the accurate computation of total selling price, total cost price, and profits. Additionally, the interface displays this computed value, providing a clear and concise overview of the financial implications of inventory management.

Finally, there are two buttons at the bottom of the interface: a "Refresh" button to update the data on the screen and a "Save" button to save any changes you have made to the inventory. The "Save" button allows the administrator to store newly created categories or goods into the database. The "Refresh" button enables the display of uploaded data after the addition of new categories or goods. Additionally, the presence of a red asterisk on the interface serves as an indicator that certain fields must be completed or selected before proceeding.



**CONCLUSION**

In conclusion, data structures such as stacks and queues can be implemented in an inventory management system to manage data efficiently and effectively. The implementation of stacks and queues can be done using arrays or linked lists, depending on the requirements of the system. By using these data structures, businesses can ensure that their inventory levels, sales, and orders are managed efficiently, which in turn can help them run their operations more smoothly.