GROUP 15 MEMBERS

1. PS/CSC/20/0114

2. PS/CSC/20/0128

3.PS/CSC/20/0141

4.PS/CSC/20/0107

5.PS/CSC/20/0039

6.PS/CSC/20/0042

7.PS/CSC/20/0148

8. PS/CSC/20/0144

9. PS/CSC/20/0028

10. PS/CSC/20/0168

11. PS/CSC/19/0084

INTRODUCTION:

To keep track of their inventory, vendors, bills, and supplied goods, provision store owners need to use an inventory management system. This document describes an inventory control system for a provision shop that can carry out several tasks, including adding items to inventory, viewing suppliers, items in inventory, viewing bills, issuing items to customers, and viewing items that have been issued.

FUNCTIONALITY:

Using information such the product name, description, price, quantity, and vendor information, the inventory management system enables the store owner to add new

items to the inventory. To individually identify each product, a unique ID is given to it. The system also gives the store owner access to a list of suppliers who provide goods to the shop, along with contact information for each.

The inventory list and each item's details, including name, description, price, quantity, and vendor information, are visible to the store owner. The list can be sorted according to several factors, including the product name, price, and quantity. To help the business owner replenish the inventory in a timely manner, the system also provides alerts when the stock of a specific product goes below a predetermined threshold.

When a consumer makes a purchase, the store owner has the option of issuing the products to the buyer and creating a bill. The system can update the inventory database and subtract the quantity of the product from the inventory. The store owner can see a list of issued items as well as information about them, including the product. The list of provided items and information about them, including the product name, quantity, customer information, and bill information, are also available to the business owner.

Each bill is assigned a special ID by the system, and the business owner may view the bill's specifics, including the product name, quantity, price, and total amount. The owner of the store can also see a list of the items that have been given to customers, together with information about them such as the product name, quantity, customer information, and bill information.

ITEM CATEGORY

To effectively manage a provision store, it is important to have a well-organized inventory management system. This involves categorizing the items in the store and implementing appropriate data structures for adding, removing, and tracking them. Here are some of the categories that are useful in the store:

Beverages – coffee/tea, juice, soda, energy drinks, water, sports drinks, etc.

Bread/Bakery – sandwich loaves, dinner rolls, tortillas, bagels, croissants, etc.

Canned/Jarred Goods – vegetables, spaghetti sauce, ketchup, pickles, soups, etc.

Dairy – cheeses, eggs, milk, yoghurt, butter, sour cream, cream cheese, etc.

Dry/Baking Goods – cereals, flour, sugar, pasta, mixes, spices, baking powder, etc.

Frozen Foods – waffles, vegetables, individual meals, ice cream, frozen pizza, etc.

Meat – lunch meat, poultry, beef, pork, sausages, bacon, etc.

Produce – fruits, vegetables, herbs, mushrooms, etc.

Cleaners – all-purpose, laundry detergent, dishwashing liquid/detergent, glass cleaner, etc.

Paper Goods – paper towels, toilet paper, aluminium foil, sandwich bags, trash bags, etc.

Personal Care – shampoo, soap, hand soap, shaving cream, deodorant, toothpaste, etc.

DATA STRUCTURE IMPLEMENTATION:

To manage the inventory of the store, we suggest the following implementation of data structures:

1. Implement stacks when adding and removing items in categories 1 to 4 as they have a small number of items and LIFO (Last-In-First-Out) data structure works best for them.
2. Implement queues when adding and removing items in categories 5 to 7 as they have a large number of items and FIFO (First-In-First-Out) data structure works best for them.
3. Implement lists when adding and removing items in categories 8 to 11 as they are not time-critical and require random access. Lists provide constant time access to any element of the list.

4. For issued goods and viewing goods and bills, you can use iterators, recursion, stack and queue implementations of the list, or other techniques based on the specific requirements of the application.
5. Use maps to keep track of product sales, where the product code is the key and the number of sales is the value. This allows for easy retrieval of product sales data.
6. Use HashMap to store information about the vendors, where the vendor's name is the key and the vendor details (e.g., address, phone number, email) are the values. This allows for easy retrieval of vendor information by name.

OTHER REQUIREMENTS:

1. To maintain the balance between too high and too low stock, the store owner needs to regularly monitor inventory levels and adjust orders accordingly. This can be done by setting minimum and maximum stock levels for each item, and placing orders to replenish stock when it falls below the minimum level.

2. Report on the data structure implementation in the inventory management system for a provision store:

a) Stacks: Stacks are used when adding and removing items in categories 1 to 4. In this system, stacks are used to keep track of the inventory levels of items in the Beverages, Bread/Bakery, Canned/Jarred Goods, and Dairy categories. When a new item is added to the inventory, it is pushed onto the top of the stack. When an item is sold or removed from the inventory, it is popped off the top of the stack.

b) Queues: Queues are used when adding and removing items in categories 5 to 7. In this system, queues are used to keep track of the inventory levels of items in the Dry/Baking Goods, Frozen Foods, and Meat categories. When a new item is

added to the inventory, it is added to the back of the queue. When an item is sold or removed from the inventory, it is removed from the front of the queue.

c) Lists: Lists are used when adding and removing items in categories 8 to 11. In this system, lists are used to keep track of the inventory levels of items in the Produce, Cleaners, Paper Goods, and Personal Care categories. When a new item is added to the inventory, it is added to the end of the list. When an item is sold or removed from the inventory, it is removed from the list.
d) Maps: Maps are used to keep track of product sales. Each time a product is sold, its product code is entered into a sales file. In this system, a map is used to store the product code and the number of times it has been sold. This allows the store owner to generate reports on product sales and track which products are most popular.
e) HashMaps: HashMaps are used to store information about the vendors. In this system, a HashMap is used to store the name and contact information of each vendor. This allows the store owner to easily access vendor information when placing orders or making inquiries.

Overall, the use of different data structures in this inventory management system allows for efficient and organized management of inventory, sales, and vendor information. It enables the store owner to easily add and remove items from the inventory, track sales and product popularity, and maintain relationships with vendors.

3. Based on the provided code, here is the performance analysis of the algorithm using Big O Notation:

The constructor initializes the front, rear, capacity, and cat variables.

It connects to the database and creates a table if it doesn't exist.

It checks the size of the table and sets the capacity accordingly.

It also gets the top index and fills the queue with null values if it is empty.

Time Complexity: O(n), where n is the number of items in the table.

a) Enqueue: The enqueue method inserts an item into the queue by updating the row with the next available index.

If the queue is full, it expands the queue size by doubling it and inserts the item.

Time Complexity: O(1) for most cases. However, it can be O(n) in the worst case when the queue is full, and the size has to be doubled.

b) Dequeue: The dequeue method removes an item from the front of the queue by updating the row with null values and shifting the rows to fill the gap.

If the queue is empty, it displays an error message.

Time Complexity: O(n), where n is the number of items in the queue because it has to shift all the rows to fill the gap.

c) Expand Queue Size: The expandQueueSize method doubles the queue's size by inserting null values into the table.

Time Complexity: O(n), where n is the number of items that need to be inserted.

d) Shift Rows: The shift Rows method shifts all the rows in the table to fill the gap left by the removed item.

Time Complexity: O(n), where n is the number of rows that need to be shifted.

The overall time complexity of this algorithm is O(n) in the worst-case scenario when the queue is full, and the size has to be doubled or when dequeuing, and all the rows need to be shifted. In most cases, the time complexity is O(1) for enqueue and O(n) for dequeue. However, the algorithm's performance may vary depending on the number of items in the queue and the size of the table.

CONCLUSION:

Based on the given requirements, a provision store inventory management system has been developed. The system provides features for adding goods, viewing vendors, viewing bills, issued goods, and viewing issued goods. In addition, other functionality has been included, such as maintaining the balance between too high and too low stock, generating reports, implementing searching and sorting algorithms, and analyzing the algorithm's performance using Big O Notation. The data structure implementation has been done using stacks, queues, and lists, depending on the categories of items. Maps and HashMaps have been used to keep track of product sales and vendor information, respectively. The system's interface has been designed to be user-friendly and intuitive, enabling store owners to manage their inventory efficiently. Overall, the provision store inventory management system is an effective tool for store owners to manage their inventory and track sales. The system's various functionalities and data structure implementations enable efficient stock management and sales tracking. The system's analysis using Big O Notation has helped identify any performance bottlenecks and optimize the system's performance.