CSC212 Project

Data Structures

E-Commerce Inventory &

Order Management System

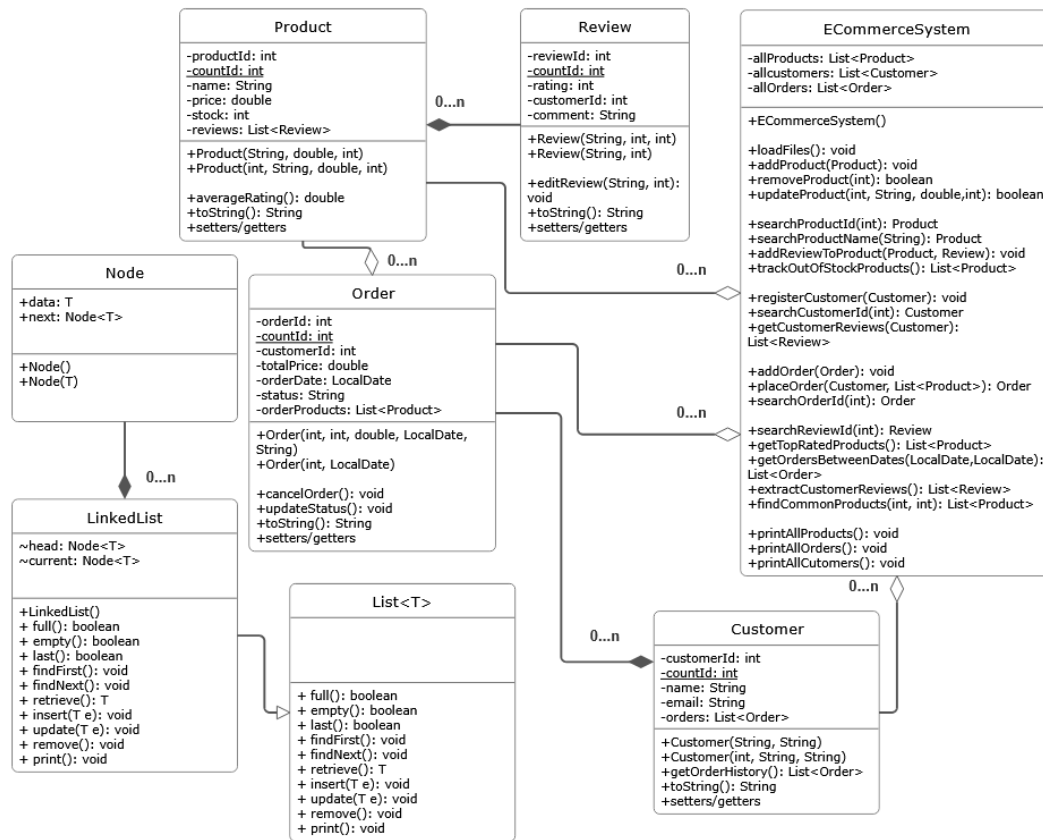## PHASE 1

Yazan Almuzaini 445103365

Abdulaziz Alnahedh 445102328

Sultan Almandeel 445100971

Class Diagram:

Product

-productId: int
-countId: int
-name: String
-price: double
-stock: int
-reviews: List<Review>

+Product(String, double, int)
+Product(int, String, double, int)

+averageRating(): double
+toString(): String
+setters/getters

Review

-reviewId: int
-countId: int
-rating: int
-customerId: int
-comment: String

+Review(String, int, int)
+Review(String, int)

+editReview(String, int): void
+toString(): String
+setters/getters

ECommerceSystem

-allProducts: List<Product>
-allcustomers: List<Customer>
-allOrders: List<Order>

+ECommerceSystem()

+loadFiles(): void
+addProduct(Product): void
+removeProduct(int): boolean
+updateProduct(int, String, double,int): boolean

+searchProductId(int): Product
+searchProductName(String): Product
+addReviewToProduct(Product, Review): void
+trackOutOfStockProducts(): List<Product>

+registerCustomer(Customer): void
+searchCustomerId(int): Customer
+getCustomerReviews(Customer):
List<Review>

+addOrder(Order): void
+placeOrder(Customer, List<Product>): Order
+searchOrderId(int): Order

+searchReviewId(int): Review
+getTopRatedProducts(): List<Product>
+getOrdersBetweenDates(LocalDate,LocalDate):
List<Order>
+extractCustomerReviews(): List<Review>
+findCommonProducts(int, int): List<Product>

+printAllProducts(): void
+printAllOrders(): void
+printAllCutomers(): void

Node

+data: T
+next: Node<T>

+Node()
+Node(T)

Order

-orderId: int
-countId: int
-customerId: int
-totalPrice: double
-orderDate: LocalDate
-status: String
-orderProducts: List<Product>

+Order(int, int, double, LocalDate,
String)
+Order(int, LocalDate)

+cancelOrder(): void
+updateStatus(): void
+toString(): String
+setters/getters

LinkedList

~head: Node<T>
~current: Node<T>

+LinkedList()
+ full(): boolean
+ empty(): boolean
+ last(): boolean
+ findFirst(): void
+ findNext(): void
+ retrieve(): T
+ insert(T e): void
+ update(T e): void
+ remove(): void
+ print(): void

List<T>

+ full(): boolean
+ empty(): boolean
+ last(): boolean
+ findFirst(): void
+ findNext(): void
+ retrieve(): T
+ insert(T e): void
+ update(T e): void
+ remove(): void
+ print(): void

Customer

-customerId: int
-countId: int
-name: String
-email: String
-orders: List<Order>

+Customer(String, String)
+Customer(int, String, String)
+getOrderHistory(): List<Order>
+toString(): String
+setters/getters

0...n

All Required Methods, with Big O analysis:

Only a single data structure type was needed, which is the linked list.

Method: addProduct

Description: adds a new product and appends its information to the CSV file.

- Time complexity: O(1)
- Space Complexity: O(1)

Method: removeProduct

Description: Removes a product by its ID by searching for it and setting its stock to zero if found.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: updateProduct

Description: Updates the details (name, price, and stock) of a product found by its ID.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: searchProductId

Description: Searches the product list for a product with the specified ID and returns it if found, otherwise, returns null.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: searchProductName

Description: Searches the product list for a product whose name matches the given name and returns it if found, otherwise, returns null.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: trackOutOfStockProducts

Description: Creates and returns a list of all products that currently have zero stock.

- Time complexity: O(n)
- Space Complexity: O(m)

Method: registerCustomer

Description: Registers a new customer by writing their details to the customers file and adding them to the customer list.

- Time complexity: O(1)
- Space Complexity: O(1)

Method: searchCustomerId

Description: Finds and returns the customer whose ID matches the given customer ID, returns null if no match is found.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: addOrder

Description: adds a new order by writing its details (including product IDs, total price, date, and status) to the orders file and adding it to the order list.

- Time complexity: O(n)
- Space Complexity: O(m)

Method: placeOrder

Description: Creates a new order for a customer by adding the selected products, updating their stock numbers, calculating the total price, saving the order, and returning it.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: searchOrderId

Description: Searches the order list for an order with the specified ID and returns it if found, otherwise, returns null.

- Time complexity: O(n)
- Space Complexity: O(1)

Method: searchReviewId

Description : Searches through all products and their reviews to find and return the review with the specified ID, returns null if not found..

- Time complexity: O(p*r) where p is the number of products and r is the number of reviews.
- Space Complexity: O(1)

Method: getTopRatedProducts

Description : Finds and returns the top three products with the highest average ratings by going through all products and comparing their ratings..

- Time complexity: O(p*r)
- Space Complexity: O(1)

Method: searchOrderId

Description: Retrieves and returns all orders whose dates are between the specified start and end dates.

- Time complexity: O(n)
- Space Complexity: O(m)

Method: getOrdersBetweenDates

Description: Retrieves and returns all orders whose dates are between the specified start and end dates.

- Time complexity: O(n)
- Space Complexity: O(m)

Method: getCustomerReviews

Description: retrieves and returns all reviews written by a given customer by going through every product and its reviews.

- Time complexity: O(p*r)
- Space Complexity: O(m)

Method: findCommonProducts

Description: Finds and returns a list of products that were reviewed by both given customers and have an average rating greater than 4.0.

- Time complexity: O(p*r)
- Space Complexity: O(m)