Assignment 3: Byte Me!

1 Important Instructions

1. This assignment is a take-home lab assignment. No extensions whatsoever will be provided. Any submission after the deadline will not be evaluated.
2. **Usage of ChatGPT, Claude, Gemini, and other LLMs** will be treated as **plagiarism**.
3. **Copying code, whether in full or partial**, from any other course student, or from online sources **without inclusion of due credit** to the source in your README file, will be treated as **plagiarism**.
4. If you see any ambiguity or inconsistency in a question, please seek clarification from the TAs by **commenting under the GC post** of the Project Deadline only. All doubts will be resolved in the comments, so keep an eye out on the comments for any clarifications the TAs/TF might offer.
5. Doubts on the day of submission will not be entertained. With doubts coming, some parts of the assignment may get updated. So, make sure you regularly follow the GC Post and the comments section.

2 Introduction

As a part of this assignment, you have to implement a CLI-based food ordering system called "Byte Me!", designed specifically for our college canteen. Your task is to develop a complete command-line interface system that will:

- Help students browse the canteen menu, place orders, and track their delivery without leaving their comfy hostel rooms.
- Enable canteen staff to manage menu items and process orders efficiently.
- Maintain order histories (so you can remember what you ate during that 3 AM coding session).
- Use collections to organise and sort data to make this process efficient!

3 Concepts

You'll use various **Java collections** to manage data, such as:

- Menu of Food Items
- List of current orders to be processed
- Order History for each individual customer

You are free to use any of the collections available to you, such as `ArrayList`, `TreeMap`, `TreeSet`, `PriorityQueue`. There is no "one" solution to the assignment, and you can be creative, maybe even make your own Iterator! As long as it fulfills the purpose, it's okay :D.

4 User Flow

4.1 Admin Interface

- **Menu Management**
  - **Add new items:** Admin should be able to add new food items to the menu, with details like price, category, and availability.
  - **Update existing items:** Admin can update the details of existing menu items, such as price or availability.
  - **Remove items:** Admin can remove food items that are no longer available or have been discontinued. When an item is removed, the status of all pending orders containing that item will be updated to 'denied'.

- **Order Management**
  - **View pending orders:** Admin can view the list of orders waiting to be processed. Orders should be handled in the order they were received, ensuring fair processing.
  - **Update order status:** Admin can mark orders as completed or update them at different stages (e.g., preparing, out for delivery).
  - **Process refunds:** Admin can process refunds for orders that were canceled or had issues.
  - **Handle special requests:** Customers can add a brief description for special requests (e.g., "extra spicy" or "no onions").
  - **Order Priority:** Orders placed by VIP customers should be given priority over regular orders, ensuring VIP orders are processed first when multiple orders are pending.
- **Report Generation**
  - **Daily sales report:** Admin can generate a report of all sales and orders processed during the day, which includes details like total sales, most popular items, and total orders.

4.2 Customer Interface

- **Customer Types**
  - **VIP:** Orders from VIP customers should be given priority in the order processing flow. You can become a VIP by paying a specified amount.
  - **Regular:** Regular customers' orders will be processed in the order they are received but will only be handled after all VIP orders have been processed.
- **Browse Menu**
  - **View all items:** Customers can view the complete menu of food items along with their prices and availability.
  - **Search functionality:** Customers can search for specific items by name or keyword.
  - **Filter by category:** Customers can filter menu items based on categories (e.g., snacks, beverages, meals).
  - **Sort by price:** Customers can sort menu items in ascending or descending order based on price to find affordable options quickly.
- **Cart Operations**
  - **Add items:** Customers can add multiple items to their cart, specifying the quantity for each.
  - **Modify quantities:** Customers can adjust the quantity of items in their cart before checking out.
  - **Remove items:** Customers can remove items from their cart if they change their mind.
  - **View total:** Customers can view the total price of all items in their cart before finalizing the order.
  - **Checkout process:** Customers can complete their order by providing payment and delivery details.
- **Order Tracking**
  - **View order status:** Customers can track the status of their orders in real time, from 'order received' to 'delivered', or 'denied'.
  - **Cancel order:** Customers can cancel their orders before they are prepared or processed.
  - **Order history:** Customers can view their past orders and re-order previous meals if desired.
- **Item Reviews**
  - **Provide review:** Customers can leave a review for specific food items they have ordered.
  - **View reviews:** Customers can view reviews of a particular item left by other customers.

5 Marking Scheme

Total: 40 marks

1. **Implementation of all features in a working manner:** 25 marks
2. **Correct usage and identification of collections for each feature:** 10 marks
3. **Clean coding and adherence to OOPS principles:** 5 marks