Management of Orders and Collections for Quick-Bites Restaurant

In the fast-paced world of the Quick-Bites Restaurant, there is a pressing need for an efficient system to handle order processing, cancellation, and daily collection tracking. The existing manual processes have proven to be error-prone, time-consuming, and lacking in transparency. The absence of a structured data-driven approach has led to challenges in maintaining accurate records, managing order cancellations, and tracking daily collections. As a result, there is a demand for a robust software solution that leverages file-based data storage and utilizes core Java concepts to streamline the management of orders and collections within the restaurant.

Current Challenges:

1. Order Processing: The restaurant faces difficulties in efficiently processing customer orders, often leading to mistakes in order details and billing.

2. Order Cancellation: The manual cancellation process is cumbersome and prone to errors, leading to misunderstandings and inconvenience for both customers and the restaurant.

3. Collection Tracking:The inability to accurately track daily collections makes it challenging to assess the restaurant's performance and revenue on a day-to-day basis.

4. Data Management:The current manual record-keeping system lacks transparency, version control, and effective data retrieval mechanisms.

Proposed Solution:

To address these challenges, we propose the development of a Java-based software solution for order management and collection tracking for Quick-Bites Restaurant. This solution will leverage core Java concepts and file-based data storage to create a user-friendly system that offers the following functionalities:

1. Order Placement:Customers can place orders, specifying menu items and quantities, through an interactive interface.

2. Order Cancellation:The system will provide a straightforward process for canceling orders, ensuring accuracy and customer satisfaction.

3. Collection Tracking:The system will track and report daily collections, providing valuable insights into the restaurant's performance.

4. Data Storage: Utilizing file-based data storage, the system will maintain a secure and organized repository of order details, allowing for easy access and retrieval.

1. Order Processing:Customers can place orders, and the system will calculate the total bill amount based on the selected items and quantities.

2. \*\*Order Cancellation:\*\* The system will allow for the cancellation of orders, updating the order status and reflecting the change in the collection report.

3. \*\*Daily Collection Report:\*\* The system will generate a daily collection report, showing the total revenue earned by the restaurant.

4. \*\*Data Persistence:\*\* Order details, including order IDs, dates, bill amounts, menu items, and order statuses, will be stored in CSV files, ensuring data persistence and ease of management.

Features of the Proposed System:

1. RestaurantApp (Main Class):

   - This class serves as the entry point for the restaurant management application.

   - It contains the main method that interacts with the user and provides options for order management and collection tracking.

   - Uses instances of other classes to handle different functionalities.

   - Provides a loop for the user to choose actions and facilitates the flow of the application.

2. GetFileData(Helper Class for File Data Retrieval):

   - This class is responsible for retrieving data from CSV files.

   - It has methods to display complete data, search for specific data by ID, get the last ID, and get the complete file data as a 2D array.

   - This class is used to manage data related to menu items and order details.

   - It helps in retrieving essential information for order processing and reporting.

3. MenuItem (Representation of Menu Items):

   - This class represents a menu item with attributes such as MenuID, name, and price.

   - It may also include other attributes like category, description, etc., if needed in the future.

   - The `GetFileData` class uses this class to manage and retrieve menu item details from the menu list CSV file.

4. Order (Representation of Customer Orders):

   - This class represents a customer's order, containing attributes such as OrderID, menu items, quantities, total bill amount, date, and status.

   - It is used to manage order data and facilitate order placement, cancellation, and tracking.

   - The class interacts with the `GetFileData` class to save and retrieve order details from the order details CSV file.

5. CollectionReport (Representation of Daily Collection Report):

   - This class represents a daily collection report, containing attributes such as the date and total collection amount for that day.

   - It is responsible for generating the daily collection report and displaying the total revenue earned by the restaurant for a specific date.

   - The class interacts with the `GetFileData` class to retrieve order details and calculate the total collection for a given date.

Desired Output:

The desired output of this project is a Java-based software application that provides an intuitive and efficient solution for managing customer orders and tracking daily collections for Quick-Bites Restaurant. The application will be easy to use, offering a user-friendly interface for order placement, cancellation, and collection tracking. The data will be stored in CSV files, ensuring data persistence and enabling the restaurant to maintain accurate records. By implementing this solution, Quick-Bites Restaurant aims to enhance its operational efficiency, reduce errors, and gain valuable insights into its daily revenue performance.