Business Applications Programming

Assignment 4 (40%)

Please complete the following exercise and submit your solution online by **Friday 25th November @ 5 pm.** Your submission should have two parts (A) your full application solution folder (zipped) to the designated dropbox on Blackboard, and (B) your source code in a word document to the Turnitin dropbox provided (no coversheet / screenshots required). All application files must be in the project debug folder. Note only one submission is accepted.

MyBagelShop

MyBagelShop Application:

Your client MyBagelShop Inc. (MBSI) is a regional chain of gourmet bagel bars, whose rapid rate of expansion has meant that the head office management team wishes to become more systematic, and systems orientated in their business approach. Your company has been commissioned to create a well-designed electronic point of sale (EPOS) application for use by their counter staff in processing customer orders. MBSI will use the EPOS to process and record sales of their gourmet bagels to their clientele, provide inventory management to keep on top of stock levels, and provide real-time and comprehensive reports that allows them to make better business decisions.

Each outlet offers thirteen different types of gourmet bagels, each of which are available to order in five different sizes (small, medium, regular, large, extra-large). With such a large array of gourmet bagels and sizes, the head office management team are particularly focused on ensuring that it has consistency in pricing across its outlets, and that each outlet keeps an accurate live count of its stock on hand.

To this end, each outlet should have a master bagel pricing list which the EPOS application must load from file into the application at the start of the day for use in calculating transactions and displaying on the UI. The application must also keep a live count of stock, with an opening stock read in from file at the start of each day, stocks of each item (bagel type/size) are reduced as they are sold during that day's trading, with a closing stock read out to file on application close (for use on next start of application).

Basic Flow of Events

Each customer order is compiled by the counter staff one-line item at a time. The customer chooses a bagel type and size which the staff member enters into the application, after which the price of the chosen combination is automatically displayed. Next the staff enters the quantity for the item that the customer wishes to order and presses the '*Add to Order'* button. If there is sufficient stock of the item ordered, a running total for the order is displayed. If there is insufficient stock to fulfil the customer's order, the system informs the server via a messagebox, along with the amount of stock that is available. When the server presses the '*OK*' button, the cursor returns to the quantity text box with the available stock level pre-entered for processing or writing over.

If the customer wishes to add another item to the order, the process as above is repeated. If at any point the customer wishes to cancel their order or change their order, the '*Clear*' button is pressed, and the form is reset so the order can be re-entered, or another customer order taken.

Once the customer's order has been taken in full, the employee presses the '*Complete Order'* button, after which the full transaction details are saved to text file, and a receipt in the form of a formatted message box (or some other form as desired) is displayed. This receipt displays the company name, its unique transaction number, the date/time for the order, along with the full transaction details i.e. quantity, type, size and price of each line item ordered, and a total order cost at the end of the receipt.

Additional Required Functionality

- The user should be able to search for transactions by its transaction number or by date of transaction. This search functionality is **required** to be processed using an appropriate collection and its methods. If the search term is found, the complete details of the transaction(s) is displayed.

- If the outlet manager wishes to view the trading figures up to that point for the day, he/she/they can generate a summary view to an appropriate control / form that lists the total number of bagels sold, the total sales value, the total number of transactions, and the average value of a transaction that day.

- If the outlet manager wishes to view the sales per item (amount & value of each bagel type/size sold) up to that point in the day, the system should generate and display a suitability formatted daily sales report in a separate form.

- If the outlet manager wishes to view the remaining stocks available, a suitability formatted management stock report should be generated and saved to text file (in debug folder).

Design Notes

- You are expected to use the programming & design components, constructs and best practices detailed over the 12 weeks in the module when developing your application.
- Each student is to provide their own naming of each of the 13 types of bagels to be sold, along with the separate pricing and stock levels for all 65 combinations of type/size of bagels.
- Handle any exceptions that could occur in your project & provide user input validation as needed.
- Include appropriate ToolTips & Access keys for the buttons.

MyBagelShop