BIT 3444
Homework 4: A DataReader Application
Due by Thursday, March 31, 2016 (11:55pm)

Homework 4 Submission: This is an individual homework assignment. You should save your database file in the folder: <your VB project folder>/bin/debug. Compress your complete VB project folder into a .zip file. The file name should have the format: 4_LastName_F.zip. Upload the .zip file to our homework submission site.

In this assignment, you will implement an electronic stock trading system. Your system will accept trading orders and based on those orders, it will keep track of pending orders, and execute trades when possible.

1. Database design (15 points)

Design a relational database for the stock trading system using Microsoft Access. You should determine the tables to be created, the attributes in each table, and the primary key in each table.

DO NOT define foreign keys in this assignment even they do exist. It will be easier for you to debug your VB program when foreign keys are absent.
Do NOT use special characters (e.g., space) in table/attribute names.
Do NOT use any Microsoft Access Reserved word (List of MS Access Reserved words: https://support.office.com/en-us/article/Access-2007-reserved-words-and-symbols-e33eb3a9-8baa-4335-9f57-da237c63eabe).

The system maintains common stocks records. Each stock is described by a ticker symbol (e.g., BAC), a company name, exchange venue (e.g., NYSE), listing date, sector, and industry.

The system also keeps track of stocks' daily closing records. Each closing record consists of a ticker symbol, a date, the opening price, the highest price of the day, the lowest price of the day, the closing price, the trading volume on that day, 1 day price change %, and last year's return rate %.

Users can use the system to place market order requests for stock trading. A market order is a request to purchase or sell a stock at the current market price. Each order request should include a ticker symbol, date and time of the request, type of order (buy or sell), quantity of stock to be traded, and order status (pending, completed, or cancelled).

2. Database population (5 points)

Use Microsoft Access to populate at least 10 records into each of the database tables created above. You must pick at least 10 stocks from this web site: http://www.1stock1.com/1stock1_112.htm , where you can find last year's return rate % for each stock. For other stock information, please use the Yahoo! Finance (http://finance.yahoo.com/stock-center/) and search by the stock ticker symbol. The sector and industry information can be found from the company profile page on Yahoo! Finance. Please make sure that you have at least 2 sectors and 2 industries per sector, and that each industry has at least 2 stocks.

After you populate the stocks table, you should populate daily records and trading requests based on either true market data or your imagination. Please make sure that you have a mixture of different order types and statuses for different stocks. A minimum of 10 records is required for each table.

For this assignment, you do NOT need real and accurate values. If you cannot find certain values, just make up the values.

3. SimpleBroker v1.0: A DataReader Application (80 points).

MB v1.0 is a DataReader application for placing stocking trading request. You must use a ***DataReader*** object to retrieve records from the database and a ***Command*** object to update the database. The following describes the components needed for the development.

(1) User interface: You should design your user interface to support the following functions. Label objects should be used appropriately to make your interface user-friendly. (10 points)

(2) Populate the stock ticker **ComboBox** control: The user should start with a **ComboBox** control (**StocksComboBox**) that displays the complete list of stock tickers retrieved from the database. (20 points)

(3) Populate the stock closing price **ListBox** control: After the user selects a stock ticker from **StocksComboBox**, a **ListBox** control (**ClosingPricesListBox**) should display a list of past closing prices of the selected stock. Each past closing price should indicate both a date and a closing price (e.g., “03/20/2013-$90.23”). The list should be ordered by descending dates. (10 points)

(4) Populate a **ListBox** control: After the user selects a stock ticker from **StocksComboBox**, another **ListBox** control (**TradingRequestsListBox**) will display a list of currently *pending* stock trading requests for the selected stock. Each request should indicate the order type and stock quantity (e.g., “BUY-2000”). The list is ordered by the order type first and then by quantity. (10 points)

(5) Define a **TradeRequest** class for all stock trade requests (20 points).

- The class has five properties: stock ticker symbol, request date and time, order type, quantity of stock to be traded, and order status. The order type property value can only be either “SELL” or “BUY”. The order status value can only be one of the three values {“PENDING”, “COMPLETED”, “CANCELLED”}. All letters must be upper-case letters. The stock quantity must be greater or equal to 0.
- The class has a constructor method that is used to initialize the property values of a new instance. The following two properties should be initialized as: request date time = current date and time; order status=”PENDING”.
- The class has a subroutine that is used to store a stock trade request to the database. The subroutine takes an **OleDBConnection** object as its input and uses the connection object to communicate with the database. The subroutine must use either data validation or exception handling technique to deal with any possible error related to the database connection.

(6) Placing a new order request:
   - The user selects an order type (buy or sell) from another **ComboxBox** control (**TradeTypeComboBox**), and enters a stock quantity to be traded in a **TextBox** control (**QuantityTextBox**).
   - By clicking a **Button** control (**SubmitButton**), a new stock trade request object must be instantiated using the constructor method defined in the **Request** class. The new request should be stored to the database using the subroutine defined in the **Request** class. (10 points)

4. Bonus questions:

(7) Sector selection: Another **ComboBox** control (**SectorsComboBox**) should be used to display all sectors in the database. The user starts with selecting a specific sector from **SectorsComboBox**. **StocksComboBox** will then only display those stock stickers in the selected sector. (20 points)

(8) Error handling: Use data validation or exception handling to prevent the following runtime errors from interrupting the program execution:

   a. Non-numeric values in **QuantityTextBox** (5 points)
   b. OleDbException when populating the ListBox and ComboBox controls. You can get such an exception if your database table is open from Access in the Design View when you try to execute your VB program. (5 points)