Final Exam Project – BCS 426 C# Programming

Due: 5/7/2020 @ 11:59pm

You will need to submit one Visual Studio solution AND an .MDF database file.

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!! REMEMBER IT IS NOT IN THE PROJECT DIRECTORY.

You DO NOT have to write comments for this program.

This is an exam (not a homework assignment) so no late submissions will be accepted.

VERY IMPORTANT – IF THE PROGRAM DOES NOT COMPILE THERE WILL BE MAJOR POINTS TAKEN OFF.

Overview

You will be writing a new Windows Presentation Foundation application. This application will use an SQL Server Express database to store the data.

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!!

Class – Ingredient

This class must be serializable in the way that we have been doing for the majority of the semester (DataContract etc…).

Member Auto-Implemented Properties (all public)

Name (string), Quantity (int), UnitOfMeasurement (string)

Member Method Signatures and Descriptions (all public)

Signature	Description

Ingredient()	Default constructor. Sets the values of each member variable to a default value.
C# properties	Write auto-implemented properties as described above
String ToString()	This method should return a string that contains the name.

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!!

Ingredients Database

Use SQL Server Express to store ingredient data.

IMPORTANT!!! – Do NOT use any other database expect SQL Server Express for this assignment.

You can refer to the following sets of slides on Blackboard for help:

- SQL Server Express Database and Visual Studio
- ADO.NET

Here is what you need to do:

- Create a database in SQL Server Express named **Ingredients**.

- **Ingredients** Table. This table should be setup as follows:

Column Name	Data Type
Name	Text
Quantity	Int
UnitOfMeasurement	Text

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!!

Main Window

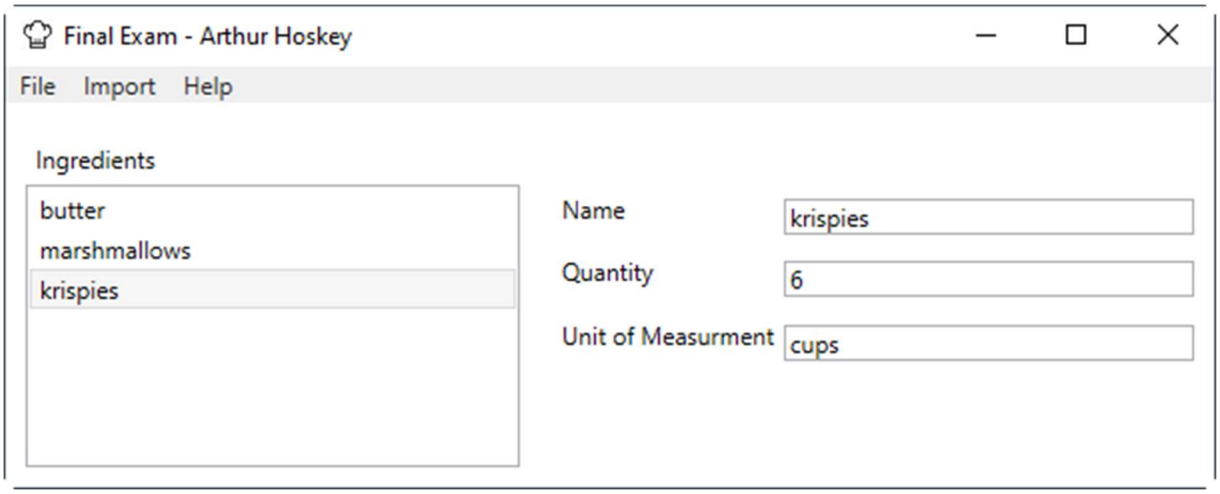

Figure 1 – Main Window

Here are the detailed specifications:

The main window displays a master/detail window. Data for the main window will be retrieved from the SQL Server Express database. The master is a list of ingredients in a ListBox (the ingredient's name shows in the ListBox). The details portion of the window shows data for the selected ingredient's name (from the ListBox). When the user selects an ingredient in the ListBox the data for that ingredient should be loaded into all the detail TextBoxes on the right side of the window.

- **Window Load Event Handling**
  - **Window Load**. When the window is loading it should populate the ingredients ListBox with data from the SQL Server database (not from a JSON file).
- **ListBox Event Handling**
  - **Ingredients ListBox**. When an item is selected in the ingredients ListBox the appropriate data should be displayed on the right side of the window. That means the TextBoxes must be populated with the appropriate data for the selected ListBox item. This data should be coming from the database (not from a JSON file).

Hint: You do not have to do it this way but you can use binding on the Text property of a TextBox to show data for a selected item. The XAML code below binds the Text property of the Quantity TextBox to the selected item's quantity field. For example:

```
Text="{Binding SelectedItem.Quantity, ElementName=listBoxIngredients,
Mode=OneWay}"
```

The above XAML code assumes that the SQL Server database column for quantity is named "Quantity" and the WPF ListBox is named listBoxIngredients. If the name you give the column in SQL Server is not Quantity the above code will not work.

- **Menus and Event Handling**. Should be defined as follows:
  - **File**. Should have the following menu items:
    - **Exit**. Closes the application.
  - **Import**. Should have the following menu items:
    - **Import Ingredients from JSON File**. Reads ingredient data from the selected ingredients JSON file into the database. You can deserialize data from the file into a List<Ingredient> instance. After deserializing data you can populate the database with data from that instance. Make sure you clear the database table BEFORE inserting the new data. This menu item should display a file open dialog to let the user select the JSON file to import from (similar to what you did on a previous assignment).
  - **Help**. Should have the following menu item:
    - **About**. Should show a dialog box that displays the program name, program version, and developer name.

Here are screenshots of each menu and the About dialog window:

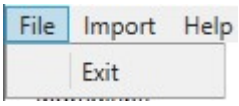

Figure 2 - File Menu

Figure 2 - Import Menu

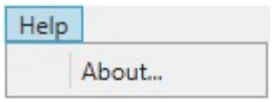

Figure 4 - Help Menu

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!!

Figure 5 - About Dialog Box

- **Tool Tip**. Add tool tips to the following:
  - Name TextBox
  - Quantity TextBox
  - Unit of Measurment TextBox

Here is a screenshot of the tool tip for the Unit of Measurement TextBox:

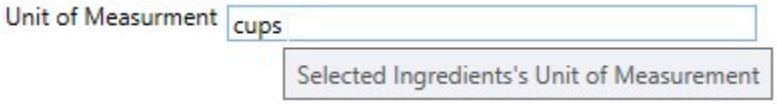

Figure 6 - Tool Tip for the Unit of Measurment TextBox

- **Icon** – Add an icon to the main window that is relevant to a recipe or food (whatever you can find that is the least bit relevant will be fine). Do not use the default icon. Here is a screenshot of the main window icon:

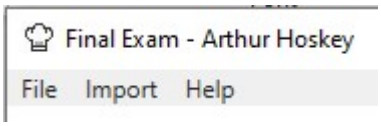

Figure 7 - Main Window Icon

Sample Ingredients JSON File (list of Ingredient)

```
[
  {
    "name": "butter",
    "quantity": 3,
    "unitofmeasurement": "tablespoons"
  },
  {
    "name": "marshmallows",
    "quantity": 10,
    "unitofmeasurement": "ounces"
  },
  {
    "name": "krispies",
    "quantity": 6,
    "unitofmeasurement": "cups"
  }
]
```

MAKE SURE YOU SUBMIT THE .MDF FILE WITH YOUR SUBMISSION!!!