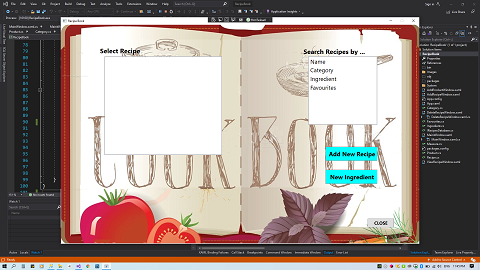
**Recipe Book – Functional Testing**

This project re-creates a recipe book compiled by my mother some 30 years ago, but this time around, instead of an old typewriter - I have used C# and SQL Server.

This document is a functional testing of that project. The first image shows the opening page of the application …

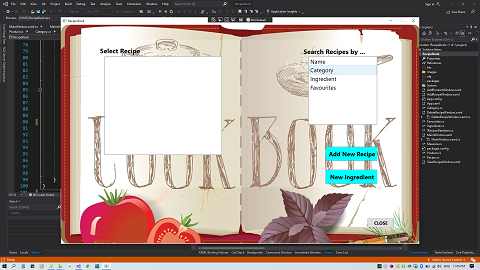


The right-hand listbox displays options for searching the recipe book, with choices for selecting recipes based upon Name, Category, recipes containing a particular Ingredient, or from a saved list of Favourites.

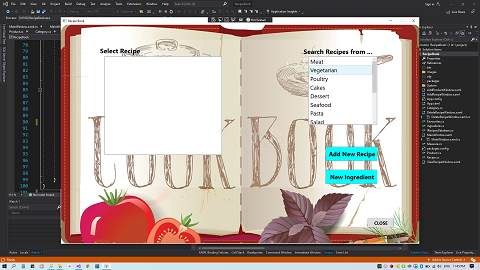
As no search option has yet been selected, the left-hand listbox is currently blank.

Buttons are also shown to ‘Add New Recipe’ and a ‘New Ingredient’ to the respective database tables. More on these later.

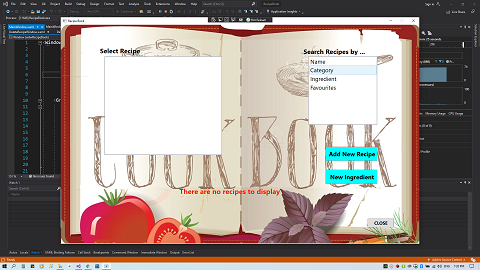
A ‘CLOSE’ button in the lower-right corner ends the application.



Selecting the Category option displays a new list of criteria in the right listbox, breaking the stored recipes down into the various recipe categories which Mum’s old recipe book was divided …

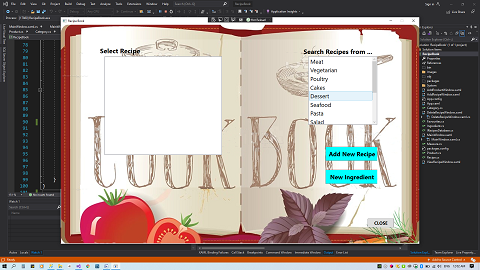


In this case, the Vegetarian category has been highlighted.

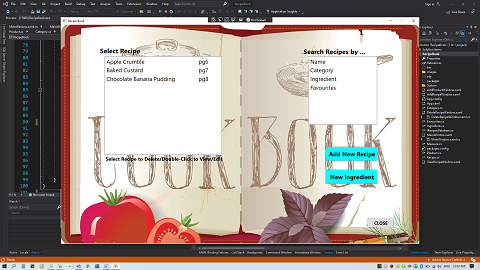


Clicking on this option now displays an error message, indicating that there are presently no Vegetarian recipes available within the Recipe database table – and why would there be ??

Let’s try a different category …

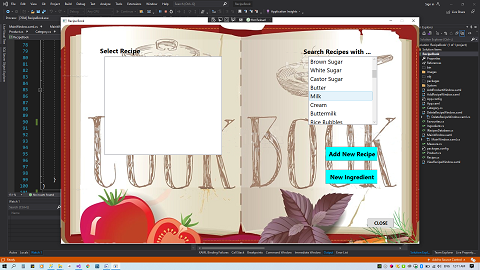


Selecting Desserts displays a list of available recipes for that category in the left-hand listbox. The recipes are listed alphabetically along with the corresponding page number (actually the RecipeID key in the database).

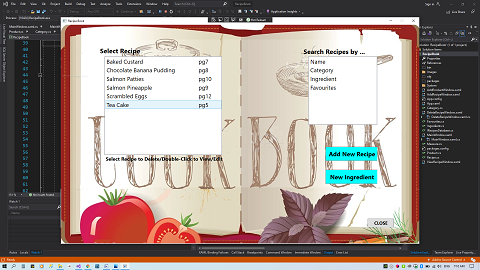


With available recipes, an instructional label now appears below the listbox directing the user to select a recipe, or double-click to View/Edit. We will come back to this in a moment.

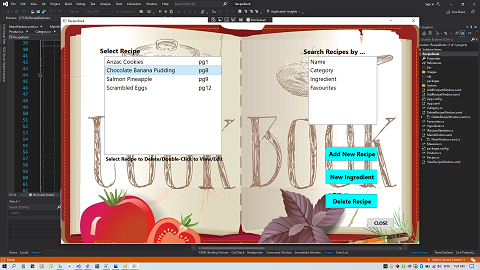
Returning to the right-hand listbox, recipes can also be selected based upon a component ingredient …



Choose ‘Milk’ and the left-hand listbox now displays a list of recipes containing that ingredient …



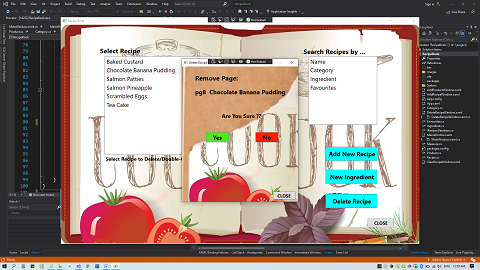
Recipes can also be selected based upon those which have been saved to a binary file of Favourites (more on that later) …



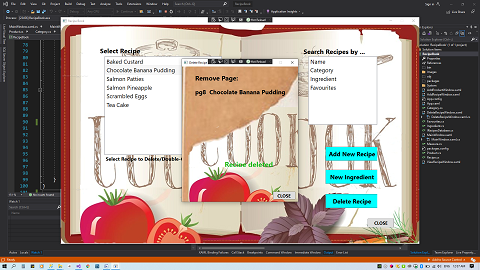
In the image above, it can be seen that selecting a recipe (by single mouse-click) causes a new button to be displayed, allowing the user to delete a recipe from the database.

Let’s press that button …

This opens a modal dialog window, allowing the user to remove the selected recipe from the database …



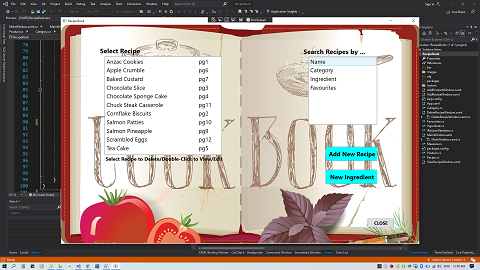
Pressing the ‘No’ button returns the user to the main menu again, but for now let’s press the ‘Yes’ button to delete the recipe …



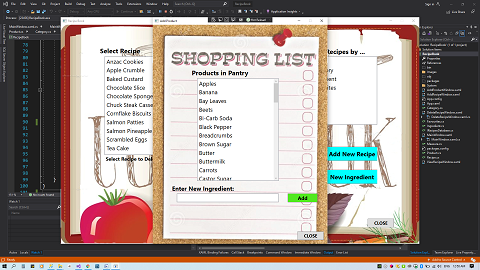
Chocolate Banana Pudding has now been deleted, and a screen-prompt is displayed (using an async Task.Delay method) for 3 seconds. An SQL transaction has been used to delete the Chocolate Banana Pudding as well as remove all of its’ ingredients, and it has also been removed from the Favourites binary file.

But do not worry, it can – and will – later be retrieved by using an SQL query to re-seed the database tables. Press the ‘CLOSE’ button to return to the main menu.

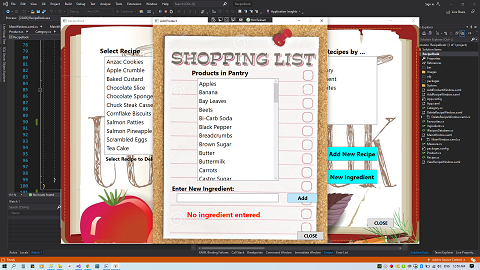
Recipes can also be selected by name, displaying an alphabetized list of all available recipes in the left-hand listbox …



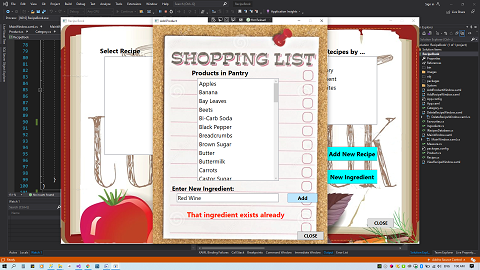
The database also contains a Product table to be used as ingredients. However, it is always possible that new recipes may require other ingredients. Thus, products can be added by pressing the ‘New Ingredient’ button to display another modal window.



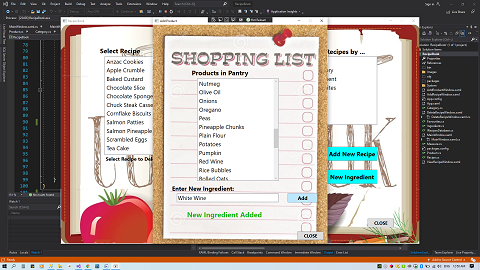
Pressing the ‘Add’ button without entering an ingredient displays a warning message …



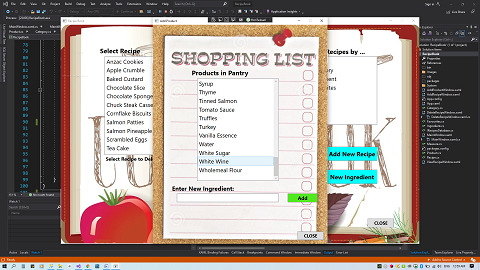
Entering an ingredient which is already in the table also displays a warning message …



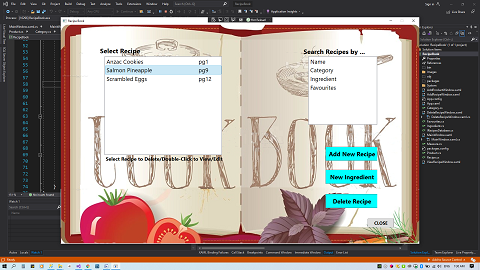
Let’s try a different ingredient …



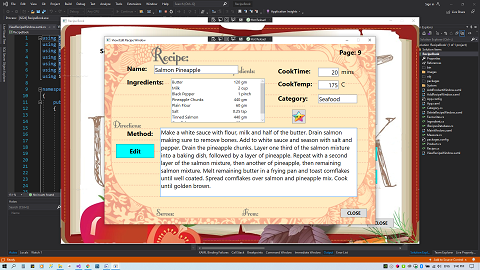
… White Wine is typed into the textbox, the ‘Add’ button is pressed and a success message is displayed.



Once the screen-prompt has faded, the alphabetized product list is re-displayed; the new ingredient can be seen in the list and is now in the Products table in the database, ready to be used in a recipe.

Returning to the main menu, we will now select and view a recipe …

Double-click the recipe in the left-hand listbox to open the ViewEditRecipe modal dialog box and view the selected recipe …

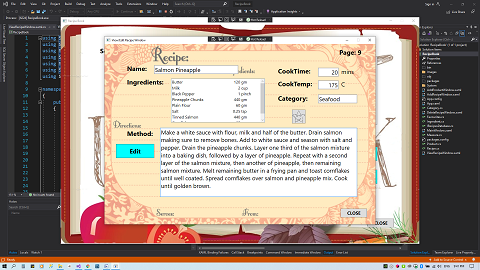


The recipe is retrieved from the database and textboxes are used to display the details, such as name, cooking time, cooking temperature, recipe category and method. A DataGrid is used to display the ingredients, together with quantities and measurement units.

Incidentally, note that the RecipeID number is displayed in the upper-right corner, again as the page number.

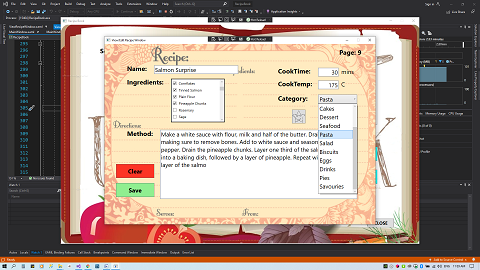
This recipe (a childhood favourite) is marked with a colourful star icon, to indicate that it has already been set as a Favourite; as such it appears in the list when Favourites are selected on the Main Menu.

Pressing the icon (a XAML button) replaces the colourful star with a bland, grey one …



… this recipe has now been removed from the Favourites list, and the binary data file.

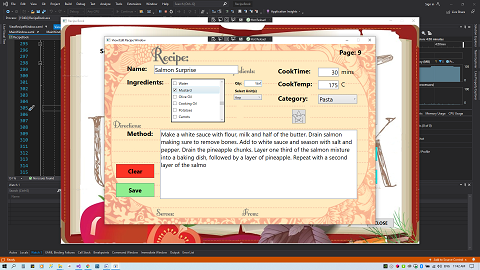
Pressing the ‘Edit`’ button puts the page into edit-mode. The ingredients are displayed in a SyncFusion CheckListBox control. Existing ingredients are pre-checked, and displayed first in the list and others can be selected and de-selected …



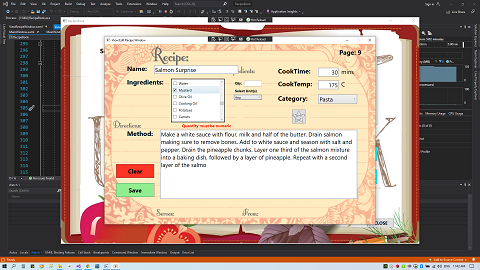
The textboxes are now editable, and a drop-down combo-box menu replaces the category textbox.

For each added ingredient, a textbox and combo-box are presented to the user to add quantity and measurement unit details. Single-clicking an ingredient line displays previously set values, if the item has been checked; double-clicking selects/de-selects an ingredient.

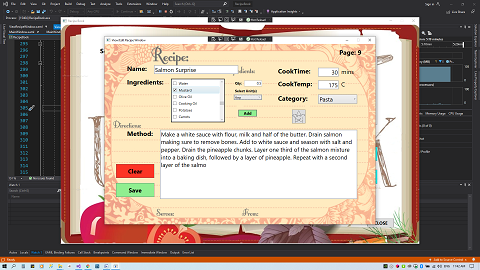
The Quantity entry must be numeric, non-zero and non-negative …



… or an error message will be displayed …



If there are no errors, an ‘Add’ button appears once quantity is entered and a unit of measure has been selected …



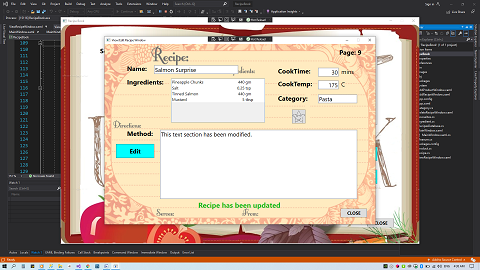
The newly selected ingredient is added to the recipe ingredients …



… and a confirmation message is displayed.

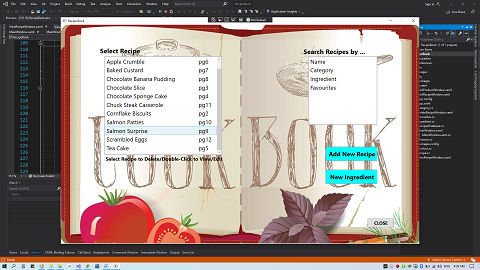
Although not displayed here, this page has error-checking on all input fields to ensure that valid numeric values are entered where necessary, as well as checking that any modified recipe name does not already exist in the recipe list.

Once all required changes have been made, pressing the ‘Save’ button displays a confirmation message and stores the newly-modified recipe into the database (or press ‘Quit’ to restore the recipe to its’ original status).

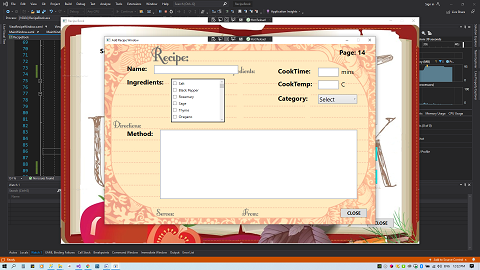


The modified recipe is displayed in the DataGrid panel.

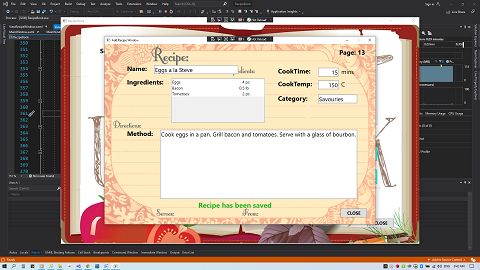
Returning to the main menu, we can see that the newly edited recipe appears in the list under its’ new name, Salmon Surprise …



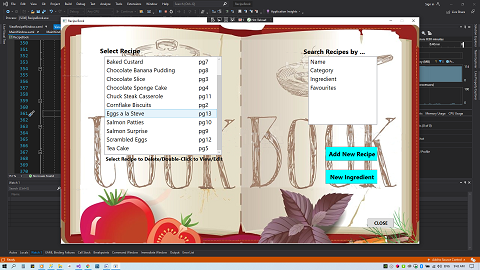
Now let’s create a new recipe by selecting the button marked ‘Add New Recipe’. This takes us to a blank recipe page; here all details can be entered in exactly the same manner as for the View/Edit page …



When (i) all textboxes have some form of input, (ii) the category combo-box has a selection, and (iii) at least one ingredient has been added, the ‘Save’ and ‘Clear’ buttons will be displayed. Press the ‘Clear’ button to start over, or press the ‘Save’ button to store your new recipe into the database, and display it in the DataGrid panel ...



Returning to the Main Menu once again, we can see the new recipe, Eggs a la Steve, has been added to the list …



**The End**

**CREDITS**

Many thanks to fellow student, David Stern, for his help in solving some of the more technical issues I encountered throughout this project.

Also, thanks to Batrice Ramsey, Global Product Consultant of SyncFusion. This company produces controls which, in my personal opinion, are far superior to those found as standard controls for the C# programming language (as well as other languages).