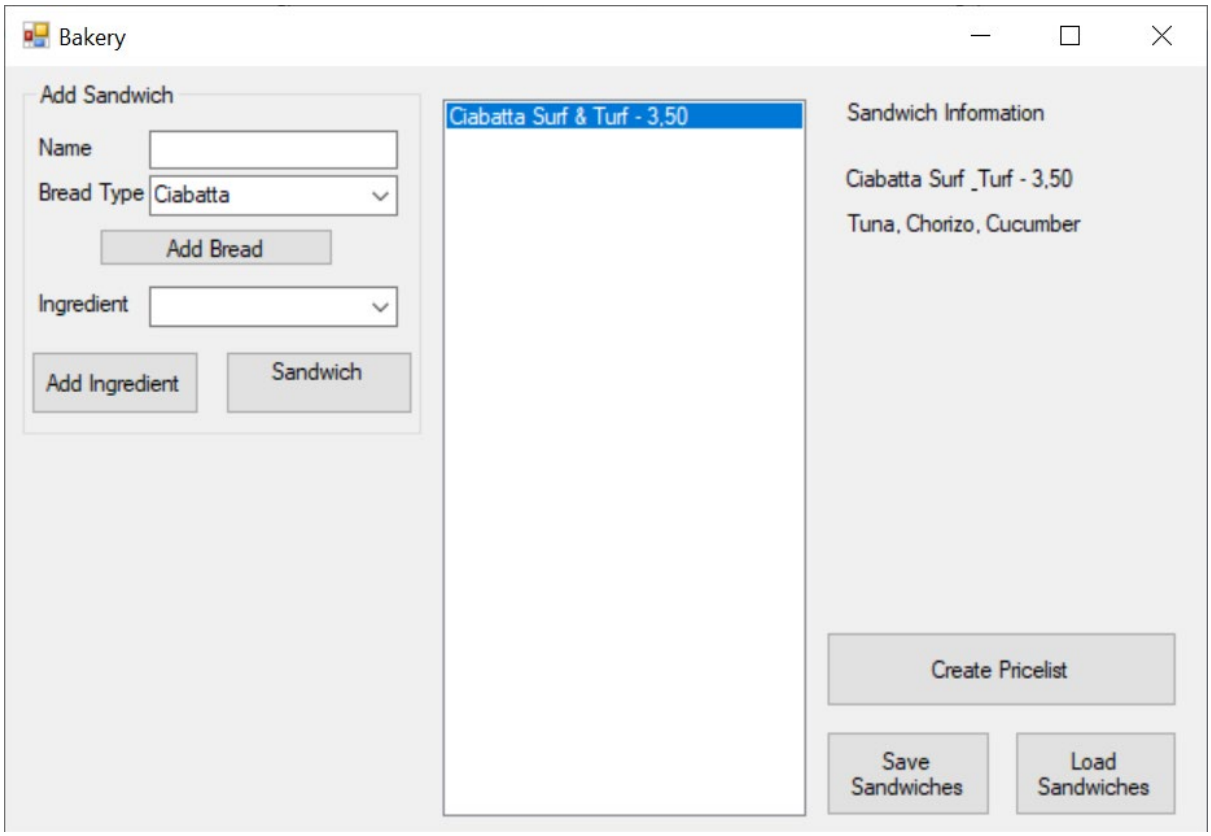
Assignment Bakery

A bakery sells different kinds of sandwiches to companies. They would like to have an application in which they can create different sandwiches and print pricelists. You are going to program this application in this assignment.

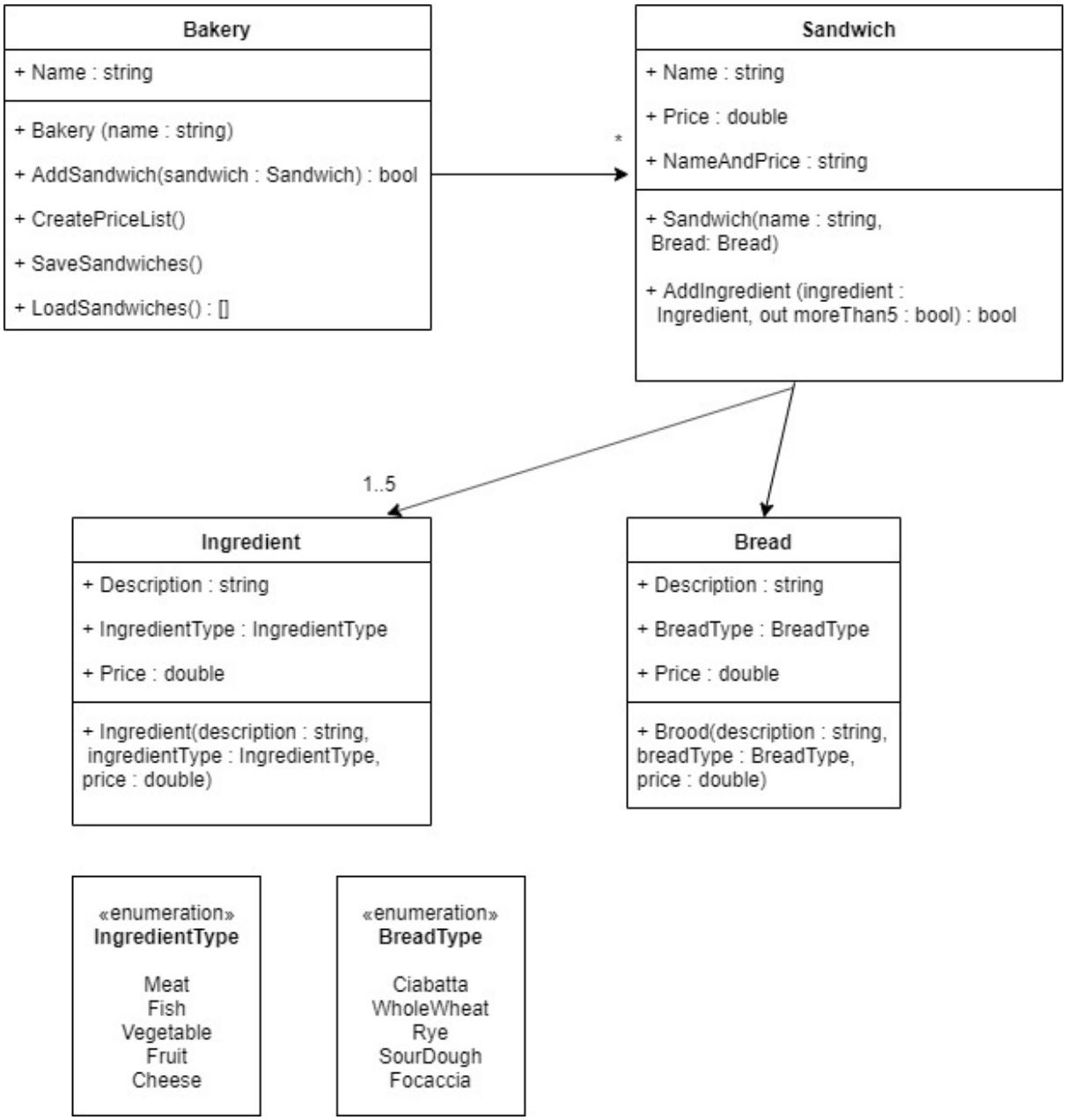
Here you see what the GUI should look like (it will be provided in the start up project).



Functionalities of the application

- Sandwiches can be added. Enter a name for the sandwich and choose the kind of bread used for the sandwich.
- If the base sandwich is added, you can add ingredients to it (with a maximum of 5 and an ingredient can be only added once.
- After adding the ingredients you can add the sandwich to the list by clicking on the “Sandwich Finished” button.
- The added sandwhich is now visible in the listbox and when you click on the sandwich you see all its details on the right.
- A pricelist can be generated (it creates a txt-file).
- The sandwiches can be saved.
- It must be possible to load previous saved sandwiches into the program.

Class diagram of the application



Step 1

Open the start-project which is part of this assignment. The GUI is already there. Now implement the class diagram, but you don't have to implement the methods just yet.

Step 2

In the class Bakery, you are going to implement the method "AddSandwich". This methods adds a new sandwich. The name of the sandwich must be unique. When this is not the case, the method should return false and the sandwich will not be added. Otherwhises, the method return true and the sandwich is added to the list of sandwiches.

Step 3

The next method we are going to implement is “CreatePriceList”, in the Bakery class. All sandwiches from the bakery will be saved into a textfile. Use the property “NameAndPrice” from the Sandwich class, make sure the name of the sandwich and the price are seperated with a “ – “. For example: “Sandwich Deluxe – 5,50”.

Step 4

The method “SaveSandwiches” will save all sandwiches in a binary file named “sandwiches.bin”. Implement this method.

Step 5

The method “LoadSandwiches” will load all sandwiches from the binary file named “sandwiches.bin”, only if this file exists of course. The method returns a List<Sandwich>

Step 6

We now move to the class Sandwich. Implement the method AddIngredient. This method returns a bool which will be false when the ingredient already exists on the sandwich. This method also has an out parameter called “MoreThan5”. This will return false when there are more than 5 ingredients on a sandwich. Implement this method.

Step 7

We now programmed all methods. Now we have to make the GUI work!
Make sure you declare a global Bakery object in your form. Initialise it in the constructor of the form.

Step 8

Add a method to your form code “FillBreadTypesCombobox” which will fill your combobox with the available bread types. You can use the following lines of code:

```
List<Bread> breadTypes = new List<Bread>();

breadTypes.Add(new Bread("Ciabatta", BreadType.Ciabatta, 3));
breadTypes.Add(new Bread("Whole Wheat", BreadType.WholeWheat, 2));
breadTypes.Add(new Bread("Rye", BreadType.Rye, 2.5));
breadTypes.Add(new Bread("Sour Dough", BreadType.SourDough, 3.5));
breadTypes.Add(new Bread("Focaccia", BreadType.Focaccia, 3.25));

cbBreadTypes.DataSource = breadTypes;
cbBreadTypes.DisplayMember = "Description";
```

Step 9

Add a method to your form code “FillIngredientsCombobox” which will fill your combobox with several ingredients. Use the following lines of code:

```
List<Ingredient> ingredients = new List<Ingredient>();

ingredients.Add(new Ingredient("Cucumber", IngredientType.Vegetable, 0.05));
ingredients.Add(new Ingredient("Tomato", IngredientType.Vegetable, 0.15));
ingredients.Add(new Ingredient("Cheddar", IngredientType.Cheese, 0.2));
ingredients.Add(new Ingredient("Ham", IngredientType.Meat, 0.15));
```

```
ingredients.Add(new Ingredient("Chorizo", IngredientType.Meat, 0.25));
ingredients.Add(new Ingredient("Apple", IngredientType.Fruit, 0.1));
ingredients.Add(new Ingredient("Tuna", IngredientType.Fish, 0.2));
ingredients.Add(new Ingredient("Smoked Salmon", IngredientType.Fish, 0.4));
ingredients.Add(new Ingredient("Cream cheese", IngredientType.Cheese, 0.15));
ingredients.Add(new Ingredient("Gouda cheese", IngredientType.Cheese, 0.1));

cbIngredients.DataSource = ingredients;
cbIngredients.DisplayMember = "Description";
```

Make sure this method will be called in the constructor of the form.

Step 10

Program the button “AddBread”. This will add a base sandwich which we can use to add ingredients to. When the “AddBread” button is clicked, a new Bread object will be created with the given name and the choosen breadType. Check if there is already a sandwich which the name, if so, then the user should now he should choose another name.
Because we will work with this bread in other methods, it is a good idea to declare it globally, so you can use it in other methods, name this variable “CurrentBread”.

If the user did not fill in a name or choose a breadtype you should give a message.

Step 11
The next button we program is “Add Ingredient”. This will make sure the ingredient will be added to the sandwich. There should be a message if the ingredient is already on the sandwich or if the maximum of 5 ingredients is exceeded.

Step 12
Program the button “Sandwich Finished”, which will make sure the listbox will be filled again and the just added Sandwich will also be visible in the listbox. Use the property “NameAndPrice” as displaymember for the listbox.

Step 13
When the user clicks on a sandwich in the listbox, their should be information about this sandwich on the right. See the GUI screenshot in the beginning of the assignment.

Step 14
Now program the buttons “Create Pricelist” and “Save sandwiches” and “Load sandwiches”.