National Taipei University of Technology

Windows Programming (Fall 2018)

Homework #2

Deadline: 10/9 (Tue), before 4:00 PM

Attention: the rules as follows are strict, you get zero point if any of the following rules are violated.

1. No late homework is accepted.
2. Plagiarism is not allowed – every homework must be completed by your own.

Continuing the last homework, we will keep working on the customer's program. We will add a new start-up form (window) to allow the user to choose from either Customer's program (frontend) or Restaurant's program (backend). In the customer's program, we will add a delete function to the list of orders on the right (DataGridView). In addition, we will add a RichTextBox to display a meal's detailed description.

This homework is incremental, meaning that you need to make sure that your design is easy to maintain. You will need to extend your program in the future.

[1 pts] Start-Up Form GUI

Because we have two kinds of programs in this homework, we need a start-up form to allow the user to choose the program to start. There are three Buttons in the start-up page, the frontend (customer's program), backend (restaurant's program), and exit buttons.

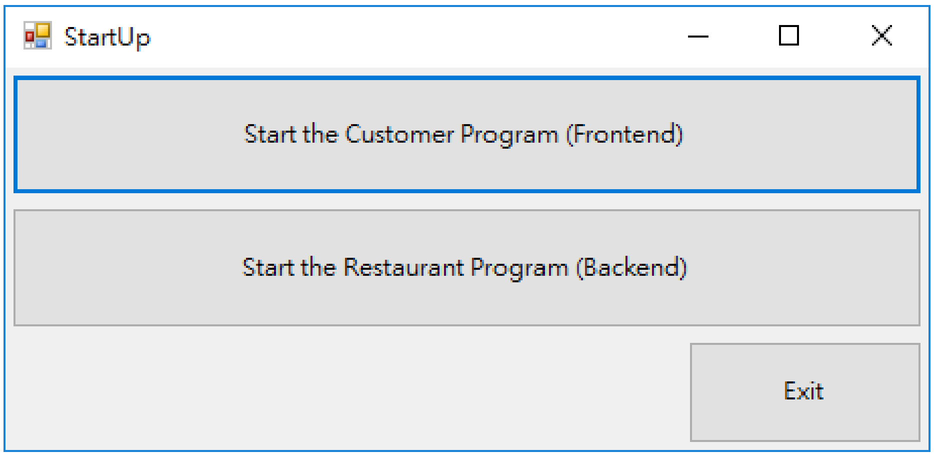
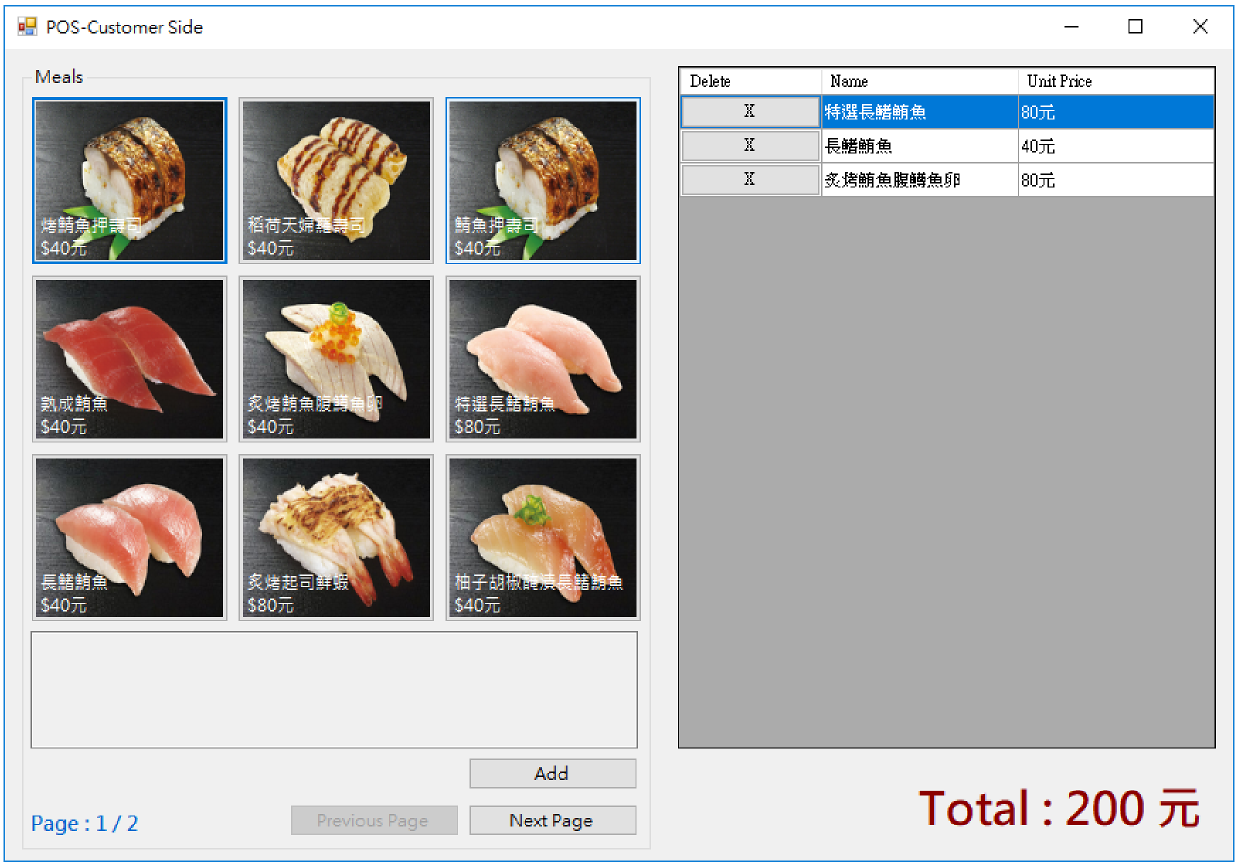

StartUp
Start the Customer Program (Frontend)
Start the Restaurant Program (Backend)
Exit

Figure 1.The GUI of Start-Up Form in this homework

[3 pts] Customer's Form GUI

Add a delete **Button** on the first column of each row in **DataGridView** so that the user can delete a particular order. In addition, we add a RichTextArea to display the detailed description of the meal (shown under the meals groupbox).



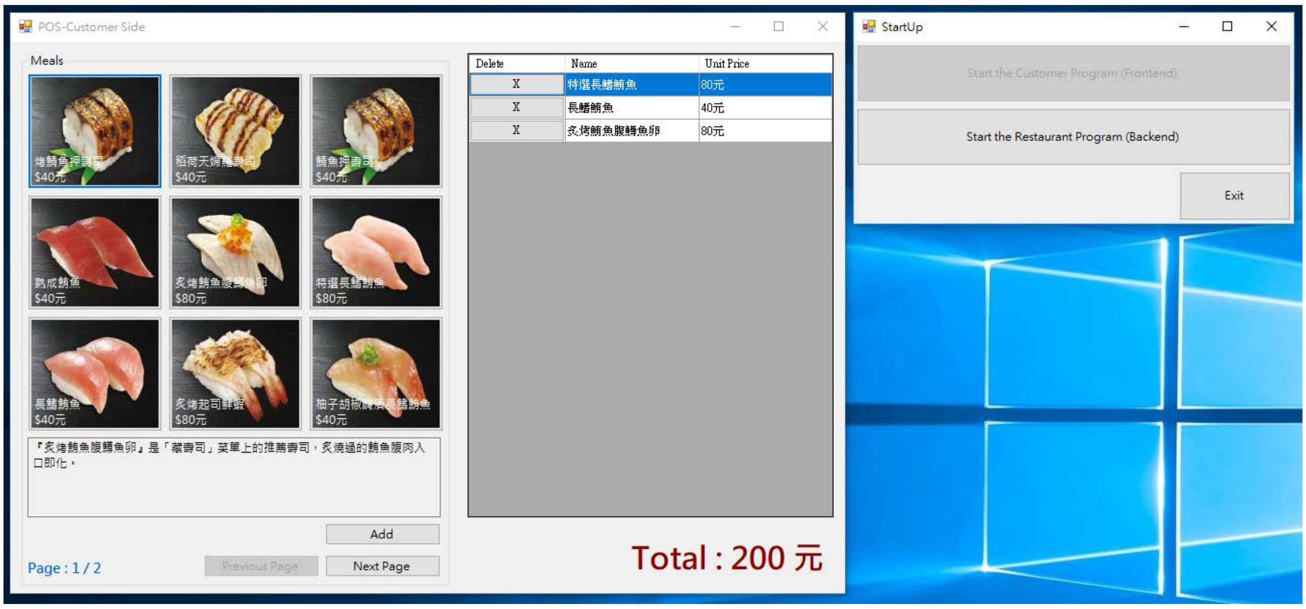
Delete	Name	Unit Price
X	特選長鰭鮪魚	80元
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

Total : 200 元

Figure 2. The GUI of Customer’s Program

[2 pts] Modeless Customer’s Program Form and Restaurant’s Program Form

Forms and dialog boxes are either *modal* or *modeless*. When using a modal form (or dialog), the form must be closed (or hidden) before you can continue working with the rest of the application. Modeless forms, on the other hand, allow you to change the focus between one form and the other without having to close the initial form. The user can continue to work elsewhere in any application while the form is displayed. Our customer’s program is a **modeless** form. That means after you click “Customer (frontend)” button in the start-up form, you can still focus on start-up form and click another button on the form. The customer’s program can have only *one instance*, which means you cannot have two customer’s programs running at the same time (note: please do NOT use singleton pattern). If you open a customer’s program, the “Customer Program (Frontend)” button in the start-up form should be disabled. If you click exit button in the start-up form while the customer’s program is opened, all forms (start-up form and customer’s program form) should be closed simultaneously. Since we do not have a Restaurant’s program yet, you can simply display a **MessageBox (a kind of form)** with a message “Coming Soon.” The **MessageBox** should be **Modeless** too, and should also have **only one instance.** Please enable/disable the buttons as necessary. For example, when opening the customer’s program, the Customer (frontend) button should be disabled (Shown in Figure 3). And when closing the Customer program, the corresponding button should be enabled again.

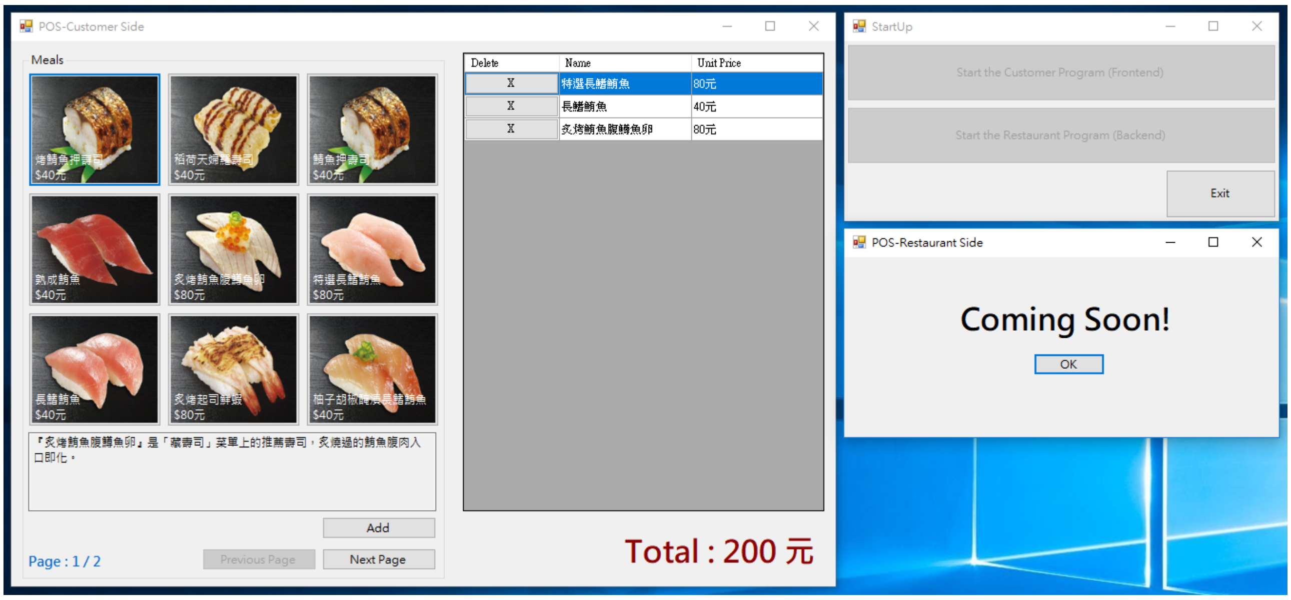


『炙烤鮪魚腹鱒魚卵』是「藏壽司」菜單上的推薦壽司，炙燒過的鮪魚腹肉入口即化。

Delete	Name	Unit Price
X	特選長鰭鮪魚	80元
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

Total : 200 元

Figure 3. After clicking the “Customer (frontend)” button you can still focus on the Start-Up form. The Customer (frontend) button in start-up form should be disabled because the customer’s program has been opened.



『炙烤鮪魚腹鱒魚卵』是「藏壽司」菜單上的推薦壽司，炙燒過的鮪魚腹肉入口即化。

Delete	Name	Unit Price
X	特選長鰭鮪魚	80元
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

Total : 200 元

Coming Soon!

Figure 4. The restaurant’s side program (a simple dialogue for now) and the customer’s side program can be displayed simultaneously, and you can focus on either of the form as you wish. Similarly, the Start Restaurant Program button is disabled because the restaurant’s side program has been opened.

[3 pts] Meal Button with Image

You should make the meal buttons display image of the meal. You **should not** use designer to add an image to the button. Instead, you need to load images to the buttons programmatically. You can add a data member to your meal class called “imageRelativePath” which record the relative path of the image in the project. Your view class can use this “imageRelativePath” to show the image on the button. **Do not** record absolute path of the image because a different computer would not have the same path.

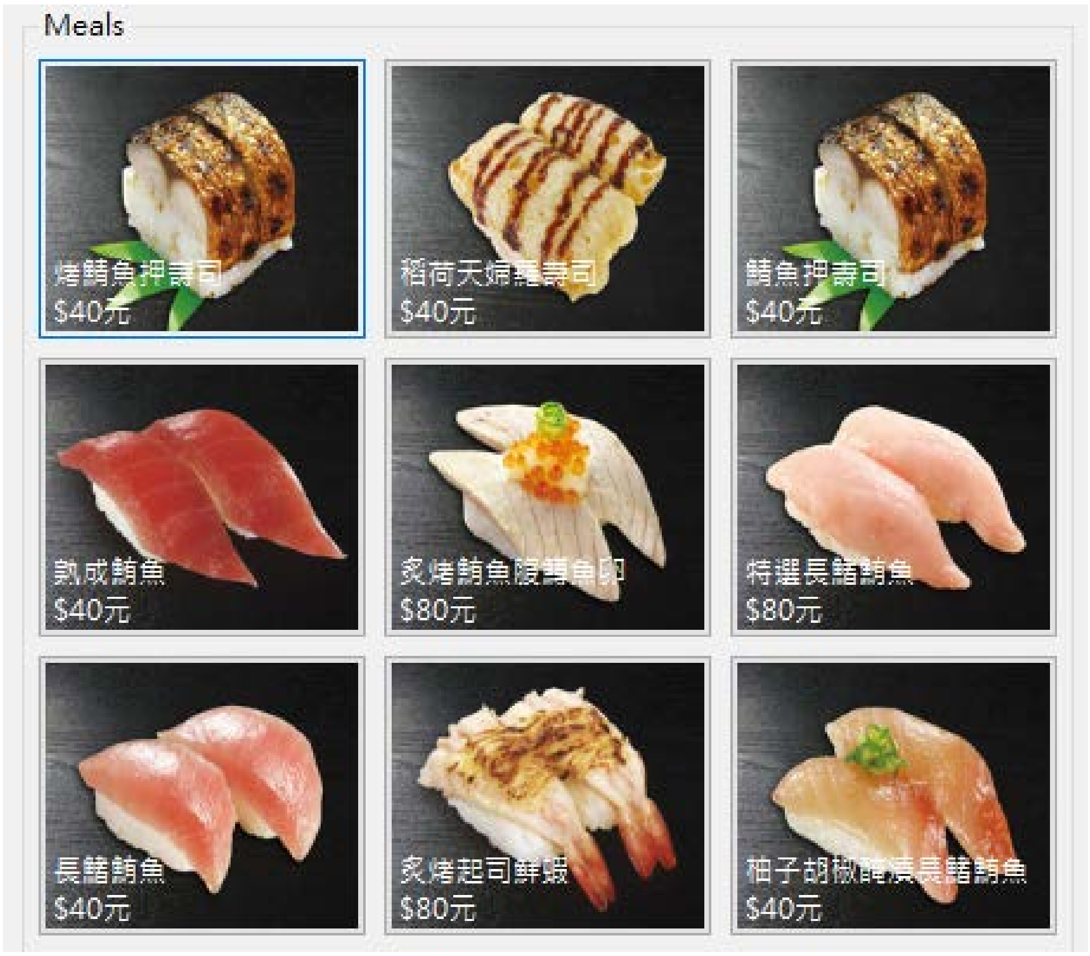
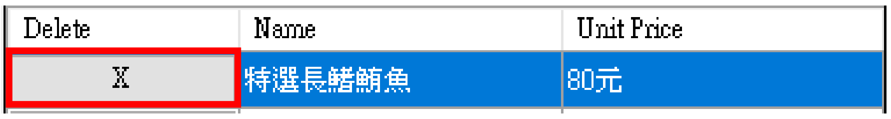


Figure 5. Buttons with meal image.

[4 pts] Deleting Orders

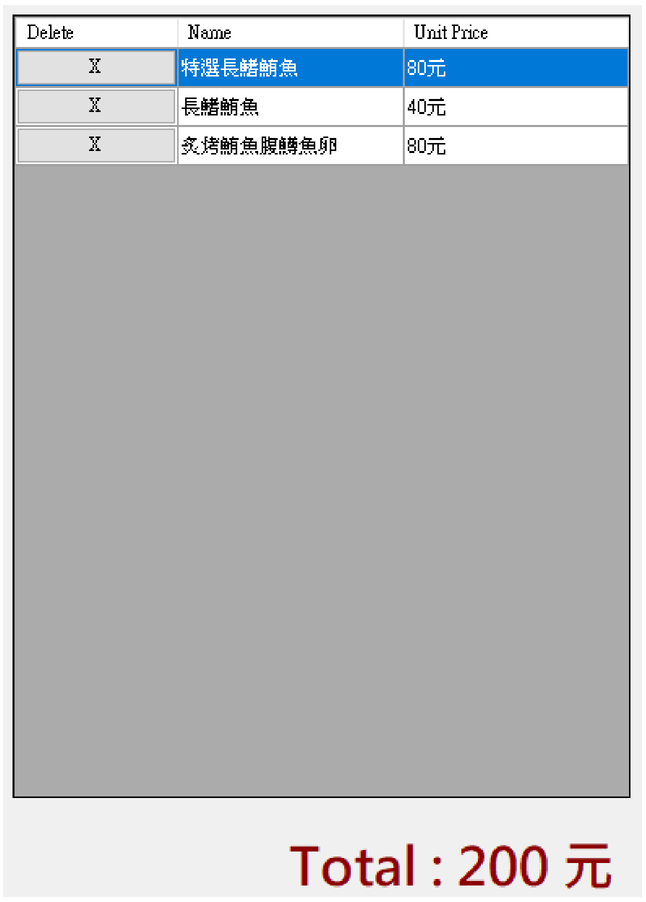
When you click a particular order’s delete **Button**, the order should be deleted. After deleting the meal, you should update the total price of the orders. You also need to remove item in the model.



Delete	Name	Unit Price
X	特選長鰭鮪魚	80元

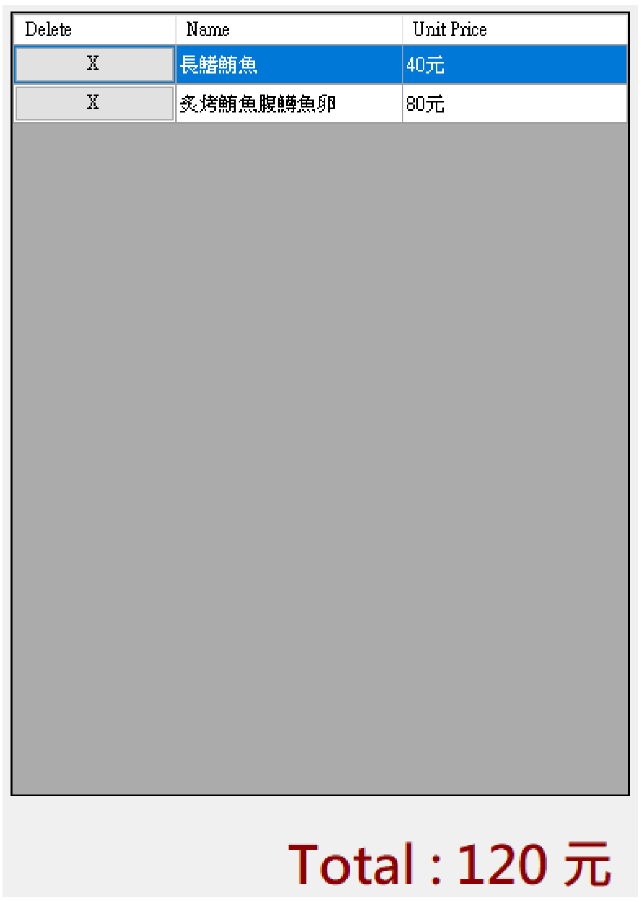
Figure 6. The first column is a delete button

.



Delete	Name	Unit Price
X	特選長鰭鮪魚	80元
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

Total : 200 元



Delete	Name	Unit Price
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

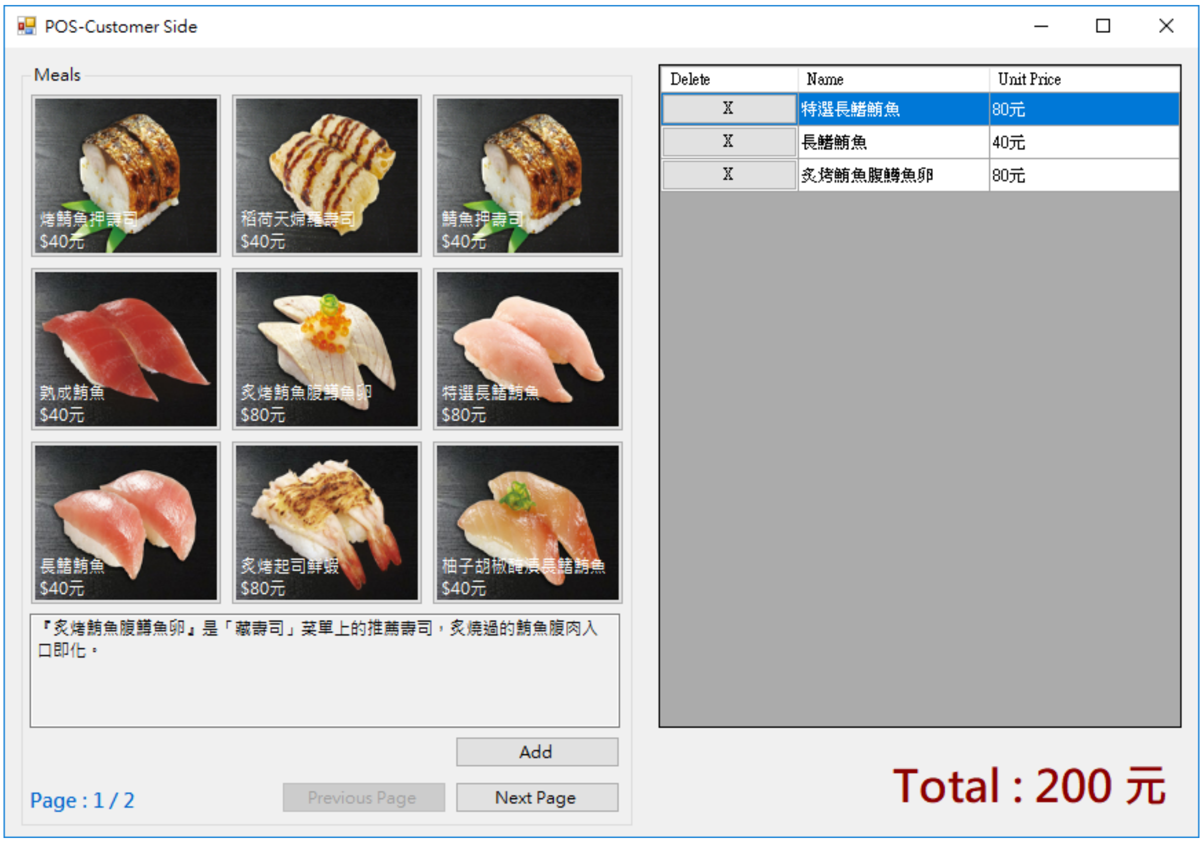
Total : 120 元

Figure 7. Before deleting an order.

Figure 8. After deleting an order.

[2 pts] Meal’s description

The **RichTextBox** should display the description of the meal, when the meal button has been selected.



『炙烤鮪魚腹鱒魚卵』是「藏壽司」菜單上的推薦壽司，炙燒過的鮪魚腹肉入口即化。

Delete	Name	Unit Price
X	特選長鰭鮪魚	80元
X	長鰭鮪魚	40元
X	炙烤鮪魚腹鱒魚卵	80元

Figure 9. When the meal has been selected, the **RichTextBox** displays the description of the meal.

[1 pts] Closing Start-Up Form

When the user clicks the exit button in the start-up page, the whole program should be closed. If any programs (either the customer’s or Restaurant’s program) are still opened, they should be closed simultaneously.

[2 pts] UI Control

Controls should be properly enabled/disabled depending on the state of the program.

[5 pts] Model View Controller (MVC) Pattern and Presentation Model Pattern

This requirement will be graded by the quality of your MVC design. You should make your UI as thin as possible to avoid the fat view smell. For example, the meal Buttons should not handle the logics of recording data to DataGridView. Instead, the handler should simply delegate the logic to the model (by calling a method of the model) or presentation model. In addition, a good design must enforce a one-way dependency. You are encouraged to follow the reference class diagram shown in Figure 5.

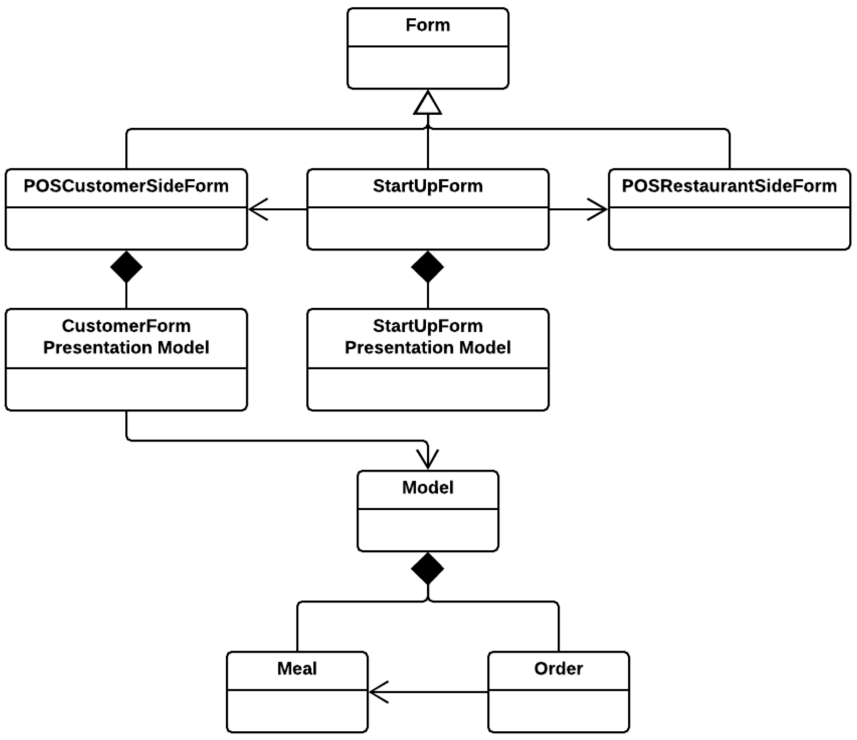


Figure 10. A reference class diagram

[7 pts] Code Quality

The grading of code quality will be based on whether your programs have bad smells. Your score depends on the smell density (number of smells per thousand lines of code) of your code. **You have to ensure that the code generated by Windows Form Designer do not infringe your coding standard**.

[2 pts] Summary

You are required to turn in your homework summary with time log (the time in hours that you spent in this homework, including documentation, as precisely as possible) in Word format. Please download a template from the instructor's website.

Additional Information

The following information is useful for your homework. If you need more details, please look at the textbook or MSDN website.

1. Add a background image to a button. – You can use the following code to add an image to a button as its background.

   The following code assume your image is stored at (Project Directory)/Resources/1.png

```
using System.IO;
…
Button buttonYouWantToAddAbackgroundImage = new Button();
string projectPath = Path.GetDirectoryName(Path.GetDirectoryName(System.IO.Directory.GetCurrentDirectory()));
const string IMAGE_RELATIVE_PATH = "/Resources/1.png";
buttonYouWantToAddAbackgroundImage.BackgroundImage = Image.FromFile(projectPath +
IMAGE_RELATIVE_PATH);
buttonYouWantToAddAbackgroundImage.BackgroundImageLayout = ImageLayout.Stretch;
```

2. Add a button column to DataGridView. – You can use following code to add a button column to DataGridView.

```
DataGridViewButtonColumn deleteColumn = new System.Windows.Forms.DataGridViewButtonColumn();
deleteColumn.HeaderText = "Delete";
deleteColumn.Name = "deleteColumn";
deleteColumn.Text = "X";
deleteColumn.UseColumnTextForButtonValue = true;
recordDGV.Columns.Add(this.deleteColumn);
```

   For more information, please visit MSDN :
   https://msdn.microsoft.com/en-us/library/system.windows.forms.datagridviewbuttoncolumn(v=vs.110).aspx

3. Delete a row in DataGridView – You can use the following code to delete a row in DataGridView. The following code assume that you want to remove the row at index 0.

```
recordDGV.Rows.RemoveAt(0);
```

   For more information, please visit MSDN :
   https://msdn.microsoft.com/zh-tw/library/system.windows.forms.datagridview.rows(v=vs.110).aspx

4. Show a modeless windows form – You can use following code to show a modeless windows form.

```
Form customerProgram = new CustomerPOSForm();
customerProgram.Show();
```

   For more information, please visit:
   http://www.thejoyofcode.com/Modeless_Dialogs_in_WinForms.aspx