National Taipei University of Technology

Windows Programming (Fall 2018)

Homework #4

Deadline: 11/7 (Wed), before 2:00 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.

In this week you are required to write the unit tests for your program. Please write the unit tests for your previous programs first and then complete the functions required in this homework and write the unit tests for them.

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.

[10 pts] Unit Test

Unit testing is graded based on the quality of your test cases, not the quality of your production code. To improve test-case quality, your test code should have code coverage as close as 100% as possible. Please follow the following steps when you write test cases:

Step 1 Count and report the number of classes and methods that are GUI related (i.e., classes and methods that directly control or update GUI widgets). Do the same thing for the classes and methods that are not GUI related (i.e., classes and methods that are designed for model). Note that if you separated your view and model classes nicely, GUI related methods should all reside in the view classes and non-GUI related methods should all reside in the model classes.

Step 2 Write test cases for your program. If you find a method difficult to test, you may revise the method before writing test cases for it. Note: for every testable method (i.e., non-GUI related methods), you must write a test method to verify the correctness of the method (except for methods that are too simple to break). In addition to the testing of methods, you should also test the integrated behaviors of the model (i.e., possible user scenarios).

Step 3 Please count the number of methods that are tested and the time spent in Step 2.

Step 4 Run tests and report the number of failed tests.

Step 5 Fix bugs, if there are any.

Step 6 Please report the time spent in Step 5.

Step 7 Go to Step 4 until all bugs are fixed.

[8 pts] Cascaded Delete – Category

If you delete a category from Category Manager, the meals belonging to that category should also be deleted. Note that, to make it simpler, if there are any orders containing the meal that belongs to the category to be deleted, the deletion is not allowed.

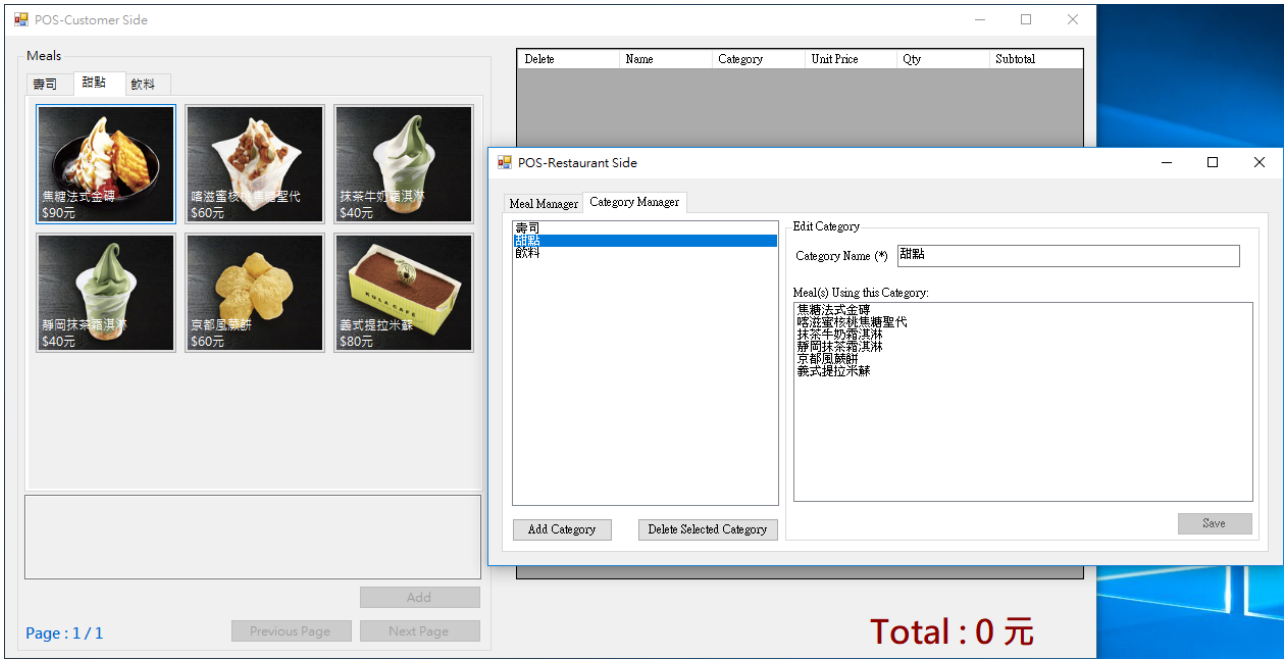
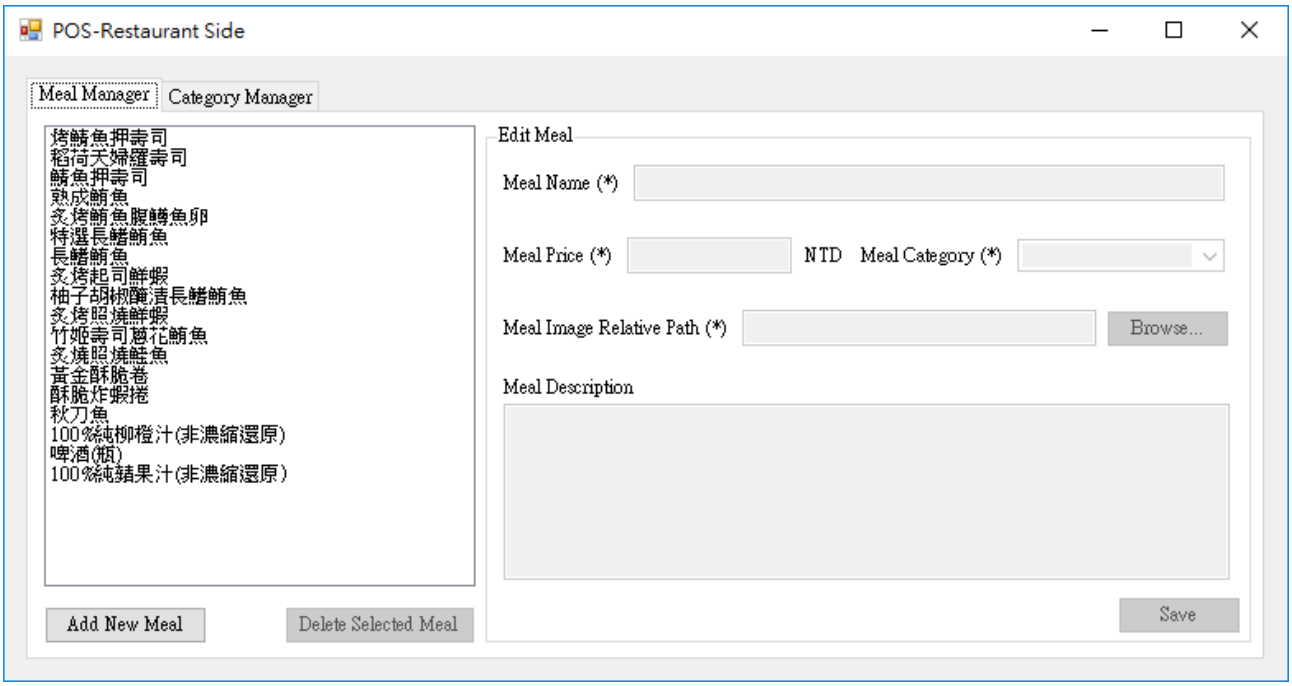


Figure 1. Before Deleting a Category.



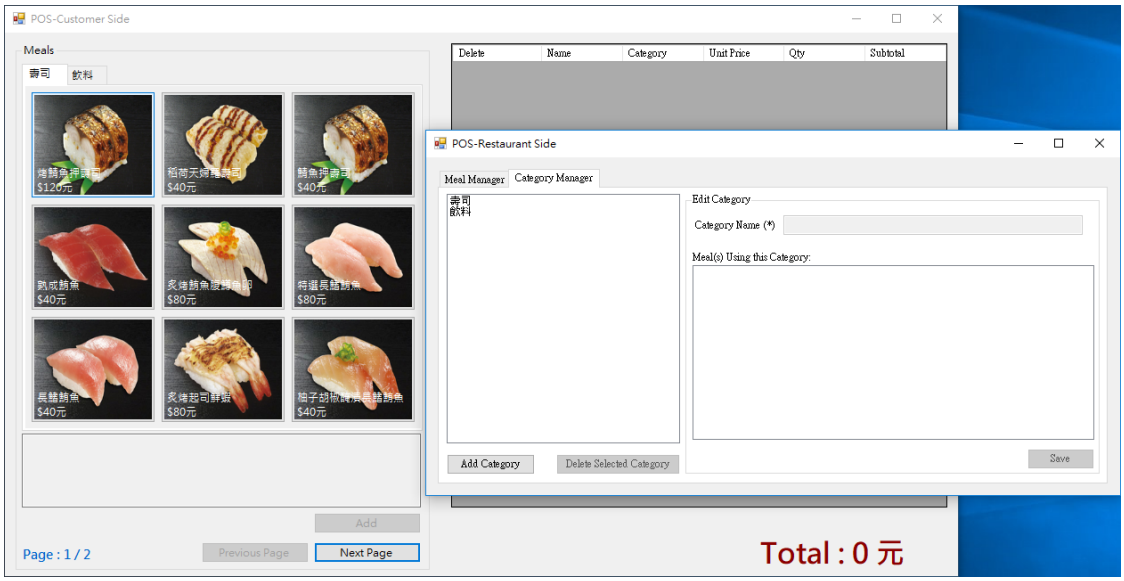


Figure 2. After Deleting a Category

[5 pts] MVC and Presentation Model

Ideally, the view should be as thin as possible. Business logics should reside in the model. However, there are times that the view also needs view-dependent logics (e.g., enabling/disabling buttons). In that case you should put these logics into **Presentation Model** to prevent from producing the **fat view** bad smell.

There should be a one-way dependency from StartupForm to the two client forms, CustomerSideForm and ResturantSideForm.

[2 pts] UI Control

Controls should be properly enabled/disabled depending on the state of the program. Note that if the “Meal Price” is not a proper integer, the “Save” button of the Meal Manager should be disabled.

[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.

Notice:

You need to put your unit test files to an independent “project”.