SCIT

School of Computing and Information Technology

Spring 2020

CSIT121/821 — Programming Fundamentals

Final Project (40 marks)

Due Time and Date:

Due by Sunday 29th November 2020 10 pm

General Requirements:

- You should create your programs with good programming style and form using proper blank spaces, indentation and braces to make your code easy to read and understand;
- You should create identifiers with sensible names;
- You should make comments to describe your code segments where they are necessary for readers to understand what your code intends to achieve.
- Logical structures and statements are properly used for specific purposes.
- Read the assignment specification carefully, and make sure that you follow whatever directed in this assignment. In every assignment that you will submit in this subject, you must put the following information in the header of your program:

```
/*-------------------------------------------------------
My name:
My student number:
My course code: CSIT121
My email address:
Assignment number: Final Project
-------------------------------------------------------*/
```

Objectives

This project is the replacement of the final exam. You will be asked to design and complete a java program with the skills and concepts learnt from CSIT121.

Background

Café Food Ordering System

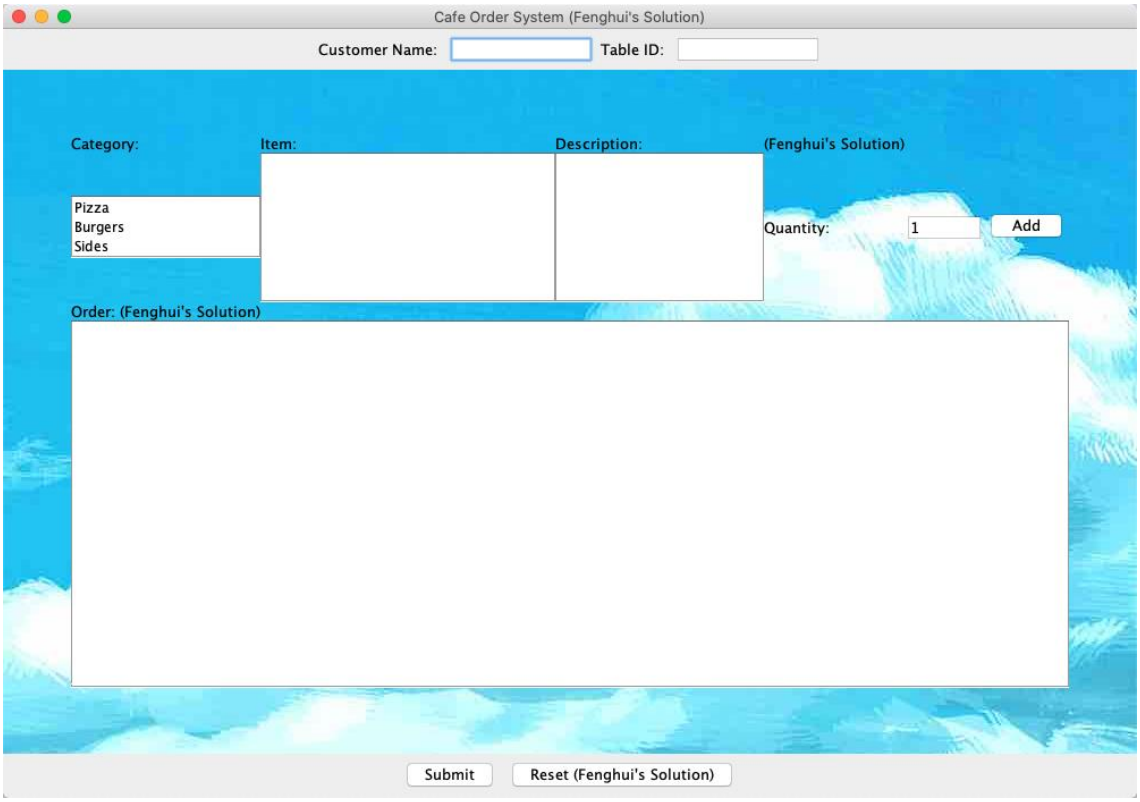
You are required to design and implement a simple Café Food Ordering System based on the above menu. The detailed requirements are as follows:

- The system shall be design by using the object-oriented programming concepts introduced in CSIT121 and implement by using the Java programming language.
- The OOP principle such as inheritance, polymorphism, abstract class/method, interface shall be employed in the development of the system.
- The system shall have a GUI. Customers can input name and the table number as the identification on the top of the GUI. The GUI shall be able to show all the information from the above menu, such as the type, the name, the price, and the extra information of each dish.
- Customers can add any dish to the order. The system shall display the price and quantity of each dish, and automatically update the total price based on the size and the quantity of all dishes.
- Customers can reset their orders any time before they submit it. When the order is reset, the cart will be empty and the total price will be reset to zero.
- Once customers submit the order, the order will be saved as a text file (named as “customerName_tableID.txt”. The text file shall clearly show all the details of the order, including the customer name, table ID, each dish name, size and quantity and price, and the total order price with a reasonable format.
- The system shall able to check the validation of user’s input and order, and ask the user to reinput invalid information.

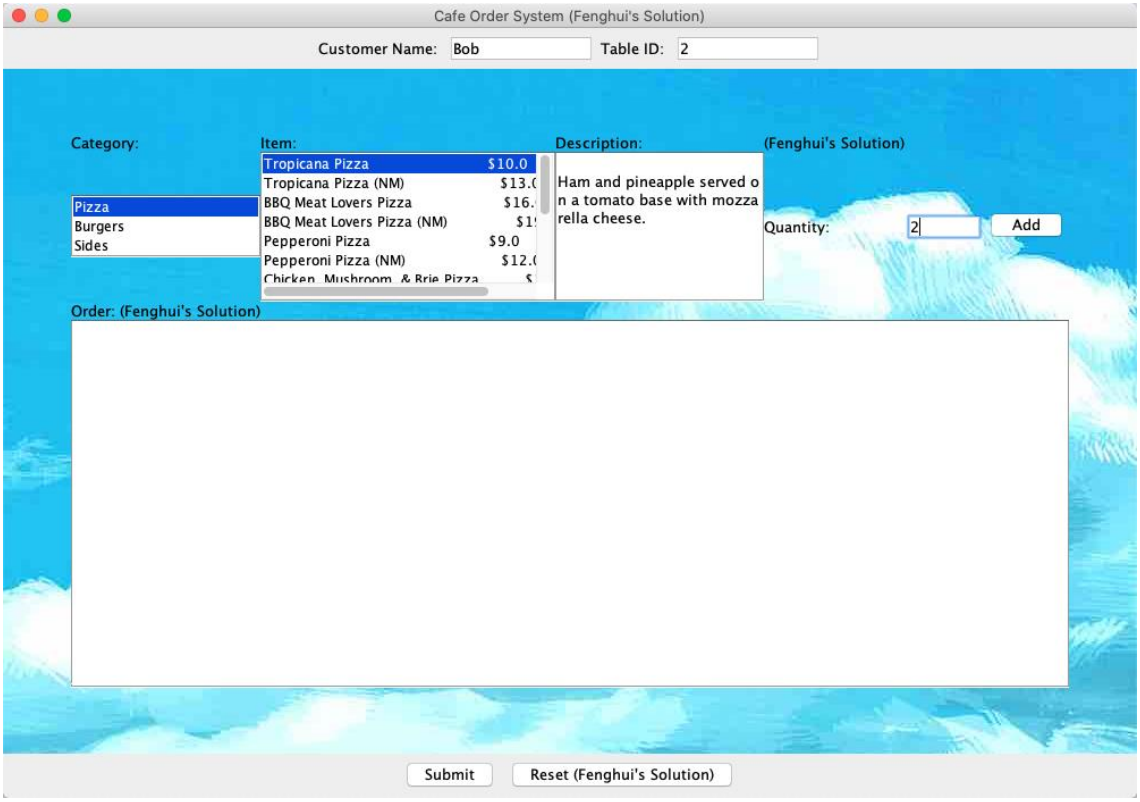
Sample Solution:

The sample solution demonstrates the procedure of using the system to complete an order. Your graphic user interface can have different design from the sample solution. (Please remove the words “Fenghui’s Solution” in your program.

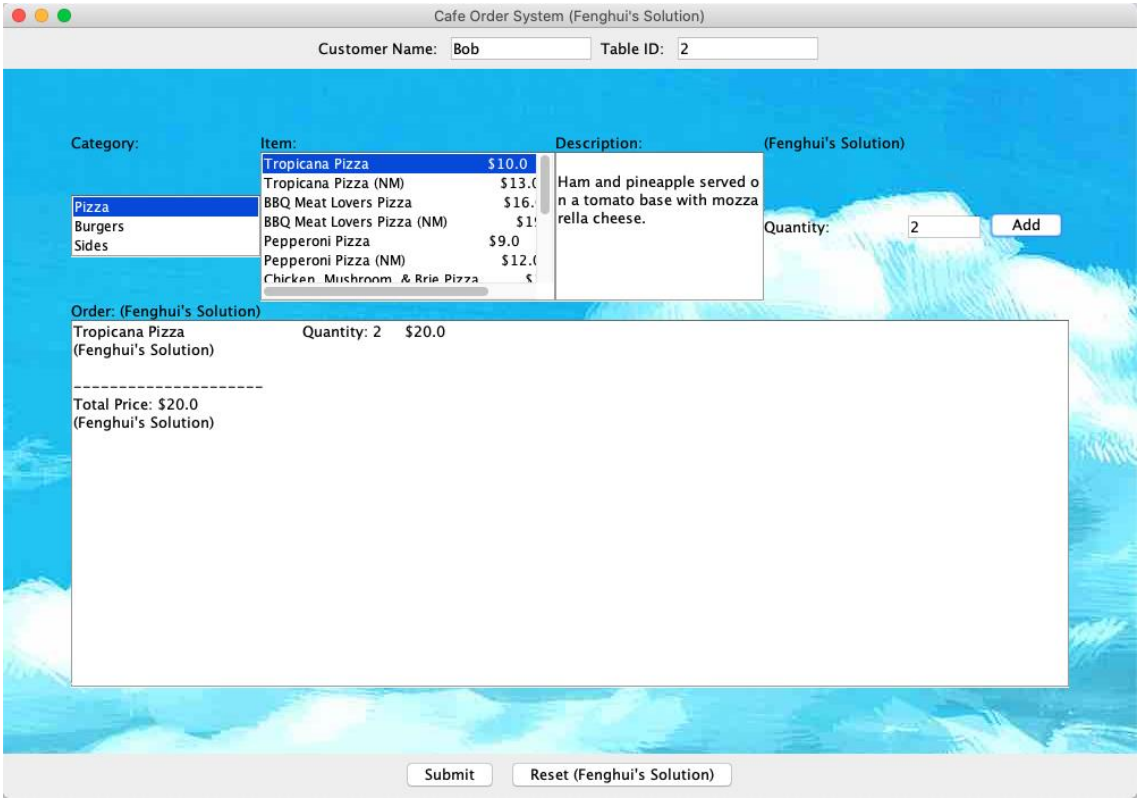
1. Graphic User Interface



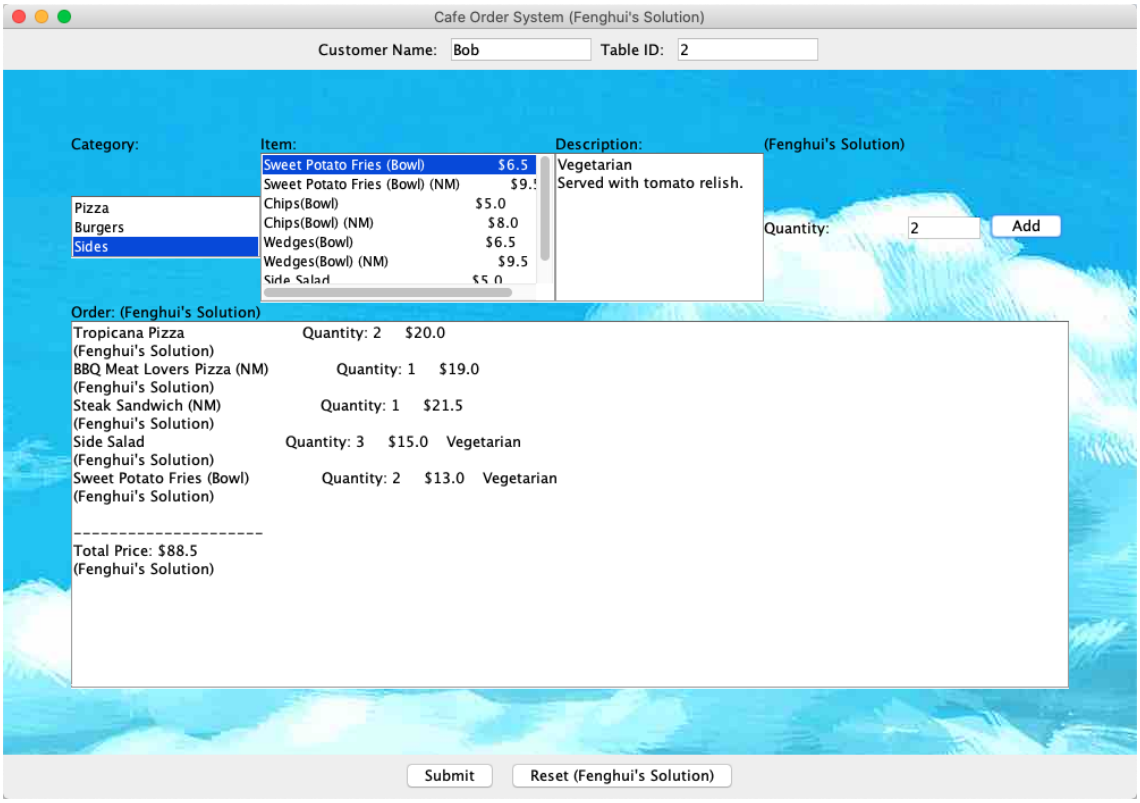
2. Select the Category, Item and Quantity.



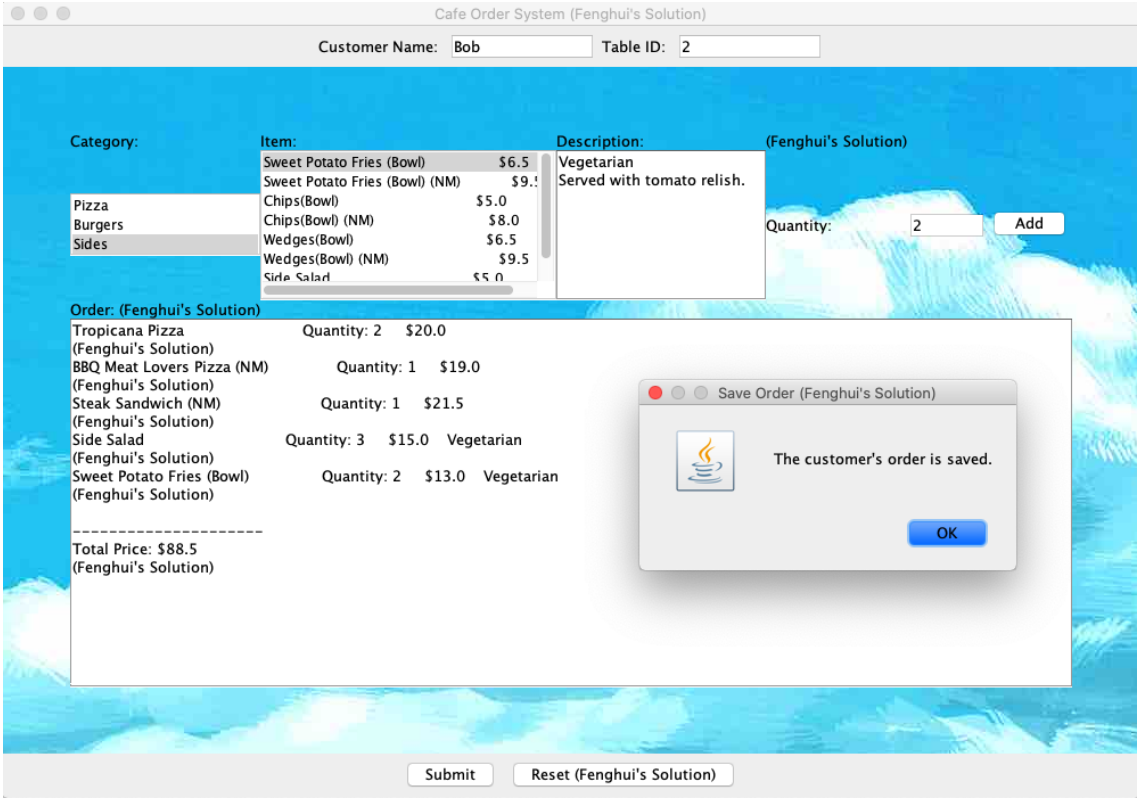
3. Add the first dish

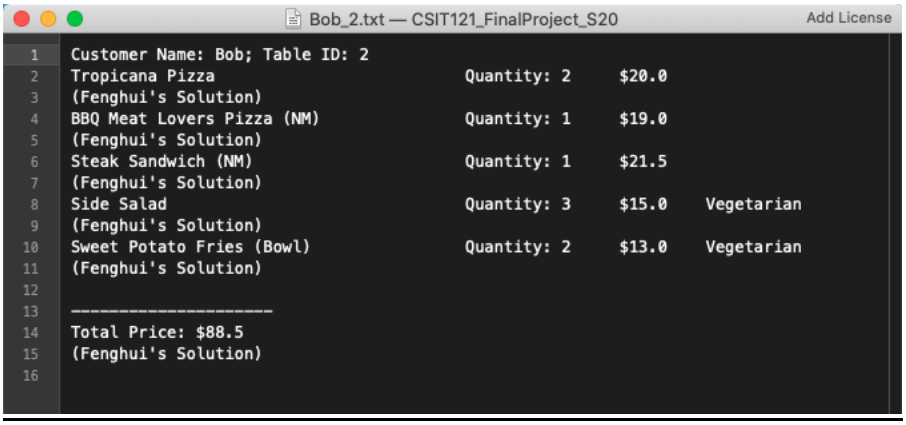


4. Add more dishes

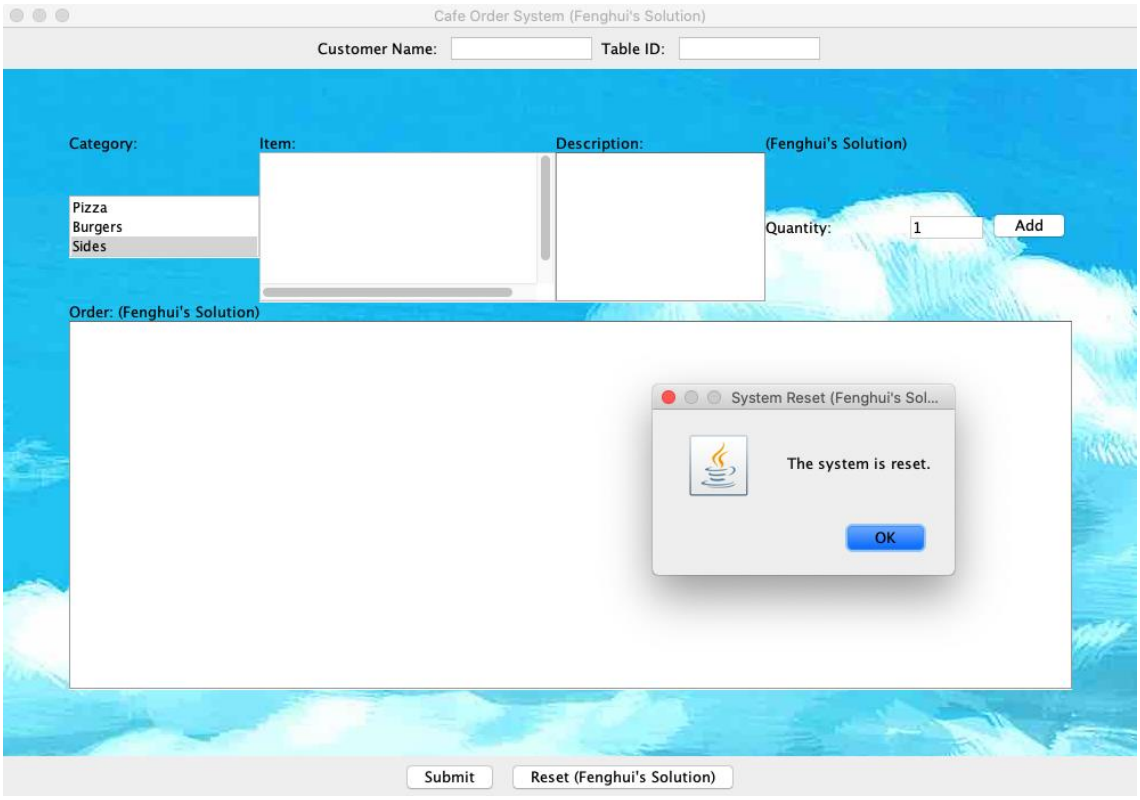


5. Submit the order and save the order as a txt file.





6. Reset the order



Tasks (40 marks):

This project contains several tasks:

1. UML class diagram: You shall draw a UML class diagram to illustrate the class design of your system. In the UML class diagram, you shall clearly specify the class name, fields, methods, associations and multiplicities. (**10 marks**)
2. Graphic User Interface (GUI): You shall design a GUI. The GUI design shall be compact and informative. All information included in the above menu shall be able to display within the GUI. Customers can use the GUI to make the selections and complete the order. (**10 marks**)
3. Coding: You shall implement the classes and GUI based on the above designs and make sure all the requirements are met. The program shall able to handle exceptions with the try…catch block and avoid the runtime errors. (**15 marks**)
4. Testing: You shall compile and test your system with same cases and procedure in the example to make sure the system is workable and tolerant to possible exceptions. (**5 marks**)

Submission:

- Please submit your solution to Moodle (Final Project). Email submission is not accepted.
- Please submit an **individual PDF document (CafeOrderSystem.pdf)**, **a Java program (CafeOrderSystem.java).**
- The **CafeOrderSystem.java** shall **contain all classes** of the system.
- The PDF document shall contain 1) the UML class diagram; 2) the GUI design and explanations; 3) the snapshots clearly shows the compilation and the execution of your program with the testing cases (including all the exception handling); 4) the content of the txt order file.
- Please DO NOT include the java code in the PDF document.
- Turnitin will be used for the plagiarism checking.