The problem analysis and requirements document

ZHANG Yubo 19079784d

1. **Summary of the ​Ordering System functionalities and design requirements​ document in point-by-point format**

**1.1 Functional Requirements 1st: The staff will enter the order of the customers according to the following item codes**

Solution: My program use 4 string-arrays for A, B, C S items codes and another int-array named *count[16]* to record all items’ numbers. There is a string named *order* for users to select one item per time. After checking whether *order* is valid, my program use multiple if-else to match *order* with *count[i]* (my program don’t use switch because switch case can’t contain a string-variable)

**1.2 Functional Requirements 2nd-1: If some of the ordered items can be packaged as a set meal, the System will automatically package these items as a set meal.**

Solution: My program wrote a function named *package()* to package a set meal by the most cost-effective priority. If some items have been packaged, this function can change the items’ numbers in *count[ ]* and deduct the difference price of price. Finally, this function returns the new price.

**1.3 Functional Requirements 2nd-2&3rd: A customer can order items from each menu. The ​Ordering System (OS) then adds the ordered items and reports the sum. For any order that is over $100, there is a 5% discount for the order.**

Solution: As mention before, *count[ ]*and price have recorded each item’s number and total price. After named *package()*, price has been assigned a new value, and then price = price\*0.95 if price>100. Finally, my program call a function named *print\_order()* to print the order of ois (order identified string) that contains each ordered items’ name, number, unit price and total price of the order

**1.4 Functional Requirements 4th: When the receptionist/employee enters the wrong codes, the Ordering System (OS) will reject the order and request a re-submission.**

Solution: All command code in my program are integers. The “check input”part will check whether the input code is a acceptable integer. If not, it will print a wrong message and then go to the “tag”before the “input part”then prompt user to input again.

**1.5 Required Execution Flow of the Ordering System 1st System welcome message**

Solution: There is a function named *init()* in my program. It can print a welcome message in the format of a character drawing and then reset all elements in *count[ ]* with 0.

**1.6 Required Execution Flow of the Ordering System 2nd: ask the user to input an Order Identification String (OIS).**

Solution: There is a string variable named *ois* to record user input Order Identification String. At the end, it will be printed with the order.

**1.7 Other Required Execution Flows of the OS have been mentioned in the Function Requirements’ part.**

1. Section of additional requirements that I think the user will need
   1. **This program should have an option for user to edit the ordered items. (Solved)**

Solution: My program build an additional function named *edit\_order()* that provide users an option to edit ordered items and prevent other not ordered items from being edited. This function can also allow users to edit multiple items per time and the function will stop if all items have been deleted. As mentioned before, after users selected an item, the program asks users to choose from 3 options which contains the edit option.

* 1. **This program should have a function to specify the combination of S5 because S5 has 4 combination method:**

**Coffee+Pie, Coffee+Quiche, Tea+Pie and Tea+Quiche. (Solved)**

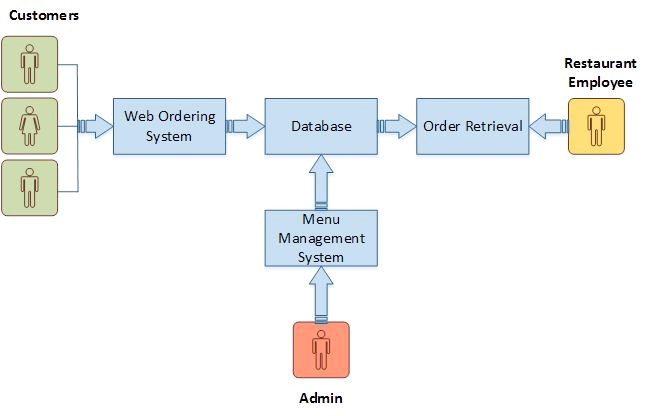
Solution: My program build an additional function named *combination\_S5()*. This function prompt users to choose which drink+food and return a character number to identify which combination. For multiple S5, there is a string named *S5* to record each combination of S5 and finally print the combinations according to this string

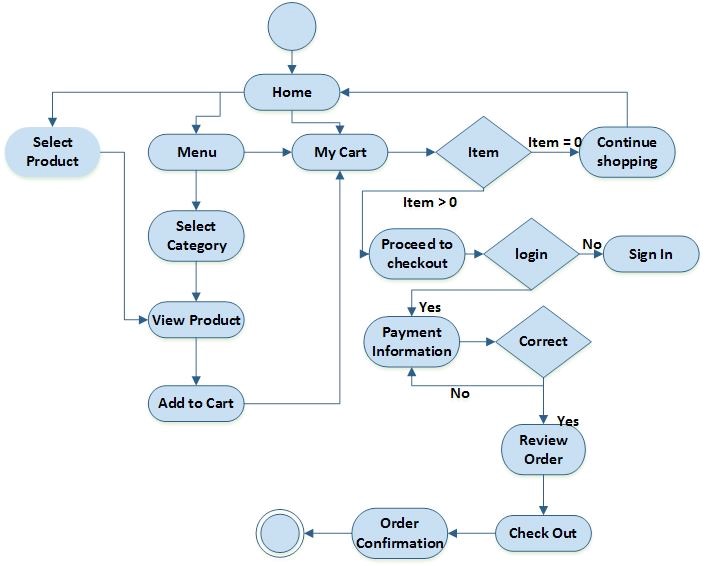
* 1. **This program should use class to store each customer’s order which is easier for users to check history order and daily order statistic data. (Unsolved)**

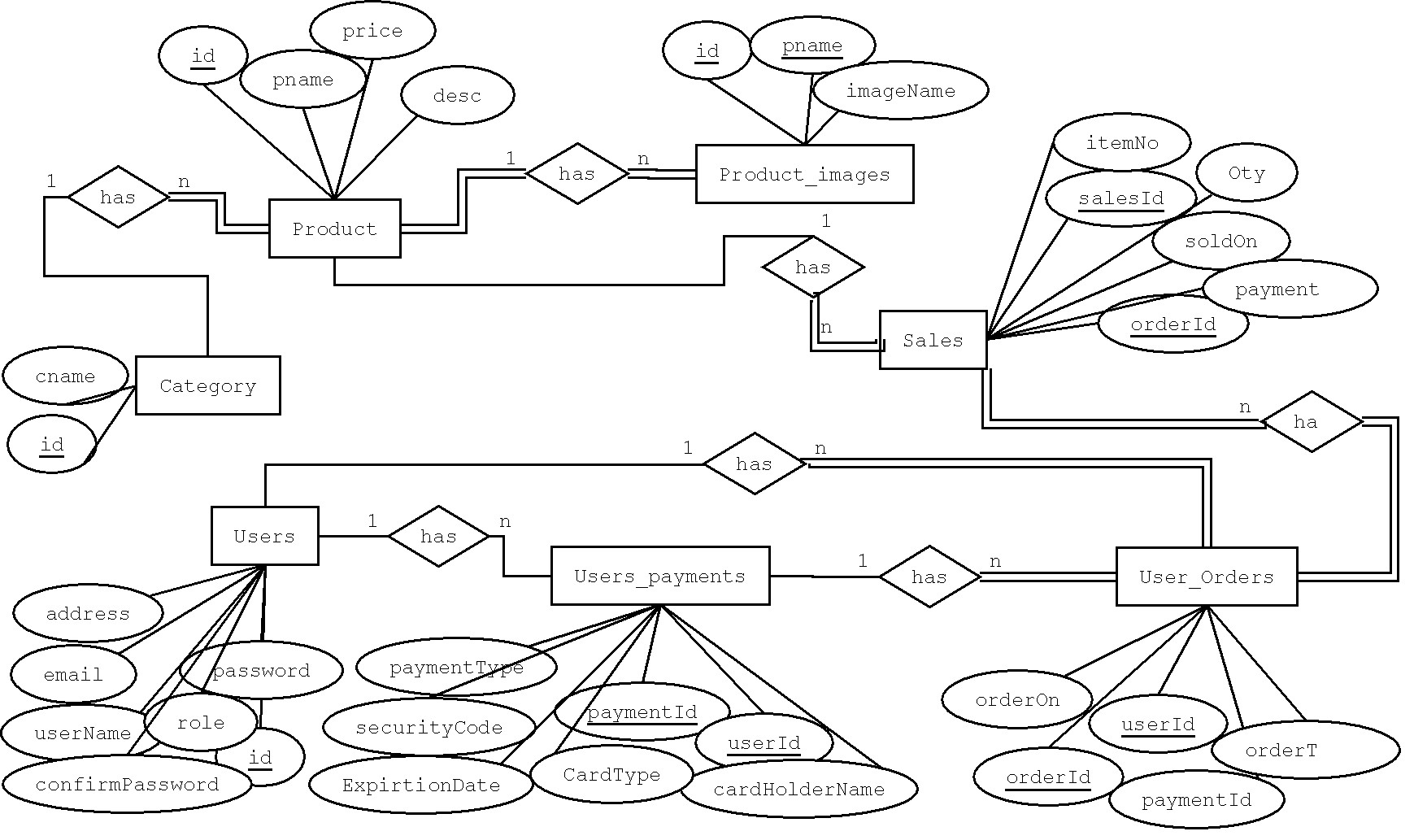
Solution: I have tried use class to store customers’ orders with the ois as class variable’s name. But it’s impossible for c++ to create a new variable name after another variable’s value. Someone says we can use “map”to imitate this process. But it’s also need a defined array contains known oiss which isn’t suitable for this assignment that need users input unknown ois per time. I also tried use multiple arrays to store customers’ orders, but the initial of array need specify the number of customers which is unknown. So, from what I have learnt before, I think maybe other language like python is more suitable to solve this problem.

By the way, I also tried use class to store menu, but I find that S is different from A, B and C. And class isn’t convenient to iterate, so finally I changed to use arrays.

1. Some useful attached charts for Ordering System that support what has been mention above.







Charts Reference: [https://www.google.com/url?sa=i&url=https%3A%2F%2F](https://www.google.com/url?sa=i&url=https%3A%2F%2Fscholarworks.gvsu.edu%2Fcgi%2Fviewcontent.cgi%3Farticle%3D1222%26context%3Dcistechlib&psig=AOvVaw2GGv3Xa7iba3RjuNc_nlFh&ust=1589098089531000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCIjZxOuppukCFQAAAAAdAAAAABAQ)