



**SCHOOL OF COMPUTER SCIENCE
SEMESTER 1 SESSION 2020/2021
CPT 111 – PRINCIPLES OF PROGRAMMING
ASSIGNMENT 2**

Course: CPT 111 – PRINCIPLES OF PROGRAMMING

Title: Food Ordering System

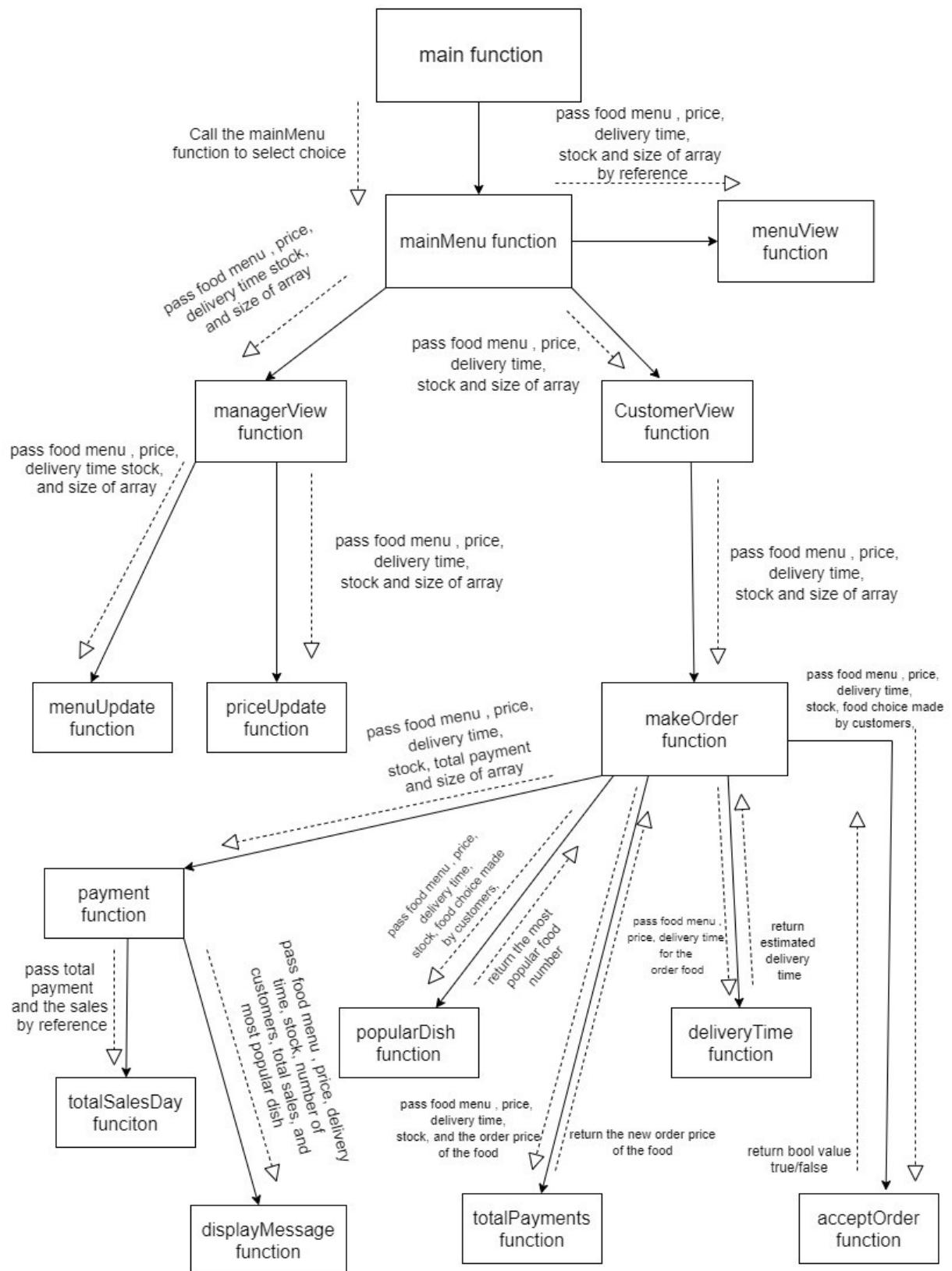
Name: CHEONG TZE YUAN

Matric No: 153779

Group Number: Group B

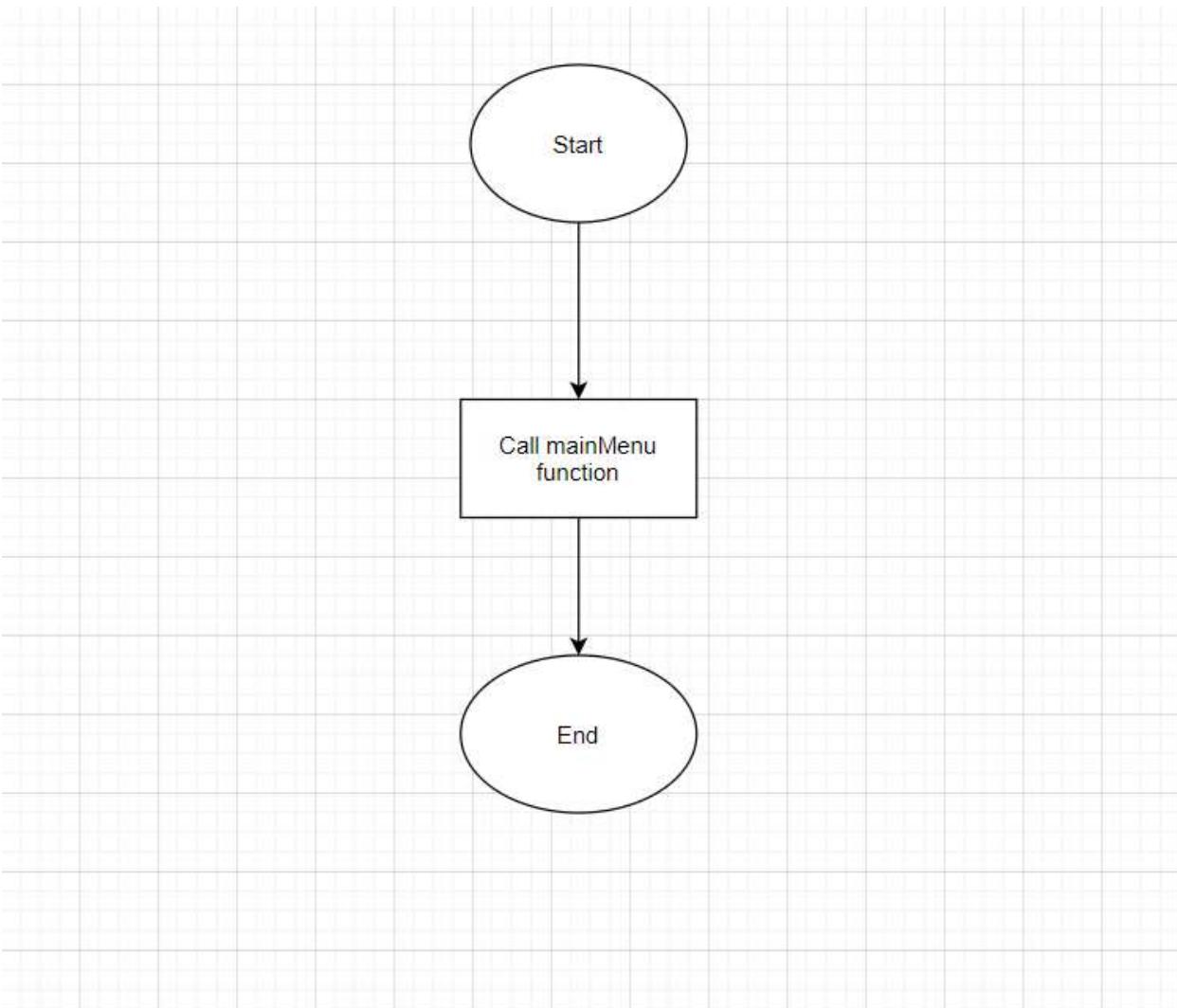
Lecturer: PUAN MAZIANI BINTI SABUDIN

Structure chart



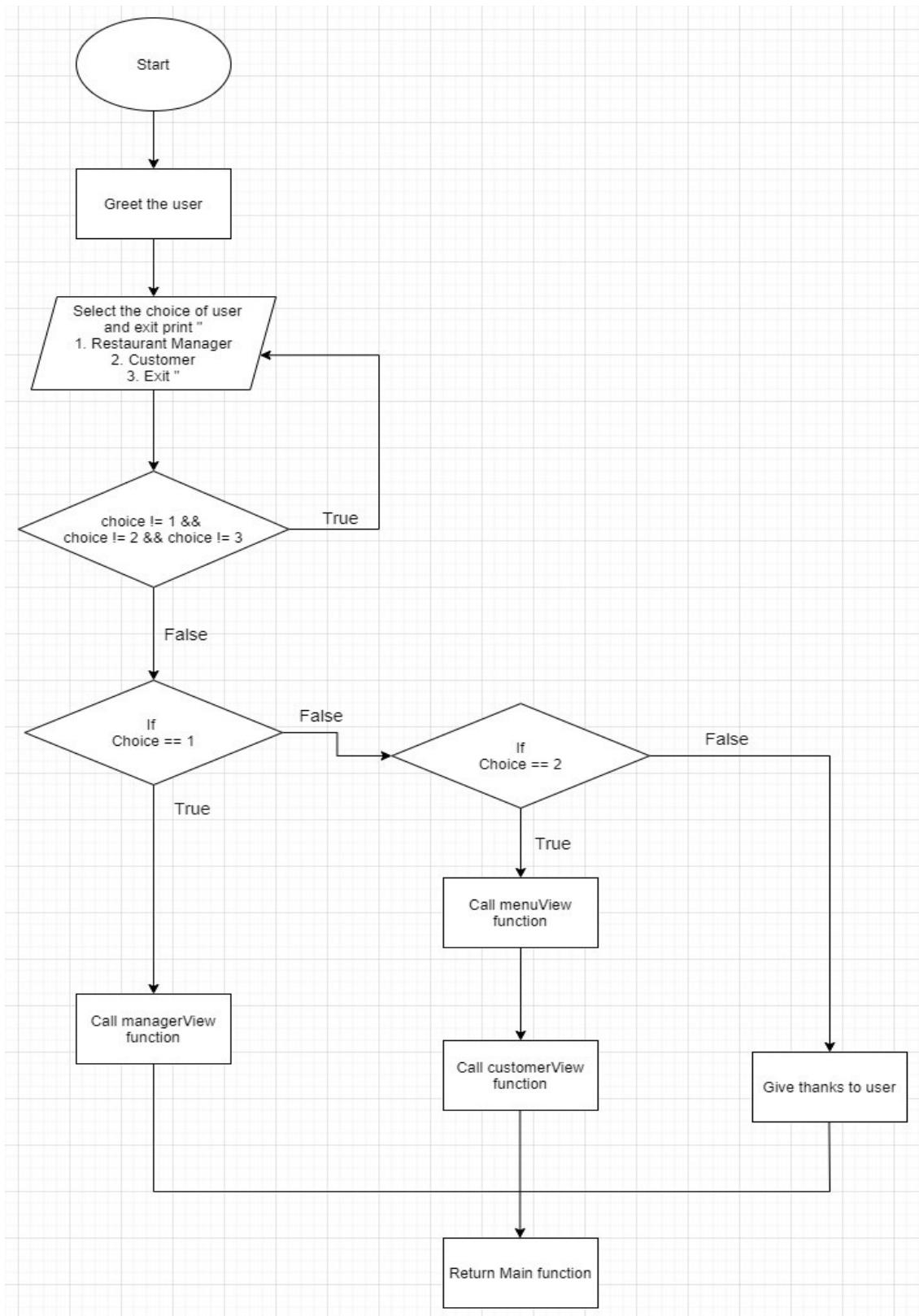
Flowchart

1. Main function

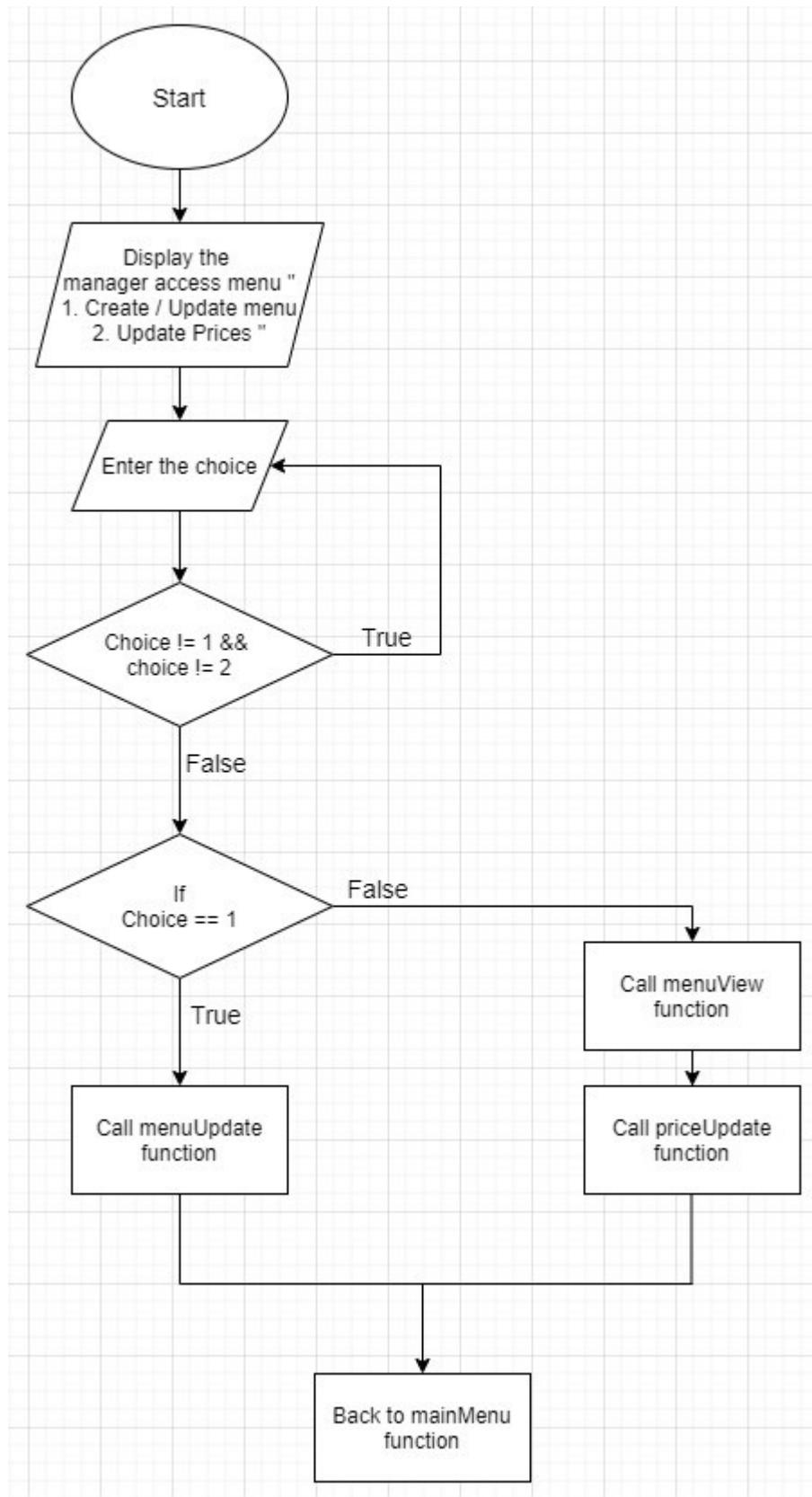


User-defined function

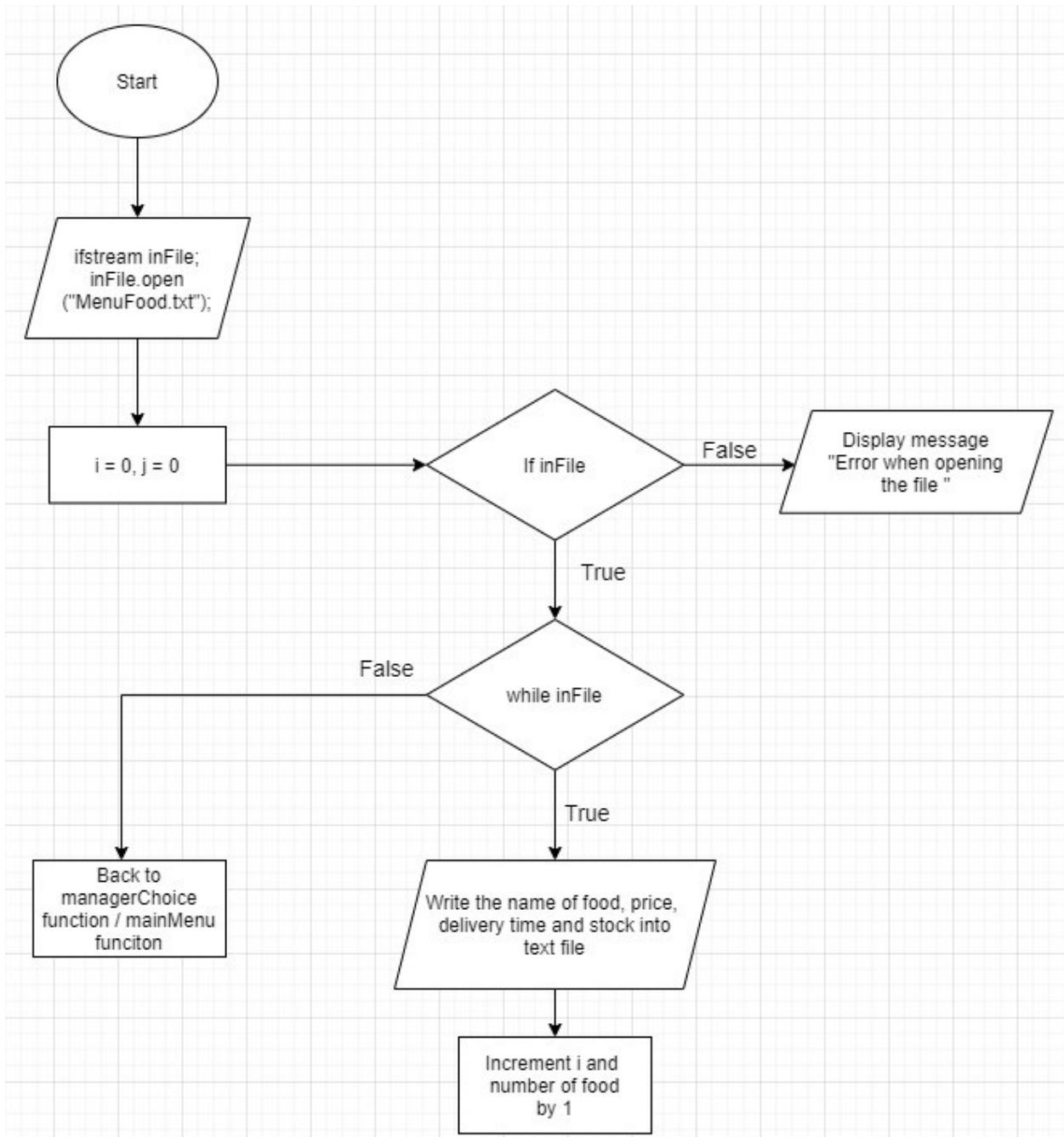
2. mainMenu function



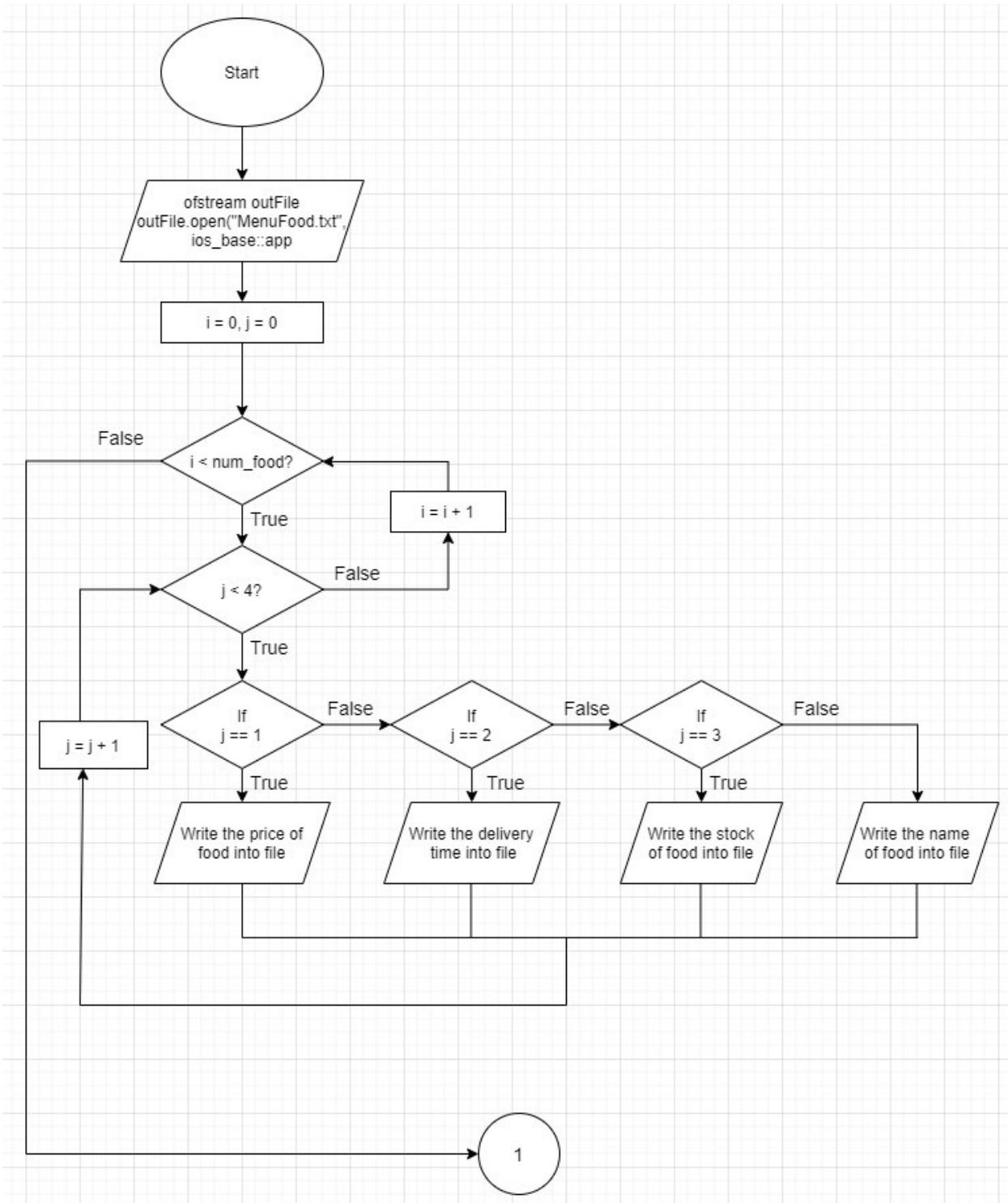
3. managerView function

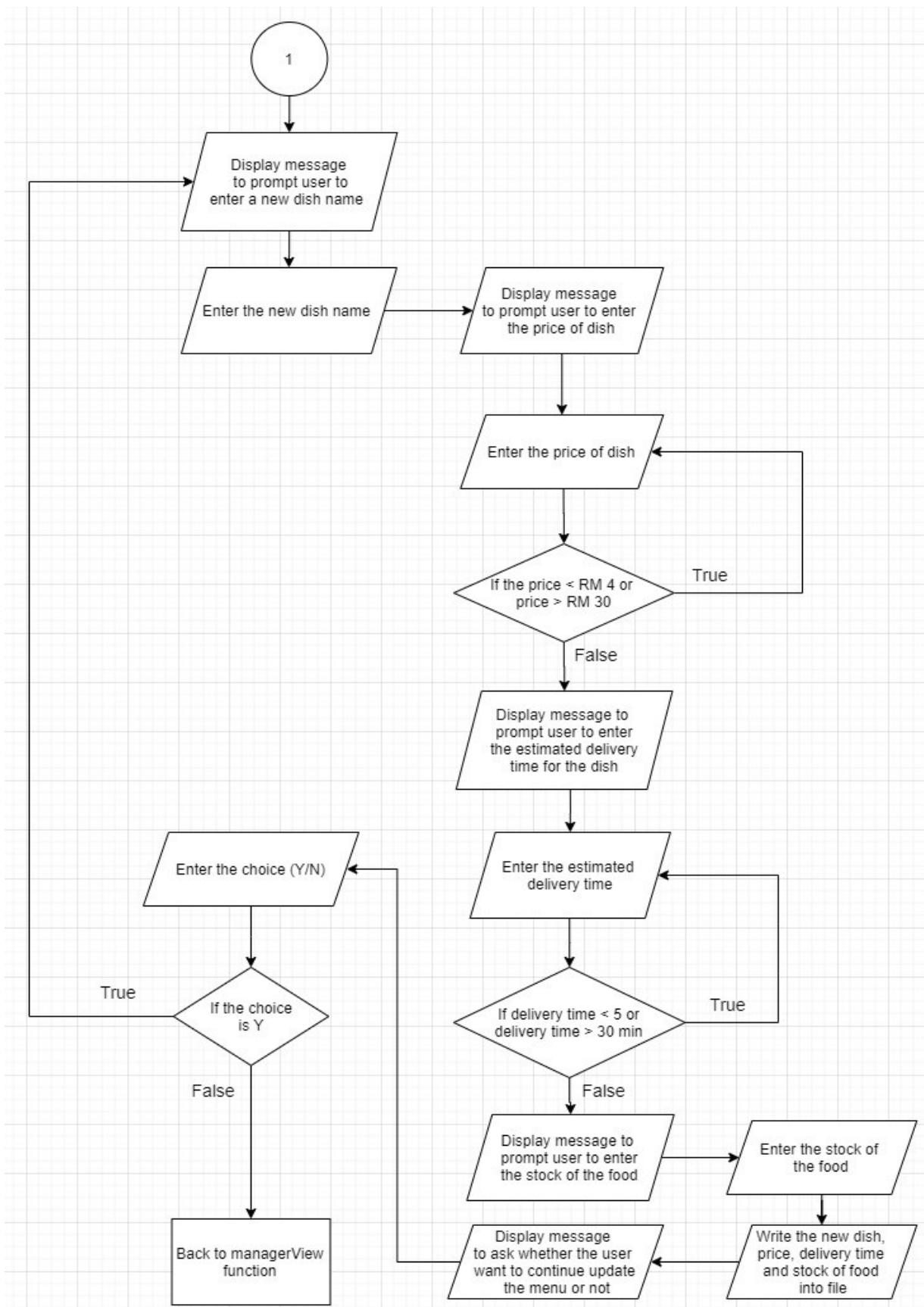


4. menuView function

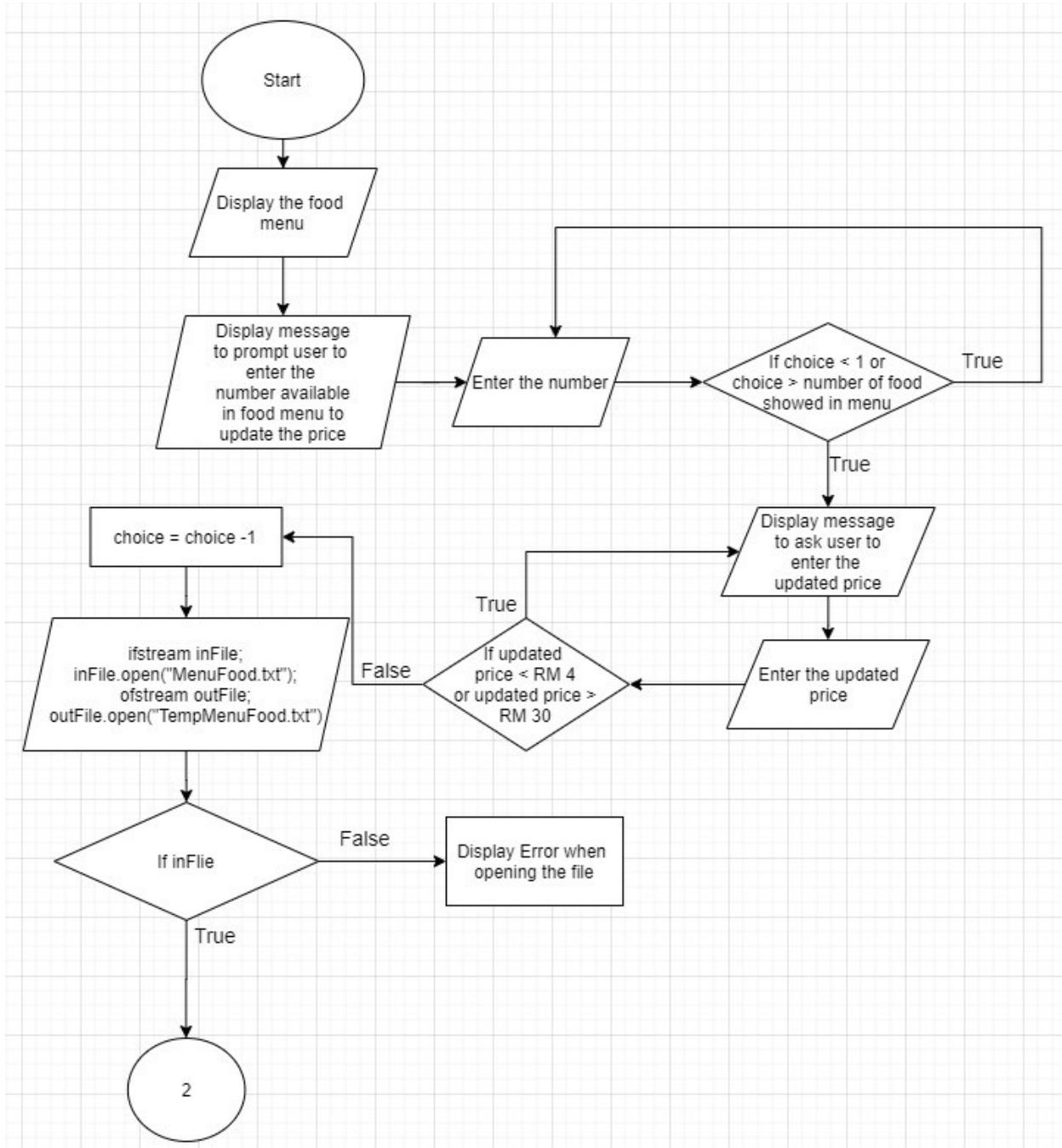


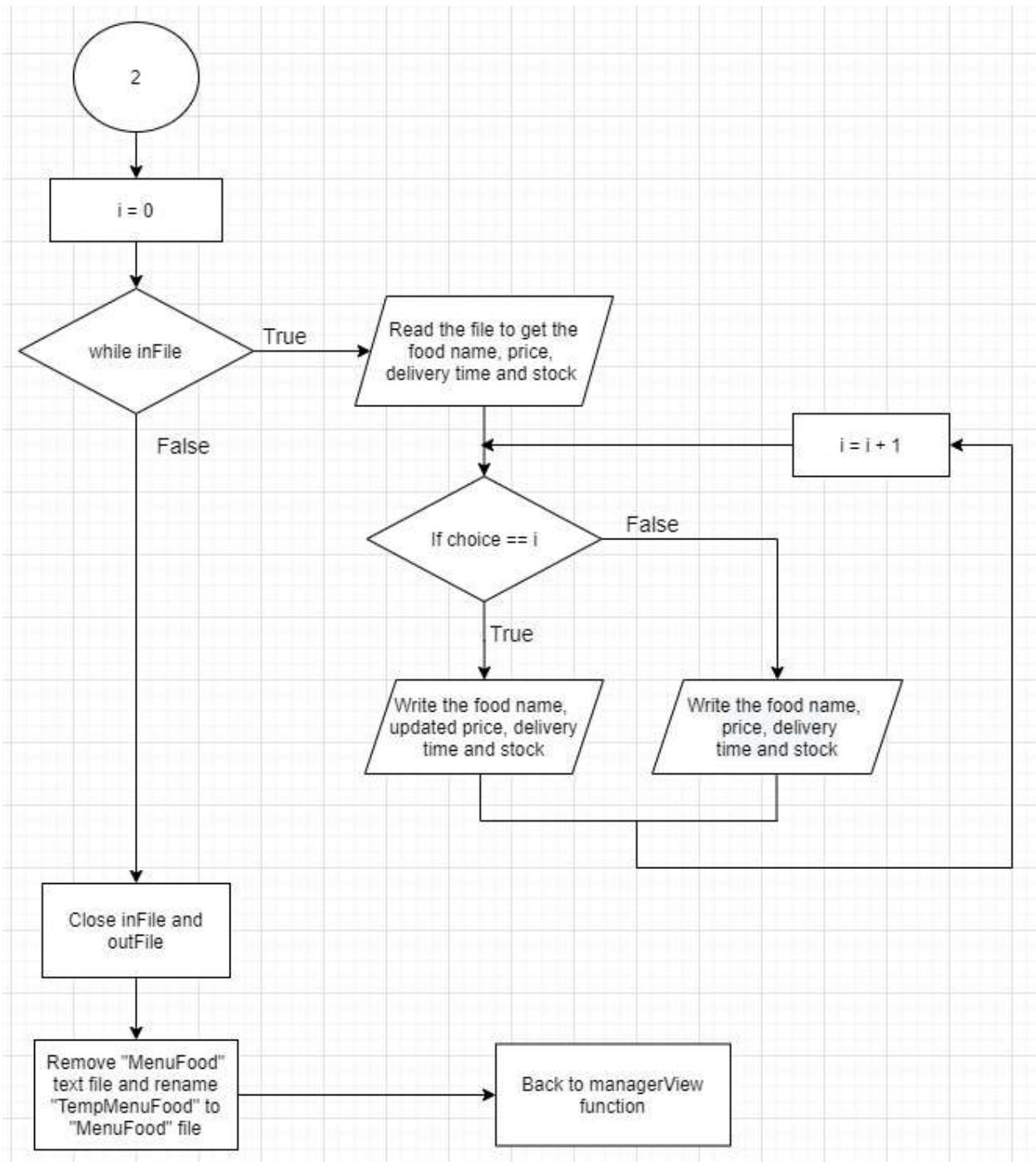
5. menuUpdate function



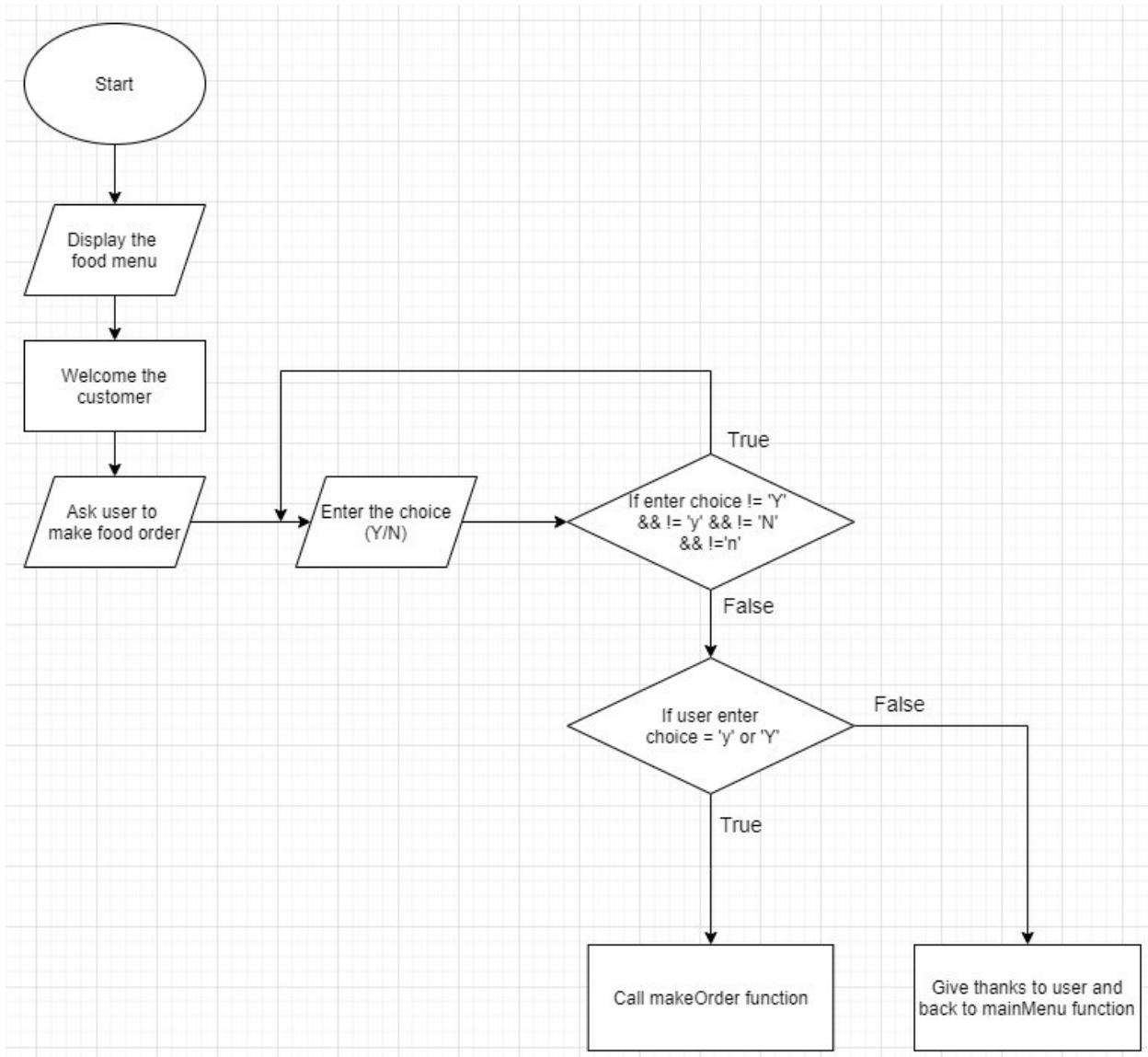


6. priceUpdate function

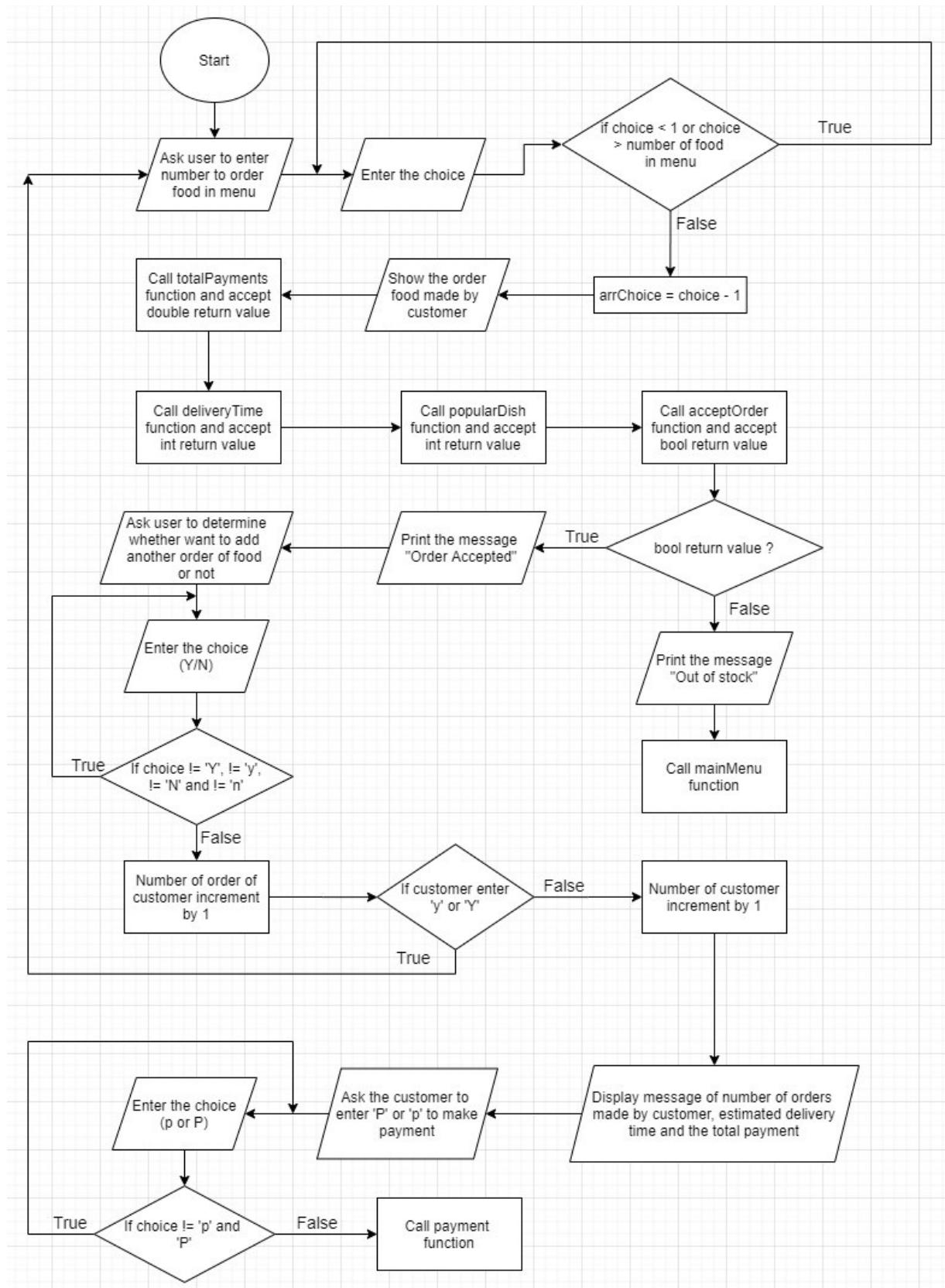




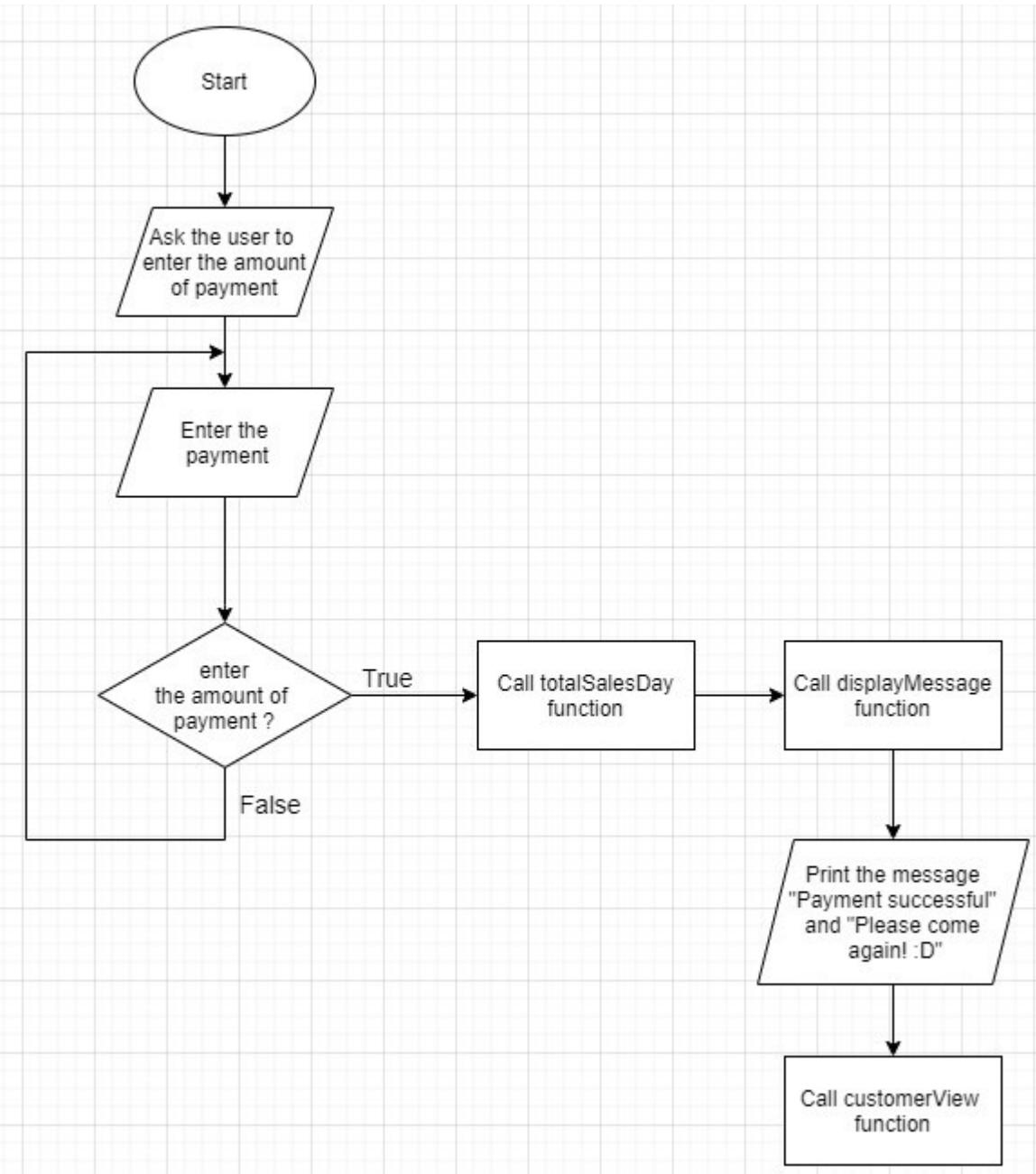
7. customerView function



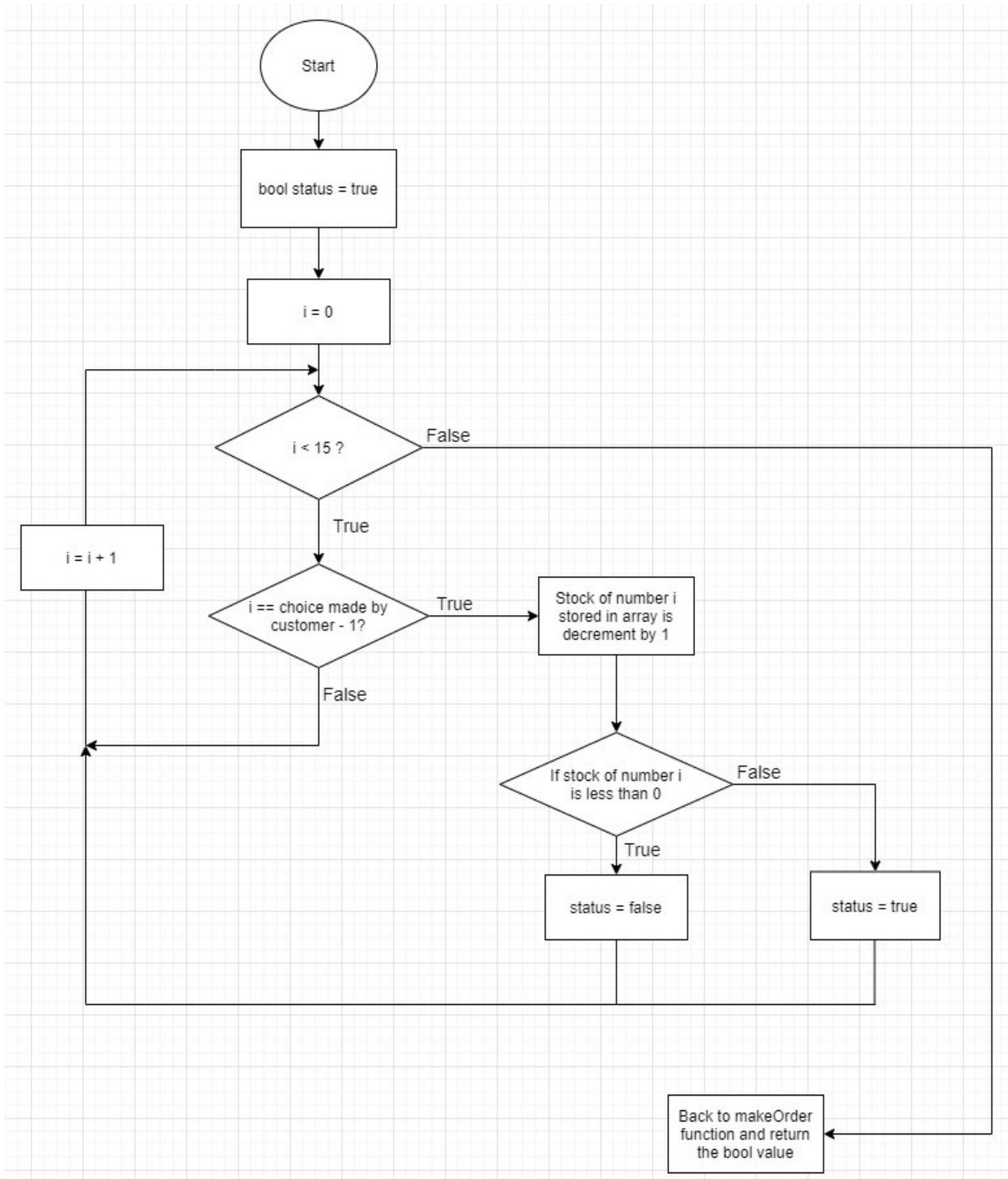
8. makeOrder function



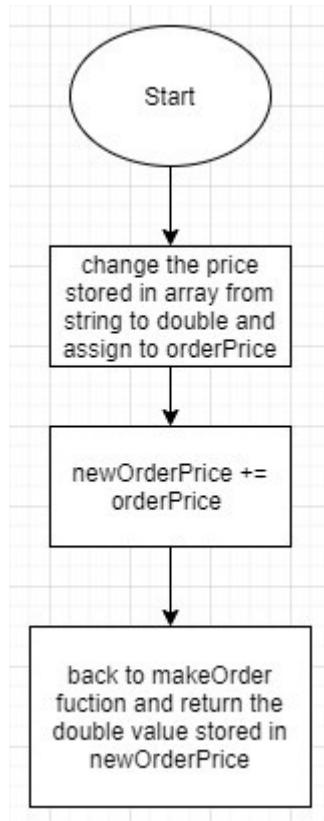
9. payment function



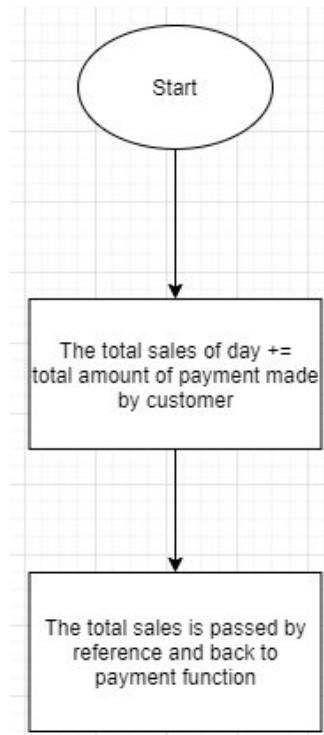
10. acceptOrder function



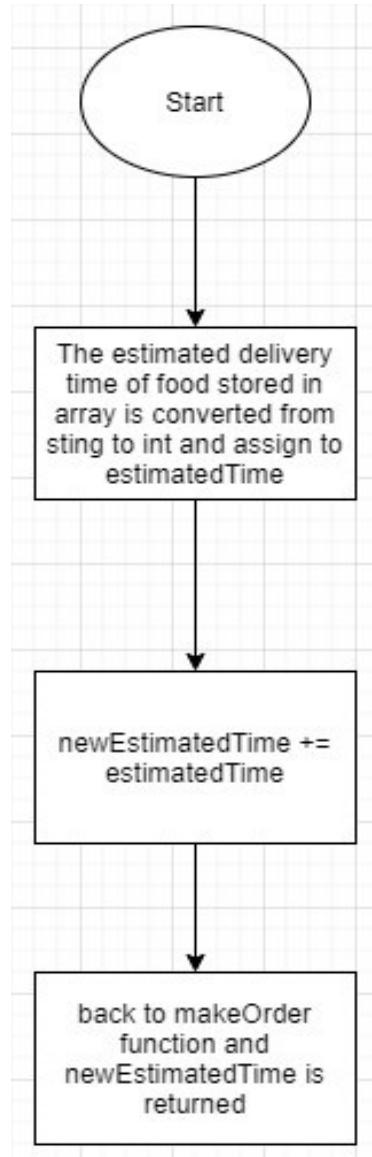
11. totalPayments function



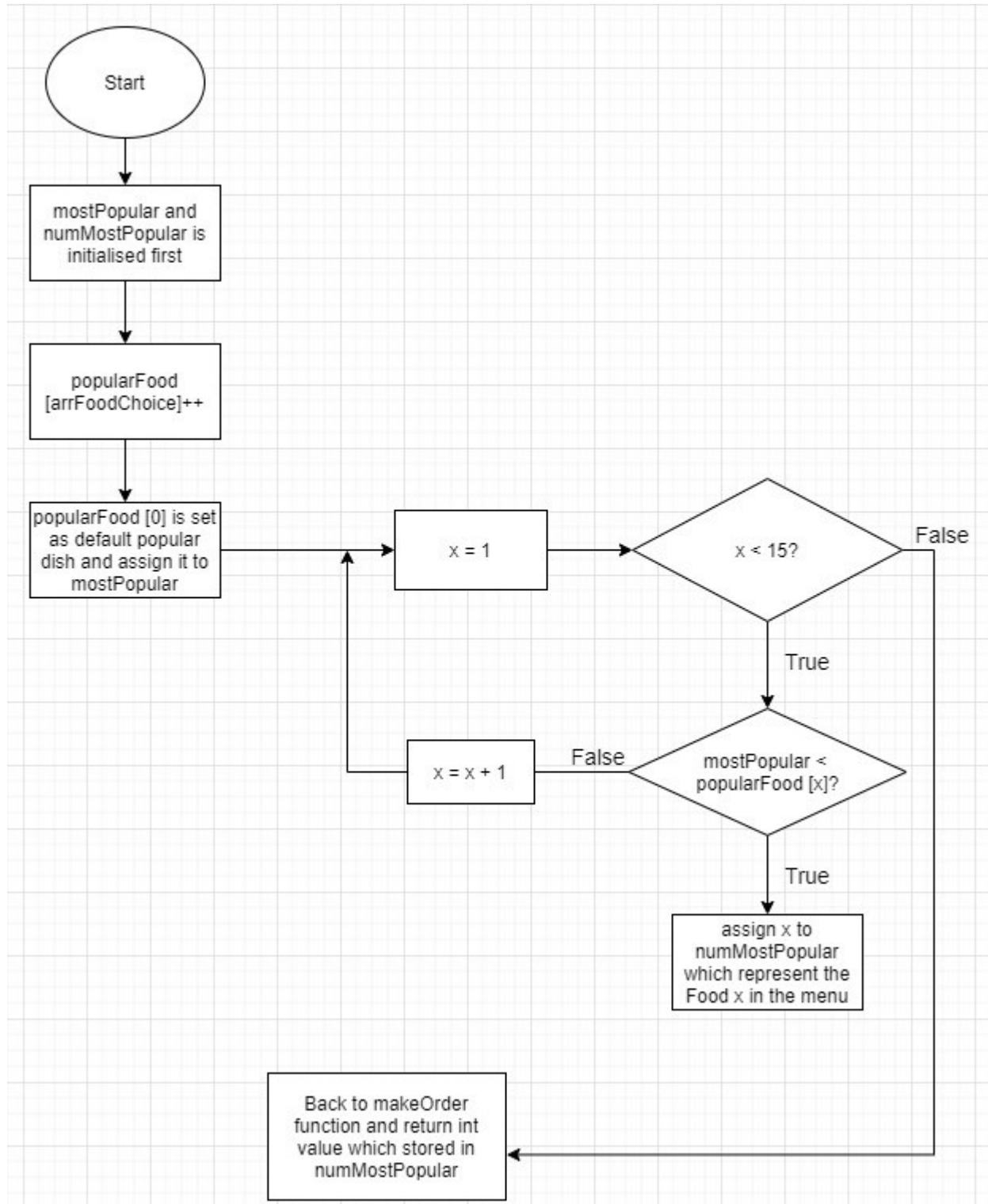
12. totalSalesDay function



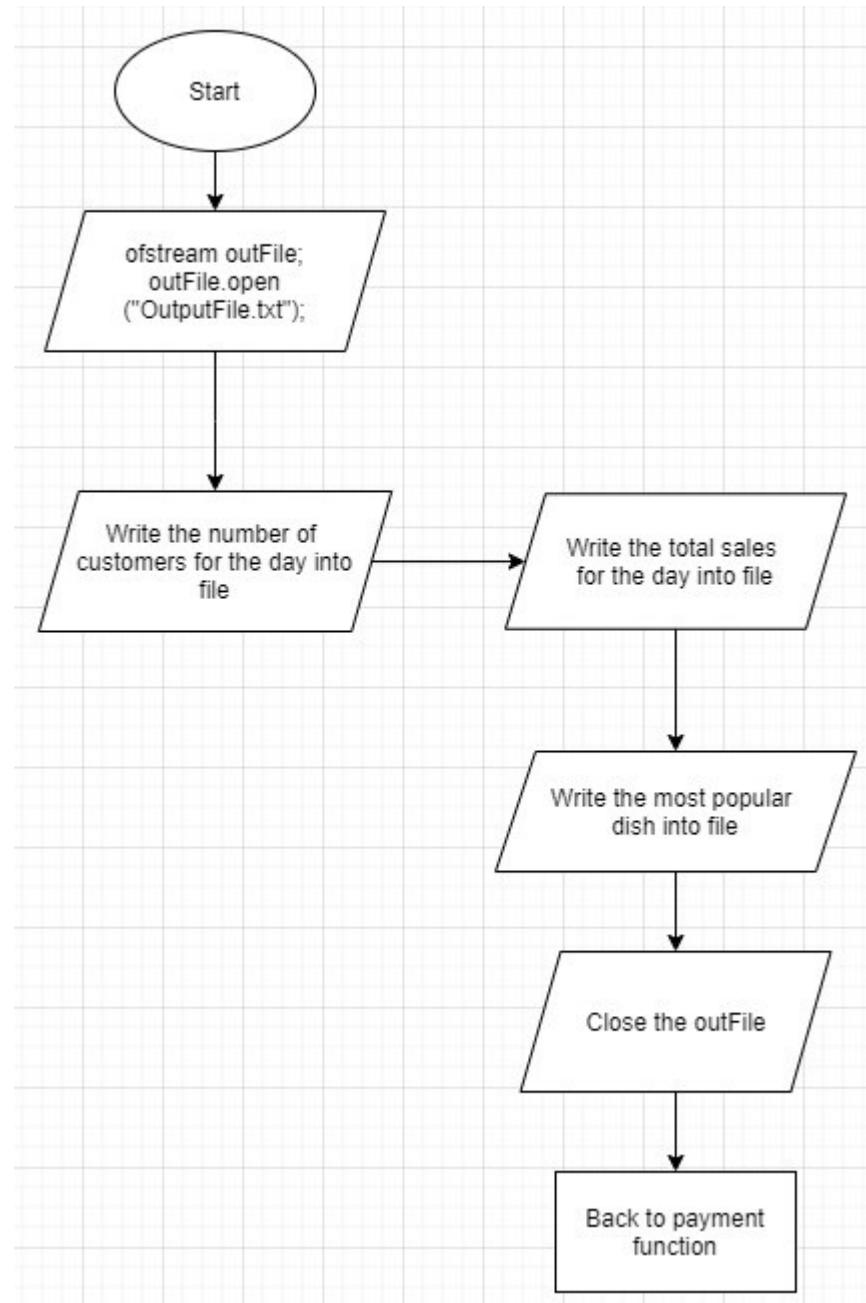
13. deliveryTime function



14. popularDish function



15. displayMessage function



C++ Program listings

```
01. #include <iostream>
02. #include <fstream>
03. #include <string>
04. #include <iomanip>
05. #include <cstdlib>
06. using namespace std;
07.
08. // Global constants
09. const int FOOD = 15;
10. const int COL = 4;
11.
12. // Function prototype
13. void mainMenu(int[FOOD]);
14. void menuView(string[ ][COL], int&);
15.
16. void managerView(string[ ][COL], int);
17. void menuUpdate(string[ ][COL], int);
18. void priceUpdate(string[ ][COL], int);
19. bool acceptOrder(string[ ][COL], int, int, int[FOOD]);
20. double totalPayments(string[ ][COL], int, double, double &);
21. int deliveryTime(string[ ][COL], int, int, int&);
22. void totalSalesDay(double, double&);
23. int popularDish(string[ ][COL], int, int, int[FOOD]);
24.
25. void customerView(string[ ][COL], int, int[FOOD]);
26. void makeOrder(string[ ][COL], int, int[FOOD]);
27. void payment(string[ ][COL], int, double, int[FOOD], int, int);
28. void displayMessage (string[ ][COL], int, double, int);
29.
30. // Main function
31. int main(){
32.
33.     int popularFood[FOOD];
34.
35.     for (int i = 0; i < FOOD; i++){
36.
37.         popularFood[i] = 0;
38.
39.     }
40.
41.     mainMenu(popularFood);
42.
43.     return 0;
44. }
```



```
99. // **** Manager Access ****
100. // **** Manager Access ****
101. // **** Manager Access ****
102.
103. void managerView(string nameFoodPrice[][COL], int num_food){
104.
105.     int managerChoice;
106.
107.     cout << "\t\t Manager Access " << endl << endl;
108.     cout << "1. Create / Update Menu " << endl;
109.     cout << "2. Update Prices " << endl << endl;
110.
111.     cout << "Please select a number for further details: ";
112.     cin >> managerChoice;
113.
114.     while(managerChoice != 1 && managerChoice !=2){
115.         cout << endl;
116.         cout << "Invalid number! Please try again! " << endl;
117.         cin >> managerChoice;
118.     }
119.
120.     if(managerChoice == 1){
121.         cin.ignore();
122.         cout << endl << endl;
123.         system("pause");
124.         system("CLS");
125.         menuUpdate(nameFoodPrice, num_food);
126.     }
127.     else{
128.         cout << endl << endl;
129.         system("pause");
130.         system("CLS");
131.         menuView(nameFoodPrice, num_food);
132.         priceUpdate(nameFoodPrice, num_food);
133.     }
134.
135.
136. }
```

```
137. // *****
138. // ***** Food Menu Access *****
139. // *****
140.
141.
142. void menuView(string nameFoodPrice[][COL], int &num_food){
143.
144.     ifstream inFile;
145.     inFile.open("MenuFood.txt");
146.
147.     if (inFile){
148.
149.         int i = 0;
150.
151.         while (inFile){
152.
153.             int j = 0;
154.
155.             getline(inFile, nameFoodPrice[i][j], '\t');
156.             getline(inFile, nameFoodPrice[i][j+1], '\t');
157.             getline(inFile, nameFoodPrice[i][j+2], ' ');
158.             getline(inFile, nameFoodPrice[i][j+3]);
159.             i++;
160.             num_food++;
161.
162.             inFile.ignore();
163.         }
164.
165.     }
166.     else{
167.         cout << "Error when opening the file " << endl;
168.     }
169.
170. }
171.
```

```
172. // ****
173. // Food Menu Updation
174. // ****
175.
176. void menuUpdate(string nameFoodPrice[][COL], int num_food){
177.
178.     int timeDelivery, stockFood;
179.     char choice;
180.     string newFoodName;
181.     double newFoodPrice;
182.
183.     ofstream outFile;
184.     outFile.open("MenuFood.txt", ios_base::app);
185.
186.     for (int i = 0; i < num_food; i++){
187.
188.         for (int j = 0; j < COL; j++){
189.
190.             if(j == 1){
191.                 outFile << '\t';
192.                 outFile << nameFoodPrice[i][j];
193.             }
194.
195.             else if(j == 2){
196.                 outFile << '\t';
197.                 outFile << nameFoodPrice[i][j];
198.             }
199.
200.             else if (j == 3){
201.                 outFile << ' ';
202.                 outFile << nameFoodPrice[i][j];
203.             }
204.             else{
205.                 outFile << ' ' << nameFoodPrice[i][j];
206.             }
207.         }
208.         outFile << '\n';
209.     }
210.
211.     do{
212.         cout << endl << "Please enter a new food name: " ;
213.         getline(cin, newFoodName);
214.
215.         cout << "Please enter the price of the food: RM ";
216.         cin >> newFoodPrice;
217.         cout << endl;
```

```
218.  
219.     while (newFoodPrice < 4 || newFoodPrice > 30){  
220.  
221.         cout << "The price of the food must be set in the range from "  
222.         << "RM 4 to RM 30 " << endl;  
223.         cout << "Please set the price of the food again: RM ";  
224.         cin >> newFoodPrice;  
225.         cout << endl;  
226.     }  
227.  
228.     cout << "Please enter the estimated delivery time of the food (in minutes): ";  
229.     cin >> timeDelivery;  
230.  
231.     while (timeDelivery < 5 || timeDelivery > 30){  
232.  
233.         cout << "The delivery time for the food must be set in the range from "  
234.         << "5 minutes to 30 minutes " << endl;  
235.         cout << "Please set the delivery time again: ";  
236.         cin >> timeDelivery;  
237.         cout << endl;  
238.  
239.     }  
240.  
241.     cout << "Please enter the stock of the food: ";  
242.     cin >> stockFood;  
243.     cout << endl;  
244.  
245.     outFile << setprecision(2) << fixed << showpoint;  
246.     outFile << '\n';  
247.     outFile << ' ' << newFoodName << '\t' << newFoodPrice << '\t' << timeDelivery << ' ' << stockFood;  
248.  
249.     cout << "Would you like to add another food? (Y/N) Your choice: ";  
250.     cin >> choice;  
251.  
252.     while(choice != 'Y' && choice != 'y' && choice != 'N' && choice != 'n'){  
253.         cout << "Invalid ! Please try again! " << endl;  
254.         cin >> choice;  
255.     }  
256.  
257.     cin.ignore();  
258.  
259. } while (choice == 'Y' || choice == 'y');  
260.  
261. cout << endl << "Update Successfully! Please check the updated menu " << endl;  
262.  
263. }
```

```

264. // *****
265. //          Food Price Updation
266. // *****
267.
268. void priceUpdate(string nameFoodPrice[][COL], int num_food){
269.
270.     cout << "-----" << endl;
271.     cout<<"\n";
272.     cout<<"   / \\" \n";
273.     cout<<"   / \\" / \\" \\" / \\" \\" | \\" | \\" \n";
274.     cout<<"   / \\" / \\" / \\" | \\" | \\" | \\" \n";
275.     cout<<"   \\" | \\" / \\" > \\" | \\" | \\" \n\n";
276.     cout << "-----" << endl;
277.
278.     for (int x = 0; x < num_food; x++){
279.         cout << "Food " << x + 1 << ": ";
280.
281.         for (int y = 0; y < COL - 2; y++){
282.
283.             cout << nameFoodPrice[x][y] << "\t\t";
284.
285.             if (y == 1){
286.
287.                 if (x < 3){
288.                     cout << "RM " << nameFoodPrice[x][y];
289.                 }
290.
291.                 else{
292.                     cout << "\tRM " << nameFoodPrice[x][y];
293.                 }
294.             }
295.
296.             cout << endl;
297.         }
298.
299.     }
300.
301.     cout << endl;
302.
303.     int updateChoice;
304.     double updatePrice;
305.
306.     cout << "Please enter the number available in order to update the food price: Food ";
307.     cin >> updateChoice;
308.
309.     while (updateChoice < 1 || updateChoice > num_food){
310.
311.         cout << "Invalid number. Please enter again: Food ";
312.         cin >> updateChoice;
313.     }
314.
315.     cout << "Please enter the new price for the dish: RM ";
316.     cin >> updatePrice;
317.

```

```
318.     while (updatePrice < 4 || updatePrice > 30){
319.
320.         cout << "The price of the dish must be updated in the range from "
321.             << "RM 4 to RM 30 " << endl;
322.         cout << "Please update the price of the dish again: RM ";
323.         cin >> updatePrice;
324.         cout << endl;
325.
326.
327.         updateChoice--;
328.
329.         ifstream inFile;
330.         inFile.open("MenuFood.txt");
331.
332.         ofstream outFile;
333.         outFile.open("TempMenuFood.txt");
334.
335.         if (inFile){
336.
337.             int i = 0;
338.
339.             while (inFile){
340.
341.                 int j = 0;
342.
343.                 getline(inFile, nameFoodPrice[i][j], '\t');
344.                 getline(inFile, nameFoodPrice[i][j+1], '\t');
345.                 getline(inFile, nameFoodPrice[i][j+2], ' ');
346.                 getline(inFile, nameFoodPrice[i][j+3]);
347.
348.                 outFile << setprecision(2) << fixed << showpoint;
349.                 if (updateChoice == i){
350.                     outFile << ' ' << nameFoodPrice[i][j] << '\t'
351.                         << updatePrice << '\t' << nameFoodPrice[i][j+2] << ' ' << nameFoodPrice[i][j+3] << '\n';
352.                 }
353.
354.                 else{
355.                     outFile << ' ' << nameFoodPrice[i][j];
356.                     outFile << '\t' << nameFoodPrice[i][j+1] << '\t' << nameFoodPrice[i][j+2]
357.                         << ' ' << nameFoodPrice[i][j+3] << '\n';
358.                 }
359.
360.                 i++;
361.
362.                 inFile.ignore();
363.             }
364.
365.         }
366.
367.         else{
368.             cout << "Error when opening the file..." << endl;
369.         }
370.
371.         inFile.close();
372.         outFile.close();
373.         remove("MenuFood.txt");
374.         rename("TempMenuFood.txt", "MenuFood.txt");
375.
376.         cout << "Update Successfully! Kindly check the updated menu " << endl;
377.     }
```

```
378.  
379. // ****  
380. // Accept Order Function  
381. // ****  
382.  
383. bool acceptOrder(string nameFoodPrice[ ][COL], int num_food, int arrFoodChoice, int stockFood[FOOD]) {  
384.  
385.     bool status = true;  
386.  
387.     for (int i = 0; i < FOOD; i++) {  
388.  
389.         if (i == arrFoodChoice) {  
390.             stockFood[i]--;  
391.  
392.             if (stockFood[i] < 0) {  
393.                 status = false;  
394.             }  
395.  
396.             else {  
397.                 status = true;  
398.             }  
399.  
400.         }  
401.  
402.     }  
403.  
404.     return status;  
405. }  
406.  
407. // ****  
408. // Total Payments Calculation  
409. // ****  
410.  
411. double totalPayments(string nameFoodPrice[ ][COL], int arrFoodChoice, double orderPrice, double &newOrderPrice) {  
412.  
413.  
414.     orderPrice = strtod(nameFoodPrice[arrFoodChoice][1].c_str(), NULL);  
415.     newOrderPrice += orderPrice;  
416.  
417.     return newOrderPrice;  
418.  
419. }  
420.  
421. // ****  
422. // Total Estimated Delivery Time Calculation  
423. // ****  
424.  
425. int deliveryTime(string nameFoodPrice[ ][COL], int arrFoodChoice, int estimatedTime, int &newEstimatedTime) {  
426.  
427.     estimatedTime = strtod(nameFoodPrice[arrFoodChoice][2].c_str(), NULL);  
428.     newEstimatedTime += estimatedTime;  
429.  
430.     return newEstimatedTime;  
431. }
```



```
491. // *****
492. //          Customers make order
493. // *****
494.
495. void makeOrder(string nameFoodPrice[][COL], int num_food, int popularFood[FOOD]){
496.
497.     const int NUM_ORDER = 15;
498.     char confirm, paymentChoice;
499.     int foodChoice, arrFoodChoice;
500.     int totalDeliveryTime, estimatedTime, newEstimatedTime = 0;
501.     static int num_customers = 0;
502.     int num_orders_customers = 0;
503.     double orderPrice, newOrderPrice = 0, totalPay;
504.     bool orderFlag;
505.     int num_most_popular_dish;
506.     int stockFood[FOOD];
507.
508.     for (int i = 0; i < num_food; i++){
509.
510.         if (i < num_food){
511.
512.             stockFood[i] = stoi(nameFoodPrice[i][3]);
513.         }
514.
515.         else{
516.             stockFood[i] = 0;
517.         }
518.     }
519.
520.
521.     do{
522.         cout << endl;
523.         cout << "Please enter the number as shown in menu to order the food: ";
524.         cin >> foodChoice;
525.
526.         while (foodChoice < 1 || foodChoice > num_food){
527.
528.             cout << "Invalid number. Please enter the number again: Food ";
529.             cin >> foodChoice;
530.         }
531.
532.         arrFoodChoice = foodChoice - 1;
533.
534.         cout << setprecision(2) << fixed << showpoint;
535.         cout << "You order of food is " << nameFoodPrice[arrFoodChoice][0] << endl;
536.
537.         totalPay = totalPayments(nameFoodPrice, arrFoodChoice, orderPrice, newOrderPrice);
538.         totalDeliveryTime = deliveryTime(nameFoodPrice, arrFoodChoice, estimatedTime, newEstimatedTime);
```

```
539.     num_most_popular_dish = popularDish(nameFoodPrice, num_food, arrFoodChoice, popularFood);
540.     orderFlag = acceptOrder(nameFoodPrice, num_food, arrFoodChoice, stockFood);
541.
542.     if (orderFlag == true){
543.         cout << "Order Accepted! " << endl;
544.     }
545.
546.     else{
547.         cout << "Out of stock! You will be returned to main menu " << endl;
548.         system("pause");
549.         system("CLS");
550.         mainMenu(popularFood);
551.     }
552.
553.     cout << "The total price is RM" << totalPay << endl << endl;
554.     cout << "Would you like to add more? (Y/N)" << endl;
555.     cin >> confirm;
556.
557.     while(confirm != 'Y' && confirm != 'y' && confirm != 'N' && confirm != 'n'){
558.         cout << "Invalid! Please try again! " << endl;
559.         cin >> confirm;
560.
561.     }
562.
563.     num_orders_customers++;
564. } while (confirm == 'Y' || confirm == 'y');
565.
566. num_customers++;
567.
568. cout << endl << "You have ordered " << num_orders_customers << " dish(es) " << endl;
569. cout << "Estimated delivery time is " << totalDeliveryTime << " minutes" << endl;
570. cout << "The total payment is RM " << totalPay << endl << endl;
571. cout << "Press P to make your payment: ";
572. cin >> paymentChoice;
573.
574. while(paymentChoice != 'P' && paymentChoice != 'p'){
575.     cout << "Invalid! Please try again! " << endl;
576.     cin >> paymentChoice;
577. }
578.
579. cout << endl << endl;
580. system("pause");
581. system("CLS");
582. payment(nameFoodPrice, num_food, totalPay, popularFood, num_customers, num_most_popular_dish);
583.
584.
```

```
584.  
585. // ****  
586. // Customers make payment  
587. // ****  
588.  
589. void payment(string nameFoodPrice[][COL], int num_food, double totalPay, int popularFood[FOOD], int num_customers, int num_most_popular_dish){  
590.  
591.     double pay;  
592.     static double sales = 0;  
593.  
594.     cout << "Please enter the amount of your payment: RM ";  
595.     cin >> pay;  
596.  
597.     while (pay != totalPay){  
598.  
599.         cout << "Invalid amount! Please enter the exact amount of your payment: RM ";  
600.         cin >> pay;  
601.     }  
602.  
603.     totalSalesDay(totalPay, sales);  
604.     displayMessage(nameFoodPrice, num_customers, sales, num_most_popular_dish);  
605.  
606.     cout << "Payment successful! " << endl;  
607.     cout << "Please come again! :D " << endl;  
608.  
609.     cout << endl << endl;  
610.     system("pause");  
611.     system("CLS");  
612.     customerView(nameFoodPrice, num_food, popularFood);  
613.  
614. }  
615.  
616. // ****  
617. // Total Sales Calculation  
618. // ****  
619.  
620. void totalSalesDay(double totalPay, double &sales){  
621.  
622.     sales += totalPay;  
623.  
624. }
```

```
625.  
626. // **** Determine Popular Dish ****  
627. // ****  
628.  
629.  
630. int popularDish(string nameFoodPrice[][COL], int num_food, int arrFoodChoice, int popularFood[FOOD]) {  
631.  
632.     int mostPopular, numMostPopular;  
633.  
634.  
635.     popularFood[arrFoodChoice]++;  
636.  
637.     mostPopular = popularFood[0];  
638.  
639.  
640.     for (int x = 1; x < FOOD; x++) {  
641.  
642.         if (mostPopular < popularFood[x]) {  
643.  
644.             mostPopular = popularFood[x];  
645.             numMostPopular = x;  
646.  
647.         }  
648.  
649.     }  
650.  
651.     return numMostPopular;  
652. }  
653.  
654. // **** Output the file ****  
655. // ****  
656.  
657.  
658. void displayMessage (string nameFoodPrice[][COL], int num_customers, double sales, int num_most_popular_dish) {  
659.  
660.     ofstream outFile;  
661.     outFile.open("OutputFile.txt");  
662.  
663.     outFile << "Number of customers for the day: " << num_customers << endl;  
664.     outFile << setprecision(2) << fixed << showpoint;  
665.     outFile << "The total sales is RM " << sales << endl;  
666.     outFile << "The most popular dish is " << nameFoodPrice[num_most_popular_dish][0] << endl;  
667.  
668.     outFile.close();  
669. }
```

Sample Input and Output



File Edit Format View Help				
Grilled Chicken Burger	12.00	12	50	
Double Cheeseburger	13.00	11	50	
Filet-O-Fish Burger	11.00	11	40	
Beef Burger	14.00	13	70	
Veggie Burger	10.00	10	40	
Cheezy Wedges	7.00	5	80	
French Fries	5.00	5	80	

Food menu text file.

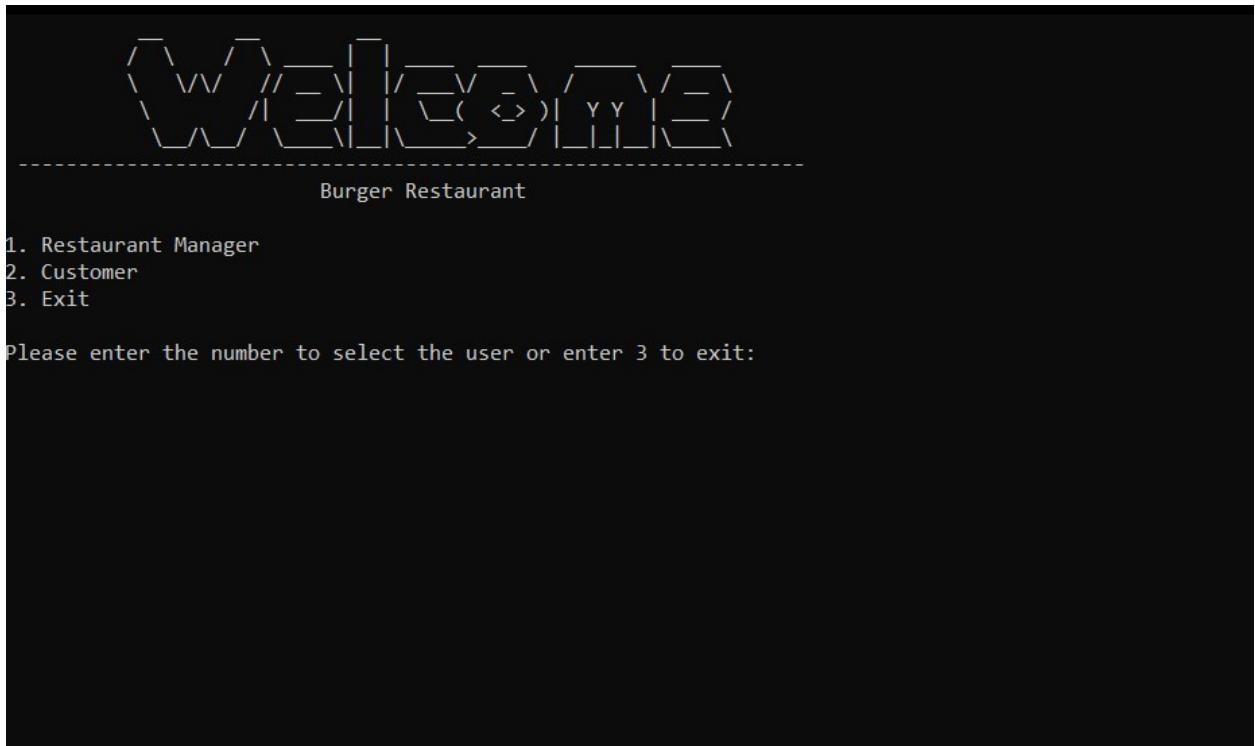
Example:

For row 1, Grilled Chicken Burger represents the name of food.

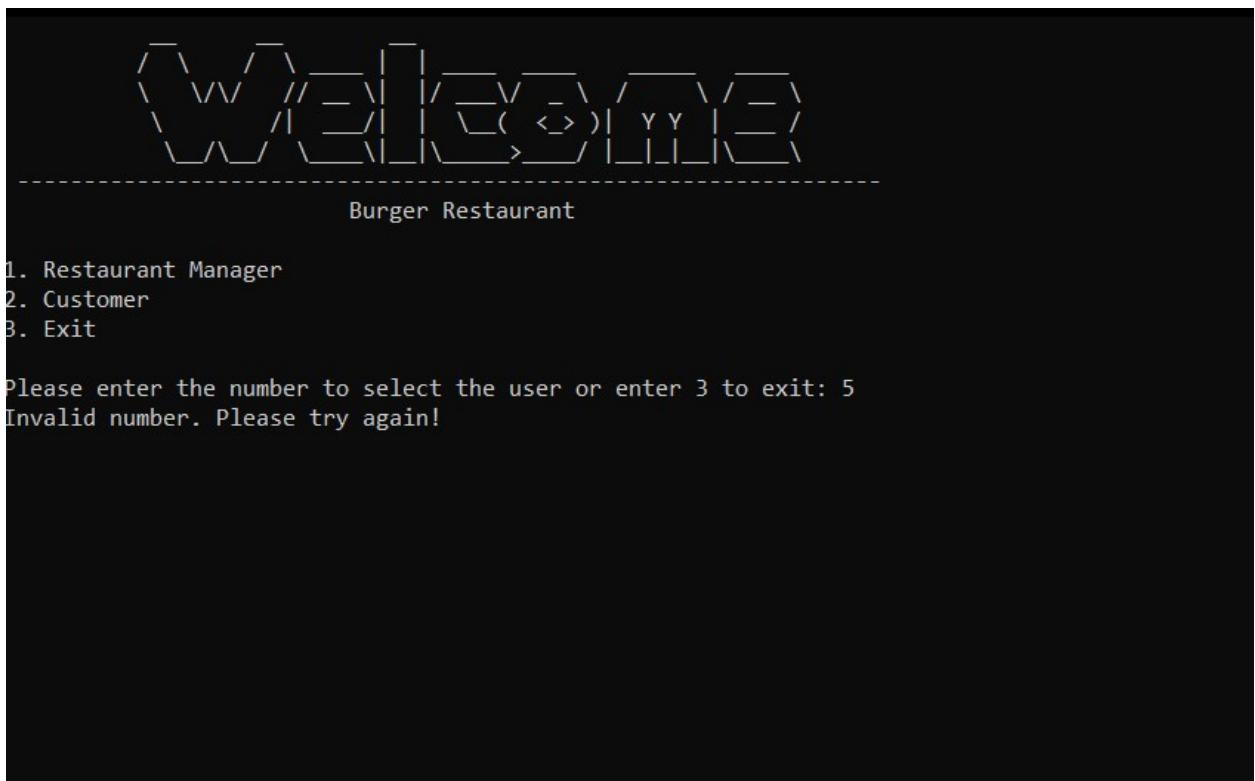
12.00 means the price of the food in ringgit Malaysia.

12 means the estimated delivery time.

50 represents the stock of the food.



Greet the user. After that, print the message and ask user to select the user.



If the user enter number other than 1,2 and 3, the program will ask the user to enter the number again.

Manager Access

1. Create / Update Menu
2. Update Prices

Please select a number for further details:

If the user selects the number 1 which is restaurant manager, display the Manager Access menu to select the choice.

Manager Access

1. Create / Update Menu
2. Update Prices

Please select a number for further details: 3

Invalid number! Please try again!

If the user enter number other than 1 and 2, the program will ask the user to enter the number again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter a new food name: Spicy Hot Chicken Burger  
Please enter the price of the food: RM 3  
  
The price of the food must be set in the range from RM 4 to RM 30  
Please set the price of the food again: RM
```

If the user enters number 1 which is create/update menu, the program will ask the user to enter a new food name and the price of the food. If the user enters the price of food which is less than RM 4 or more than RM 30, it will display the message and prompt user to enter the price of food again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter a new food name: Spicy Hot Chicken Burger  
Please enter the price of the food: RM 11  
  
Please enter the estimated delivery time for the food (in minutes): 35  
The delivery time for the food must be set in the range from 5 minutes to 30 minutes  
Please set the delivery time again:
```

Then, the program will ask the user to enter the estimated delivery time for the food. If the user enters the time which is less than 5 minutes or more than 30 minutes, the program will display message and prompt the user to enter the estimated delivery time again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter a new food name: Spicy Hot Chicken Burger
Please enter the price of the food: RM 11

Please enter the estimated delivery time of the food (in minutes): 15
Please enter the stock of the food: 50

Would you like to add another food? (Y/N) Your choice:
```

After that, the program will ask the user to enter the stock of the food. The program will continue ask the user whether the user needs to add another food or not.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter a new food name: Spicy Hot Chicken Burger
Please enter the price of the food: RM 11

Please enter the estimated delivery time of the food (in minutes): 15
Please enter the stock of the food: 50

Would you like to add another food? (Y/N) Your choice: P
Invalid ! Please try again!
```

If the user enters the choice other than ‘Y’ and ‘N’, the program will ask the user to enter the choice again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter a new food name: Spicy Hot Chicken Burger
Please enter the price of the food: RM 11

Please enter the estimated delivery time of the food (in minutes): 15
Please enter the stock of the food: 50

Would you like to add another food? (Y/N) Your choice: P
Invalid ! Please try again!
N

Update Successfully! Please check the updated menu

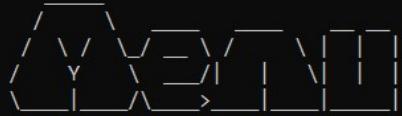
-----
Process exited after 212.3 seconds with return value 0
Press any key to continue . . .
```

If the user enters ‘N’, the program will end and the menu is updated successfully.

```
MenuFood.txt - Notepad
File Edit Format View Help
Grilled Chicken Burger 12.00 12 50
Double Cheeseburger 13.00 11 50
Filet-O-Fish Burger 11.00 11 40
Beef Burger 14.00 13 70
Veggie Burger 10.00 10 40
Cheezy Wedges 7.00 5 80
French Fries 5.00 5 80
Spicy Hot Chicken Burger 11.00 15 50
```

The new food menu is written into the MenuFood.txt file successfully.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

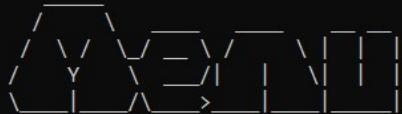


```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Please enter the number available in order to update the food price: Food
```

If the user enters number 2 which is update price, the program displays the menu and ask the user to enters the number available in food menu.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

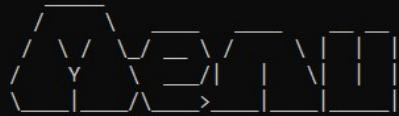


```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Please enter the number available in order to update the food price: Food 0
Invalid number. Please enter again: Food
```

If the user enters the number less than 1 or more than the number of food in the menu which is 7, the program will ask the user to enter the number again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Please enter the number available in order to update the food price: Food 7
```

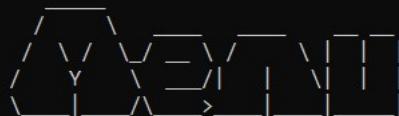
```
Please enter the new price for the dish: RM 3
```

```
The price of the dish must be updated in the range from RM 4 to RM 30
```

```
Please update the price of the dish again: RM
```

Then, the program will ask the user to enter the new price for the dish. If the price of the dish is less than RM 4 or more than RM 30, the program will display the message and prompt the user to enter the price of the dish again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Please enter the number available in order to update the food price: Food 7
```

```
Please enter the new price for the dish: RM 6
```

```
Update Successfully! Kindly check the updated menu
```

```
Process exited after 111.8 seconds with return value 0
Press any key to continue . . .
```

The updated price will write int the MenuFood.txt file.

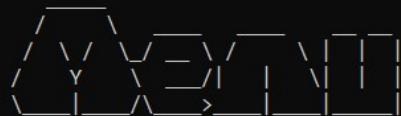
 MenuFood.txt - Notepad

File Edit Format View Help

Grilled Chicken Burger	12.00	12	50
Double Cheeseburger	13.00	11	50
Filet-O-Fish Burger	11.00	11	40
Beef Burger	14.00	13	70
Veggie Burger	10.00	10	40
Cheezy Wedges	7.00	5	80
French Fries	6.00	5	80

The updates price of the food which is French Fries is written into the file successfully.

 C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe

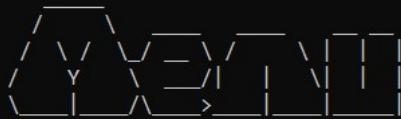


```
-----  
Food 1: Grilled Chicken Burger      RM 12.00  
Food 2: Double Cheeseburger       RM 13.00  
Food 3: Filet-O-Fish Burger       RM 11.00  
Food 4: Beef Burger               RM 14.00  
Food 5: Veggie Burger             RM 10.00  
Food 6: Cheezy Wedges             RM 7.00  
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)  
Would you like to make food order? (Y/N)  
Press Y or to make food order or press N to exit. Your choice:
```

If the user selects the number 2 which is customer, display the food menu to enter the the choice.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)
```

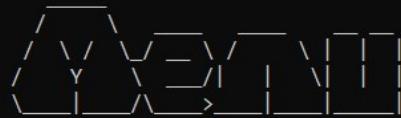
```
Would you like to make food order? (Y/N)
```

```
Press Y or to make food order or press N to exit. Your choice: P
```

```
Invalid choice ! Please try again! Your choice:
```

If the user enters the choice other than ‘Y’ and ‘N’, the program will ask the user to enter the choice again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)
```

```
Would you like to make food order? (Y/N)
```

```
Press Y or to make food order or press N to exit. Your choice: P
```

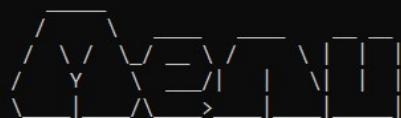
```
Invalid choice ! Please try again! Your choice: N
```

```
Thank you! Please come again! :D
```

```
Process exited after 153 seconds with return value 0
Press any key to continue . . .
```

If the user enters the choice ‘N’, the program will print the message and thanks the user.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



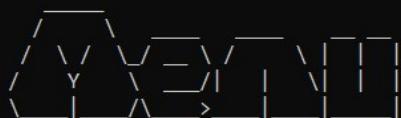
```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)
Would you like to make food order? (Y/N)
Press Y or to make food order or press N to exit. Your choice: Y
```

```
Please enter the number as shown in menu to order the food:
```

If the user enters choice ‘Y’, the program will ask the user to select the number as shown in the food menu.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



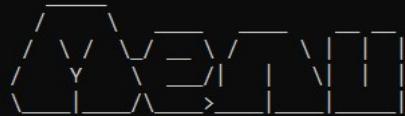
```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)
Would you like to make food order? (Y/N)
Press Y or to make food order or press N to exit. Your choice: Y
```

```
Please enter the number as shown in menu to order the food: 0
Invalid number. Please enter the number again: Food
```

If the user enters the number which is less than 1 or more than the number of foods shown in the menu, the program will ask the user to enter the number again.

C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe



Food 1: Grilled Chicken Burger	RM 12.00
Food 2: Double Cheeseburger	RM 13.00
Food 3: Filet-O-Fish Burger	RM 11.00
Food 4: Beef Burger	RM 14.00
Food 5: Veggie Burger	RM 10.00
Food 6: Cheezy Wedges	RM 7.00
Food 7: French Fries	RM 5.00

Welcome! New Customer! :)

Would you like to make food order? (Y/N)

Press Y or to make food order or press N to exit. Your choice: Y

Please enter the number as shown in menu to order the food: 1

You order of food is Grilled Chicken Burger

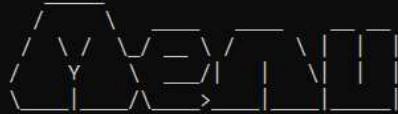
Order Accepted!

The total price is RM12.00

Would you like to add more? (Y/N)

If the user enters number 1, Food 1 which is Grilled Chicken Burger will be added into order. Then, the program will ask the user to enter the choice (Y/N). If the user enters 'Y', the program will continue ask the user to order the food until the user enter the choice 'N'.

C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe



Food 1: Grilled Chicken Burger	RM 12.00
Food 2: Double Cheeseburger	RM 13.00
Food 3: Filet-O-Fish Burger	RM 11.00
Food 4: Beef Burger	RM 14.00
Food 5: Veggie Burger	RM 10.00
Food 6: Cheezy Wedges	RM 7.00
Food 7: French Fries	RM 5.00

Welcome! New Customer! :)
Would you like to make food order? (Y/N)
Press Y or to make food order or press N to exit. Your choice: Y

Please enter the number as shown in menu to order the food: 1
You order of food is Grilled Chicken Burger
Order Accepted!
The total price is RM12.00

Would you like to add more? (Y/N)
P
Invalid ! Please try again!

If the user enters the choice other than 'Y' and 'N', the program will ask the user to enter the choice again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
-----
[menu logo]
-----
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger                RM 14.00
Food 5: Veggie Burger              RM 10.00
Food 6: Cheezy Wedges              RM 7.00
Food 7: French Fries               RM 5.00

Welcome! New Customer! :)
Would you like to make food order? (Y/N)
Press Y or to make food order or press N to exit. Your choice: Y

Please enter the number as shown in menu to order the food: 1
You order of food is Grilled Chicken Burger
Order Accepted!
The total price is RM12.00

Would you like to add more? (Y/N)
Y

Please enter the number as shown in menu to order the food: 2
You order of food is Double Cheeseburger
Order Accepted!
The total price is RM25.00

Would you like to add more? (Y/N)
N

You have ordered 2 dish(es)
Estimated delivery time is 23 minutes
The total payment is RM 25.00

Press P to make your payment:
```

After the user enters the choice ‘N’, the program will display the messages which are the number of orders by customer, the estimated delivery time and the total payment. Next, the program will ask the user to enter ‘P’ to make payment.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
Food 1: Grilled Chicken Burger      RM 12.00
Food 2: Double Cheeseburger        RM 13.00
Food 3: Filet-O-Fish Burger       RM 11.00
Food 4: Beef Burger               RM 14.00
Food 5: Veggie Burger             RM 10.00
Food 6: Cheezy Wedges             RM 7.00
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)
Would you like to make food order? (Y/N)
Press Y or to make food order or press N to exit. Your choice: Y
```

```
Please enter the number as shown in menu to order the food: 1
You order of food is Grilled Chicken Burger
Order Accepted!
The total price is RM12.00
```

```
Would you like to add more? (Y/N)
Y
```

```
Please enter the number as shown in menu to order the food: 2
You order of food is Double Cheeseburger
Order Accepted!
The total price is RM25.00
```

```
Would you like to add more? (Y/N)
N
```

```
You have ordered 2 dish(es)
Estimated delivery time is 23 minutes
The total payment is RM 25.00
```

```
Press P to make your payment: L
Invalid! Please try again!
```

If the user enters the alphabet other ‘P’, the program will ask the user to enter the alphabet again.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
Please enter the amount of your payment: RM 24
Invalid amount! Please enter the exact amount of your payment: RM
```

The program will ask the user to enter the amount of payment. If the payment entered is not the same as the amount of payment of the order, It will ask user to enter the amount of payment again.

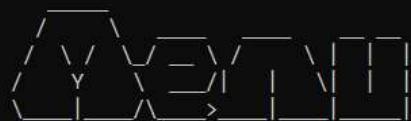
```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```

```
Please enter the amount of your payment: RM 24
Invalid amount! Please enter the exact amount of your payment: RM 25
Payment successful!
Please come again! :D
```

```
Press any key to continue . . .
```

If the payment entered is correct, the program will display message to show the payment is successful and thanks user.

```
C:\Users\cheon\OneDrive\Desktop\Assignment Part 2\Practice\TestingC++2.exe
```



```
-----  
Food 1: Grilled Chicken Burger      RM 12.00  
Food 2: Double Cheeseburger       RM 13.00  
Food 3: Filet-O-Fish Burger       RM 11.00  
Food 4: Beef Burger               RM 14.00  
Food 5: Veggie Burger             RM 10.00  
Food 6: Cheezy Wedges            RM 7.00  
Food 7: French Fries              RM 5.00
```

```
Welcome! New Customer! :)  
Would you like to make food order? (Y/N)  
Press Y or to make food order or press N to exit. Your choice:
```

The program will go back to the menu display and ask another customer to order the food.

 OutputFile.txt - Notepad

File Edit Format View Help

Number of customers for the day: 3

The total sales is RM 96.00

The most popular dish is Grilled Chicken Burger

Once the payment is made by customer successfully, the program will record the information which includes number of customers for the day, the total sales and the most popular dish into the OutputFile.txt file so that the manager is able to view the information.