



UNIVERSITI TEKNOLOGI MARA
KEDAH BRANCH
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS (CDIM144)
IML208: PROGRAMMING FOR LIBRARIES

FOOD ORDER

Prepared by:

NUR ATIFAH SOPHIA BINTI SABRY (2022457012)

GROUP KCDIM144 3F.

Prepared for:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

Submission date:

4th DECEMBER 2023

FOOD ORDER

PREPARED BY:

NUR ATIFAH SOPHIA BINTI SABRY (2022457012)

GROUP KCDIM144 3F

CDIM144 – DIPLOMA IN LIBRARY INFORMATICS

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

ACKNOWLEDGEMENT

The success and final outcome of the assignment required a lot of patience and assistance from many people, and I am extremely fortunate to have got this all along the completion of this assignment work. Whatever I have done is only due to such guidance and assistance and I could never forget them. I respect and thank Sir Airul for giving me an opportunity to do this assignment work and providing me with all the support and guidance that helped me complete the assignment on time. I am extremely grateful to him for providing such nice support and guidance.

I am very grateful to have done and completed this assignment in time thanks to the guidance of my lecturer, Sir Airul. Thus, this assignment could not have been completed without the help of my classmates and friends as well as my family who always supported me in doing this assignment.

TABLE OF CONTENT

ACKNOWLEDGEMENT	i
1.0 INTRODUCTION	1
2.0 FLOWCHART	2
3.0 PYTHON CODE	3
4.0 GUI	6
5.0 DATABASE	7
6.0 CONCLUSION	8

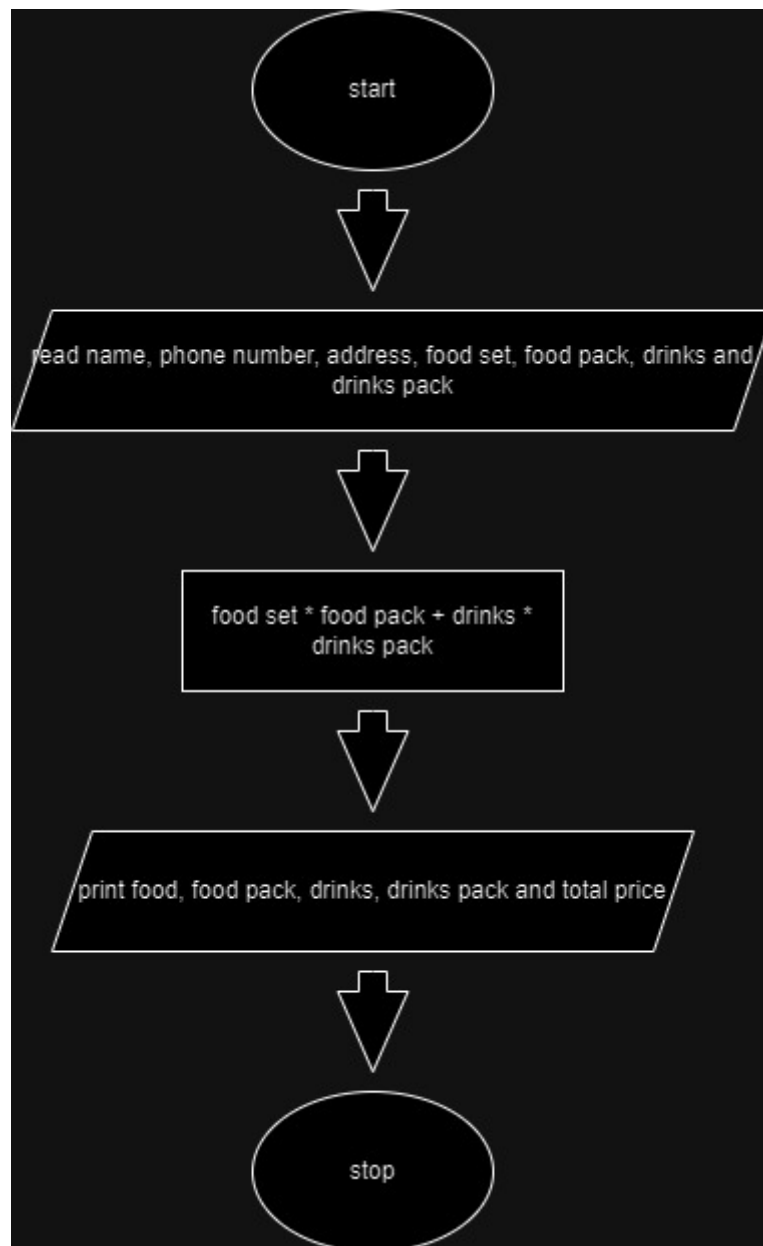
INTRODUCTION

To begin with, this “Food order” system is made to create a better and more systematic way for buyers and sellers to interact. This system aims for sellers who always interact with their customers daily such as sellers who provide food services (receive orders and make deliveries). These kinds of sellers are very common, especially among students who live in college. However, they do not have the proper system for accepting orders from the customers.

By using this system, buyers just need to insert the data for the order, and it will calculate the total amount of the order. Subsequently, all the data will be saved in the database of the seller. Because of this method, there will be less risk for sellers to overlook and miss out on any order as often happens when using the traditional way to receive orders; buyers would send their orders through WhatsApp or other communication platforms without proper form.

Furthermore, this system will need the buyers to fill in all the information needed such as their names, phone numbers, and addresses, as well as choose from the list of food and drinks. All of this information will be recorded in the database. After choosing the set of food, the system will calculate the total price of the order by clicking the “Calculate” button. This method will facilitate the process because buyers no longer need to calculate the total by themselves. In other perspectives, it will also decrease the risk of miscalculating or any error that may occur regarding the calculation.

FLOWCHART



PYTHON CODE

```
208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
1 import tkinter as tk
2 import mysql.connector
3 from tkinter import messagebox
4
5 # Connect to your MySQL database
6 mydb = mysql.connector.connect(
7     host="localhost",
8     user="root",
9     password="",
10    database="food_order"
11 )
12
13 # Create a cursor object to execute SQL queries
14 mycursor = mydb.cursor()
15
16 # Example SQL query
17 sql_query = "SELECT * FROM `order`"
18
19 # Execute the SQL query
20 mycursor.execute(sql_query)
21
22 # Fetch the result
23 result = mycursor.fetchall()
24
25 # Function to handle the calculation and database saving
26 def collect_data():
27     name = name_entry.get()
28     phone_no = int(phone_no_entry.get())
29     address = address_entry.get()
30     food_set = food_var.get()
31     food_pack = int(food_pack_entry.get())
32     drinks = drinks_var.get()
```

```
208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
32     drinks = drinks_var.get()
33     drinks_pack = int(drinks_pack_entry.get())
34
35     # Prices for the selections
36     prices_dish = {
37         "Set A": 6,
38         "Set B": 7,
39         "Set C": 8,
40     }
41
42     prices_drinks = {
43         "Orange juice": 3,
44         "Ice chocolate": 2,
45         "Sky juice": 1
46     }
47
48     # Clear entry widgets after calculation
49     name_entry.delete(0, tk.END)
50     address_entry.delete(0, tk.END)
51     phone_no_entry.delete(0, tk.END)
52     food_pack_entry.delete(0, tk.END)
53     drinks_pack_entry.delete(0, tk.END)
54
55     # Calculate the total price
56     total_price = int(prices_dish[food_set] * food_pack + prices_drinks[drinks] * drinks_pack)
57
58     # Insert data into the database
59     sql = "INSERT INTO `order` (name, phone_no, address, food_set, food_pack, drinks, drinks_pack) VALUES (%s, %s, %s, %s, %s, %s, %s)"
60     val = (name, phone_no, address, food_set, food_pack, drinks, drinks_pack)
61     mycursor.execute(sql, val)
62     mydb.commit()
```

```

208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
63
64     # Update the output label
65     output_label.config(text=f"Set: {food_set}, Packs: {food_pack}, Drinks: {drinks}, Packs: {drinks_pack}, Total Price: RM{total_price}", fg="bl
66
67 # Main window
68 window = tk.Tk()
69 window.title("Food Order")
70
71 frame = tk.Frame(window)
72 frame.grid(row=0, column=0)
73
74 # Add widgets to the main window
75 label = tk.Label(frame, text="Fill the information")
76 label.grid(row=0, column=0, padx=10, pady=10)
77
78 user_info_frame = tk.LabelFrame(frame, text="Customer Information")
79 user_info_frame.grid(row=1, column=0, padx=20, pady=10)
80
81 name_label = tk.Label(user_info_frame, text="Name")
82 name_label.grid(row=0, column=0)
83 name_entry = tk.Entry(user_info_frame)
84 name_entry.grid(row=0, column=1, padx= 10, pady= 10)
85
86 address_label = tk.Label(user_info_frame, text="Address")
87 address_label.grid(row=1, column=0)
88 address_entry = tk.Entry(user_info_frame)
89 address_entry.grid(row=1, column=1, padx= 10, pady= 20)
90
91 phone_no_label = tk.Label(user_info_frame, text="Phone Number (01XXXXXX)")
92 phone_no_label.grid(row=2, column=0)
93
94

```

```

208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
94 def is_numeric(value):
95     try:
96         integer_value = int(value)
97         return True
98     except ValueError:
99         return False
100
101 def validate_phone_number(value):
102     if is_numeric(value) or value == "":
103         return True
104     else:
105         messagebox.showerror("Error", "Phone number must be numeric")
106         return False
107
108 phone_no_entry = tk.Entry(user_info_frame, validate="key", validatecommand=(validate_phone_number, "%P"))
109 phone_no_entry.grid(row=2, column=1)
110
111 second_window = tk.Toplevel(window)
112 second_window.title("Order detail")
113
114 # Button to open the second window
115 second_window_button = tk.Button(frame, text="Go to Order Details", command= second_window)
116 second_window_button.grid(row=3, column=0, pady=10)
117
118 # Order details display
119 order_detail_label = tk.Label(second_window, text="Order Details:")
120 order_detail_label.grid()
121
122 # The defined list by using pricebox
123 prices_text = tk.Text(second_window, height=15, width=45)
124 prices_text.grid(pady=20)

```



```

208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
124 prices_text.grid(pady=20)
125 prices_text.insert(tk.END, "Rice & Prices:\n\n")
126 prices_text.insert(tk.END, "Set A: Rice, Fried chicken, Vegetables \nPrice: RM6\n\n")
127 prices_text.insert(tk.END, "Set B: Rice, Squid dish, Vegetables \nPrice: RM7\n\n")
128 prices_text.insert(tk.END, "Set C: Rice, Prawn dish, Vegetables \nPrice: RM8\n\n")
129 prices_text.configure(state='disabled')
130
131 # Trip Type Dropdown
132 food_var = tk.StringVar(second_window)
133 food_var.set("Select Your Set")
134 trip_dropdown = tk.OptionMenu(second_window, food_var, "Set A", "Set B", "Set C")
135 trip_dropdown.grid(pady=10)
136
137 # Packs Entry. Label and user can insert data thru entry
138 label = tk.Label(second_window, text="Packs:")
139 label.grid()
140 food_pack_entry = tk.Entry(second_window)
141 food_pack_entry.grid()
142
143 # Drinks type
144 drinks_var = tk.StringVar(second_window)
145 drinks_var.set("Select Your Drinks")
146 trip_dropdown = tk.OptionMenu(second_window, drinks_var, "Orange juice", "Ice chocolate", "Sky juice")
147 trip_dropdown.grid(pady=10)
148
149 # Packs Entry. Label and user can insert data thru entry
150 label_2 = tk.Label(second_window, text="Packs:")
151 label_2.grid()
152 drinks_pack_entry = tk.Entry(second_window)
153 drinks_pack_entry.grid()
154

```

```

208assignment_riceorder.py X
C: > Users > HP > OneDrive - Universiti Teknologi MARA > IML208 > iml208 > Class Materials Week 11 > Class Materials Week 11 > 208assignment_riceorder.py > collect_data
155 # Calculate button in the second window
156 calculate_button = tk.Button(second_window, text="Calculate", command=collect_data)
157 calculate_button.grid(pady=10)
158
159 # Output Label & result
160 label = tk.Label(second_window)
161 label.grid(ipadx=10, ipady=10)
162 output_label = tk.Label(second_window, text="")
163 output_label.grid()
164
165 window.mainloop()
166
167 # Close the database connection
168 mydb.close()

```

GUI

The image displays two windows from a food ordering application. The 'Order detail' window on the left shows a list of food sets and their prices. The 'Food Order' window on the right is a form for entering customer information.

Order detail

Order Details:

Rice & Prices:

Set A: Rice, Fried chicken, Vegetables
Price: RM6

Set B: Rice, Squid dish, Vegetables
Price: RM7

Set C: Rice, Prawn dish, Vegetables
Price: RM8

Select Your Set

Packs:

Select Your Drinks

Packs:

Calculate

Food Order

Fill the information

Customer Information

Name

Address

Phone Number (01XXXXXX)

Go to Order Details

DATABASE

Server: 127.0.0.1 » Database: food_order » Table: order

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#) [Triggers](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 0 (1 total, Query took 0.0038 seconds.)

`SELECT * FROM `order``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

name	phone_no	address	food_set	food_pack	drinks	drinks_pack
ali	123456789	malinja	Set A		2 Orange juice	3

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

Label: ☐ Let every user access this bookmark

[Console](#)

Server: 127.0.0.1 » Database: food_order » Table: order

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#) [Triggers](#)

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	name	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 2	phone_no	int(10)			No	None			Change Drop More
<input type="checkbox"/> 3	address	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	food_set	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	food_pack	int(10)			No	None			Change Drop More
<input type="checkbox"/> 6	drinks	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 7	drinks_pack	int(10)			No	None			Change Drop More

⬆ ☐ Check all | With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#) [Fulltext](#) [Add to central columns](#)

[Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

[Add](#) 1 column(s) after drinks_pack [Go](#)

Indexes ⓘ

⚠ No index defined!

[Console](#) index on 1 columns [Go](#)

CONCLUSION

To conclude, this system is easy to use and can be applied to small businesses which does not have the proper system to receive orders. The form (GUI) can be changed according to the required information a seller wants. In this matter, changes could be made to ensure that all the information is delivered to the customer as well as all received all the details of the order from them.

The output of the system (order details) will be highlighted in blue colour to ensure that it is noticeable to the customer. Other than that, the list of food is provided above the order entry for easy reference. Last but not least, I hope this system can facilitate the activities of small business owners out there.