



UNIVERSITI TEKNOLOGI MARA (UITM)
CAWANGAN MERBOK, SUNGAI PETANI, KEDAH
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS.

DIPLOMA IN LIBRARY INFORMATICS (CDIM 144)

PROGRAMMING FOR LIBRARIES (IML 208)

INDIVIDUAL ASSIGNMENT:

CREATE AN INTERFACE WITH PYTHON CODES, GUI AND DATABASE
(ONLINE FOOD ORDERING : BOB BURGER SURRR)

PREPARED BY:

MUHAMMAD AMMAR BIN KHAMISAN (2022895812)

GROUP: KCDIM1443B

PREPARED FOR

SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE: **WEEK 12 (4TH JANUARY 2024)**

INDIVIDUAL ASSIGNMENT:

**CREATE AN INTERFACE WITH PYTHON CODES, GUI AND DATABASE
(ONLINE FOOD ORDERING : BOB BURGER SURRR)**

BY:

MUHAMMAD AMMAR BIN KHAMISAN

2022895812

KCDIM1443B

UNIVERSITI TEKNOLOGI MARA (UITM)
CAWANGAN MERBOK, SUNGAI PETANI, KEDAH
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS.

DIPLOMA IN LIBRARY INFORMATICS

WEEK 6

ACKNOWLEDGEMENT

I'm extremely grateful to god because of gave me spirit and strength to complete this assignment 1 individual IML208. This assignment is so important for me to obtain the precious marks, because of that, I will do this assignment well with all the strength, knowledge and time that I have.

I am deeply indebted with my lecturer in subject IML208 (Programming for libraries) class KCDIM1443B, Sir Airul Shazwan Bin Norshahimi for helping me to complete this assignment in the best possible way. Without his guidance seniority, it would be difficult for me to complete this properly. When doing this assignment, I learned a lot about how to create an interface by using the python codes and GUI, and the connect the data into database. This also opened my eyes about how the systems in this world are built and the functions.

This endeavor would not have been possible without my family, they always support and motivate me to keep moving forward in goals to achieve success in my life even though they are far away at the village. They will never leave alone with problems. I will never forget their sacrifices for helping me in every problem I have faced in my life.

Lastly, words cannot express my gratitude to my classmates, roommates and other friends because of they always help and accompany me if I have problems to complete this assignment. Without them, I will not successfully to complete it.



STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

- a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.
- b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.
- c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
- d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
- e. **Furnishing false information:** Providing false information or false representation to any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in any manner, violate this pledge.

Name: **MUHAMMAD AMMAR BIN KHAMISAN**

Matric Number: **2022895812**

Course Code: **CDIM144**

Programme code: **IML209**

Faculty / Campus: **COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS.**

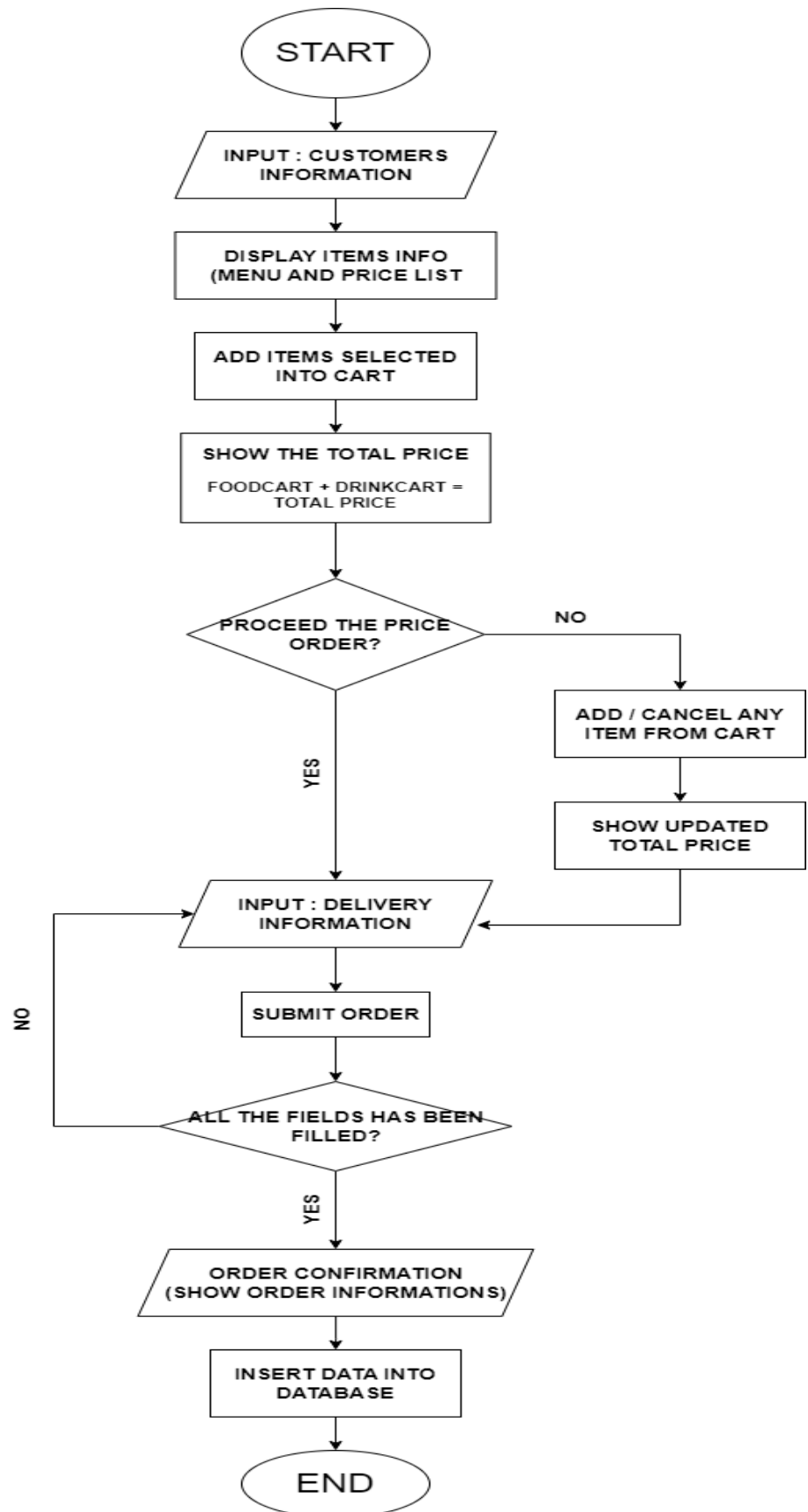
TABLE OF CONTENTS

CONTENTS	PAGE NUMBER
INTRODUCTION	5
FLOWCHART	6
SNAPSHOT OF PYTHON CODES	7 - 11
SNAPSHOT OF GUI	12 - 13
SNAPSHOT OF DATABASE	14

INTRODUCTION

The title for my individual assignment IML208 is Online Food Ordering, which describes a burger shop called as "Bob Burger Surr" that takes orders using an online food ordering system and filled in by the customers. Then the information is entered into the database to ensure that the order can be accepted by the burgershop to complete the order and delivery to customers.

FLOWCHART :



THE SNAPSHOT OF PYTHON CODES

```
C:\> Users > User > Documents > source code > ONLINE_FOOD_ORDERING.py > ...
1  import tkinter as tk
2  from tkinter import ttk, scrolledtext
3  from tkinter import messagebox
4  from tkcalendar import DateEntry
5  import mysql.connector
6
7  # Food and drink prices
8  food_prices = {'Basic Burger': 3.0, 'Special Burger': 5.0, 'King Burger': 7.0, 'Giant Burger': 10.0,
9  |             | 'Gigantic Burger': 15.0, 'Luxury Golden Burger': 18.0, 'Demon Spicy Burger': 20.0}
10
11  drink_prices = {'Mineral Water': 1.0, 'Coke': 2.0, 'Sprite': 2.0, 'Orange Juice': 3.0,
12  |             | 'Lemon Tea': 3.0, 'Sparkling Grape': 5.0, 'Mountain Dew': 6.0}
13
14  # Cart lists
15  food_cart = []
16  drink_cart = []
17
18
19  def update_cart(action, item=None, cart=None, text_var=None, listbox=None):
20      global food_items, drink_items, total_price, delivery_time_hours_combobox, delivery_time_minutes_combobox
21
22      if action == 'add':
23          cart.append(item)
24          listbox.insert(tk.END, item)
25          calculate_total_price(text_var)
26      elif action == 'cancel':
27          selected_indices = listbox.curselection()
28          canceled_items = [cart[index] for index in selected_indices if index < len(cart)]
29
30          for index in reversed(selected_indices):
31              if index < len(cart):
32                  del cart[index]
33
34          listbox.delete(0, tk.END)
35          for item in cart:
36              listbox.insert(tk.END, item)
37
38          # Subtract the price of canceled items from the total price
39          canceled_price = sum(food_prices.get(item, 0) + drink_prices.get(item, 0) for item in canceled_items)
40          new_total_price = max(float(Total_price.get().split('RM')[1]) - canceled_price, 0)
41          Total_price.set(f'Total Price: RM{new_total_price:.2f}')
42      elif action == 'submit':
43
44          if not (customer_name_entry.get() and phone_number_entry.get() and delivery_location_entry.get() and
45  |             | delivery_date_entry.get() and delivery_time_hours_combobox.get() and delivery_time_minutes_combobox.get()):
46              messagebox.showerror('Error', 'Please fill in all required fields.')
47              return
48
49          food_items = ', '.join(food_cart)
50          drink_items = ', '.join(drink_cart)
51          total_price = sum(food_prices.get(item, 0) for item in food_cart) + sum(drink_prices.get(item, 0) for item in drink_cart)
52
53          customer_name = customer_name_entry.get()
54          phone_number = phone_number_entry.get()
55          Cart = ', '.join([cart_listbox.get(i) for i in range(cart_listbox.size())])
56          delivery_location = delivery_location_entry.get()
57          delivery_date = delivery_date_entry.get()
58          delivery_time_hours = delivery_time_hours_combobox.get()
59          delivery_time_minutes = delivery_time_minutes_combobox.get()
60          additional_notes = additional_notes_entry.get()
61
```

```

62     # Connect with the mysql database
63     mydb = mysql.connector.connect(
64         host="localhost",
65         user="root",
66         password="",
67         database="online_food_ordering"
68     )
69
70     mycursor = mydb.cursor()
71
72     # Insert order informations into the database
73     sql = "INSERT INTO customer_order (customer_name, phone_number, Cart, delivery_location, delivery_date, delivery_time_hours, delivery_time_minutes, additional_notes, total_price)"
74     val = (customer_name, phone_number, Cart, delivery_location, delivery_date, delivery_time_hours,
75           delivery_time_minutes, additional_notes, total_price)
76     mycursor.execute(sql, val)
77     mydb.commit()
78
79     # Display a messagebox with the confirmation message
80     confirmation_message = f'Your order has been received! Thank you for ordering with us!\n\n' \
81                             f'Menu Ordered: {food_items}, {drink_items}\n' \
82                             f'Total Price: RM{total_price:.2f}\n' \
83                             f'Delivery Date: {delivery_date_entry.get()}\n' \
84                             f'Delivery Time: {delivery_time_hours_combobox.get()}:{delivery_time_minutes_combobox.get()}\n' \
85                             f'Delivery Location: {delivery_location_entry.get()}\n\n' \
86                             f'Additional Notes: {additional_notes_entry.get()}\n\n' \
87                             f'Please get ready the amount, we will call you before delivery to make sure you able to pickup the order.'
88     messagebox.showinfo('Order Confirmation', confirmation_message)

```

```

90     # Reset the cart and clear the entries for insert new order of another customers
91     cart.clear()
92     listbox.delete(0, tk.END)
93     customer_name_entry.delete(0, tk.END)
94     phone_number_entry.delete(0, tk.END)
95     delivery_location_entry.delete(0, tk.END)
96     delivery_date_entry.delete(0, tk.END)
97     delivery_time_hours_combobox.set('')
98     delivery_time_minutes_combobox.set('')
99     additional_notes_entry.delete(0, tk.END)
100     Total_price.set('Total Price: RM0.00') # Reset total price
101
102     def calculate_total_price(text_var):
103         food_total = sum(food_prices[item] for item in food_cart)
104         drink_total = sum(drink_prices[item] for item in drink_cart)
105         new_total_price = food_total + drink_total
106         text_var.set(f'Total Price: RM{new_total_price:.2f}')
107
108

```



```

169 add_drink_button = ttk.Button(drink_frame, text='Add Drink to Cart',
170                               command=lambda: update_cart('add', drink_dropdown.get(), drink_cart, Total_price,
171                                                           cart_listbox))
172
173 add_drink_button.grid(row=1, column=0, pady=5)
174
175 # Display the Food and Drink Prices list
176 prices_text = scrolledtext.ScrolledText(frame, width=30, height=5, wrap=tk.WORD)
177 prices_text.grid(row=2, column=0, padx=10, pady=10, columnspan=2)
178 prices_text.insert(tk.INSERT, "Food Prices:\n")
179 for food, price in food_prices.items():
180     prices_text.insert(tk.INSERT, f"{food}: RM{price:.2f}\n")
181
182 prices_text.insert(tk.INSERT, "\nDrink Prices:\n")
183 for drink, price in drink_prices.items():
184     prices_text.insert(tk.INSERT, f"{drink}: RM{price:.2f}\n")
185
186 # Make the prices_text widget non-editable
187 prices_text.config(state=tk.DISABLED)
188
189 # Cart Section
190 Cart = ttk.LabelFrame(frame, text='Cart', labelanchor='n')
191 Cart.grid(row=4, column=0, padx=10, pady=10, columnspan=2, sticky='n')
192 cart_listbox = tk.Listbox(Cart, selectmode=tk.MULTIPLE, height=5)
193 cart_listbox.grid(row=0, column=0, pady=5)
194
195 cancel_cart_button = ttk.Button(Cart, text='Cancel Cart',
196                                 command=lambda: update_cart('cancel', None, food_cart + drink_cart, Total_price,
197                                                             cart_listbox))
198 cancel_cart_button.grid(row=1, column=0, pady=5)
199

```

```

200 Total_price = tk.StringVar() # Variable to store the total price
201 confirmation_label = ttk.Label(Cart, text='', font=('Helvetica', 12), textvariable=Total_price)
202 confirmation_label.grid(row=2, column=0, pady=5)
203
204 # Delivery Information Section
205 delivery_information = ttk.LabelFrame(frame, text='Delivery Information', labelanchor='n')
206 delivery_information.grid(row=5, column=0, padx=10, pady=10, columnspan=2, sticky='n')
207
208 # delivery location
209 delivery_location = ttk.Label(delivery_information, text='Delivery Location:')
210 delivery_location.grid(row=0, column=0, sticky='e', padx=5)
211 delivery_location_entry = ttk.Entry(delivery_information)
212 delivery_location_entry.grid(row=0, column=1, padx=5)
213
214 # delivery date
215 delivery_date = ttk.Label(delivery_information, text='Delivery Date:')
216 delivery_date.grid(row=1, column=0, sticky='e', padx=5)
217 delivery_date_entry = DateEntry(delivery_information, width=17, background='darkblue', foreground='white',
218                                borderwidth=2)
219 delivery_date_entry.grid(row=1, column=1, padx=5, rowspan=2) # Spanning multiple rows of the calendar
220
221 # For time I using the 24hours formats
222 delivery_time = ttk.Label(delivery_information, text='Delivery Time:')
223 delivery_time.grid(row=1, column=2, sticky='e', padx=5)
224
225 # Time Entry (Hour)
226 delivery_time_hours_combobox = ttk.Combobox(delivery_information, values=[f"{hour:02d}" for hour in range(10, 22)],
227                                             state='readonly', width=3)
228 delivery_time_hours_combobox.grid(row=1, column=3, padx=2)
229

```

```

230 # Time Entry (Minute)
231 delivery_time_minutes_combobox = ttk.Combobox(delivery_information,
232 values=[f"{minute:02d}" for minute in range(0, 60, 5)],
233 state='readonly', width=3)
234 delivery_time_minutes_combobox.grid(row=1, column=4, padx=2)
235
236 # Additional Notes for customers to add any additional request before proceed order
237 additional_notes = ttk.Label(delivery_information, text='Additional Notes:')
238 additional_notes.grid(row=3, column=0, sticky='e', padx=5)
239 additional_notes_entry = ttk.Entry(delivery_information)
240 additional_notes_entry.grid(row=3, column=0, colspan=4, padx=5, pady=10)
241
242 # Submit Order Section
243 submit_frame = ttk.Frame(frame)
244 submit_frame.grid(row=6, column=0, padx=10, pady=10, colspan=2, sticky='n')
245
246 submit_button = ttk.Button(submit_frame, text='Submit Order', command=lambda: update_cart('submit'))
247 submit_button.grid(row=0, column=0, pady=10)
248
249 root.mainloop()
250

```

THE SNAPSHOT OF GUI

This is the interface of the GUI, firstly customers need to insert the customer information for delivery purpose. The interface must be expand (as red circle) to see the whole GUI, or just click 'maximize expand' at above the GUI

ONLINE FOOD ORDERING AND DELIVERY

Bob Burger Surr

Customer Information

Name:

Phone Number:

Food Prices:

- Basic Burger: RM3.00
- Special Burger: RM5.00
- King Burger: RM7.00
- Giant Burger: RM10.00

Food Options

Add Food to Cart

Drink Options

Add Drink to Cart

Cart

Cancel Cart

Delivery Information

Delivery Location:

Delivery Date:

Delivery Time:

Additional Notes:

After customers seen the menu and price, they can choose the menu by click that option arrow, menu, and click add to cart. The total price will appear under the cart. To change or cancel the menu, customers can click the menu in cart and click the cancel cart, the updated price will appear.

The screenshot shows a web application interface for ordering food. At the top, a 'Food Prices' list displays: Basic Burger: RM3.00, Special Burger: RM5.00, King Burger: RM7.00, and Giant Burger: RM10.00. Below this, there are two sections: 'Food Options' with a dropdown menu set to 'Special Burger' and an 'Add Food to Cart' button; and 'Drink Options' with a dropdown menu set to 'Coke' and an 'Add Drink to Cart' button. In the center, a 'Cart' section shows a list with 'Special Burger' and 'Coke', with 'Coke' highlighted in blue. Below the cart list is a 'Cancel Cart' button. At the bottom of the cart section, the 'Total Price: RM7.00' is displayed.

Next, customers need to insert the delivery information. After done, click submit order and the order information will appear. The error will appear and can't submit the order if customers not fulfill the fields.

The screenshot shows the 'Order Confirmation' dialog box overlaid on the main application. The dialog box contains the following text: 'Your order has been received! Thank you for ordering with us!', 'Menu Ordered: Special Burger, Coke', 'Total Price: RM7.00', 'Delivery Date: 1/4/24', 'Delivery Time: 15:15', 'Delivery Location: perpustakaan badlishah UITM KEDAH', 'Additional Notes: -', and 'Please get ready the amount, we will call you before delivery to make sure you able to pickup the order.' with an 'OK' button. In the background, the 'Delivery Information' form is visible, showing 'Delivery Location: adlishah UITM KEDAH', 'Delivery Date: 1/4/24', 'Delivery Time: 15:15', 'Additional Notes: -', and a 'Submit Order' button.

THE SNAPSHOT OF DATABASE :

phpMyAdmin

Server: 127.0.0.1 » Database: online_food_ordering » Table: customer_order

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

customer_name	phone_number	Cart	Total_price	delivery_location	delivery_date	delivery_time_hours	delivery_time_minutes	additional_notes
			0		1/3/24	0	0	
			0		1/3/24	0	0	
			0		1/3/24	0	0	
MUHAMMAD AMMAR BIN KHAMISAN	0134259876	Special Burger, Orange Juice		lobi kolej malinja	1/3/24	11		10 taknak sayur
HANIF AIMAN	0149307134	Special Burger, Lemon Tea, Sprite	10.0	lobi malinja	1/4/24	13		25 taknak banyak sos
KHAIRUL ANUAR	0143224578	King Burger, Orange Juice, Orange Jul...	20.0	pejabat hep uitm kedah	1/4/24	13		10 -
KHAIRUL ANUAR	0143224578	King Burger, Orange Juice, Orange	20.0	pejabat hep uitm kedah	1/4/24	13		10 -

Console

phpMyAdmin

Server: 127.0.0.1 » Database: online_food_ordering » Table: customer_order

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

customer_name	phone_number	Cart	Total_price	delivery_location	delivery_date	delivery_time_hours	delivery_time_minutes	additional_notes
AMIR YAHYA	0132453678	Orange Jul... Demon Spicy Burger, Lemon Tea	23.0	lobi kolej malinja				10 tanak sos banyak
Samad Haji Ayob	0129556178	Special Burger, Luxury Golden Burger, Sprite	28.0	Pejabat HEP, UITM Kedah	1/3/24	14		15 dah sampai call ya
Hazim Ahmad	0175437231	King Burger, Orange Juice, Orange Juice	13.0	Foodcourt Mahsuri, UITM Kedah	1/4/24	16		45 nak abang yang hensem yang hantar

☐ Show all | Number of rows: 25 | Filter rows: Search this table

- Click to sort results by this column.
 - Shift+Click to add this column to ORDER BY clause or to toggle ASC/DESC.
 - Ctrl+Click or Alt+Click (Mac): Shift+Option+Click to remove column from ORDER BY clause