



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

DIPLOMA IN LIBRARY INFORMATIC (CDIM144)

IML208: PROGRAMMING FOR LIBRARIES

**INDIVIDUAL ASSIGNMENT:
“RESERVATION SYSTEM”**

**PREPARED BY:
PUTERI NUR SYAZLEEN BINTI MOHD FAUDZY (2022877382)
GROUP CDIM1443F**

**PREPARED FOR:
MR. AIRUL SHAZWAN BIN NORSHAHIMI**

**SUBMISSION DATE:
4 JANUARY 2024**

**INDIVIDUAL ASSIGNMENT:
“RESERVATION SYSTEM”**

PREPARED BY:

PUTERI NUR SYAZLEEN BINTI MOHD FAUDZY (2022877382)

GROUP CDIM1443F

CDIM144 – DIPLOMA IN LIBRARY INFORMATICS

**SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
UNIVERSITI TEKNOLOGI MARA (UITM)
KEDAH BRANCH**

TABLE OF CONTENT

1.0	INTRODUCTION.....	1
2.0	FLOWCHART.....	2
3.0	PYTHON CODE.....	3
4.0	GRAPHICAL USER INTERFACES (GUI).....	5
5.0	DATABASE.....	6
6.0	CONCLUSION.....	7

1.0 INTRODUCTION

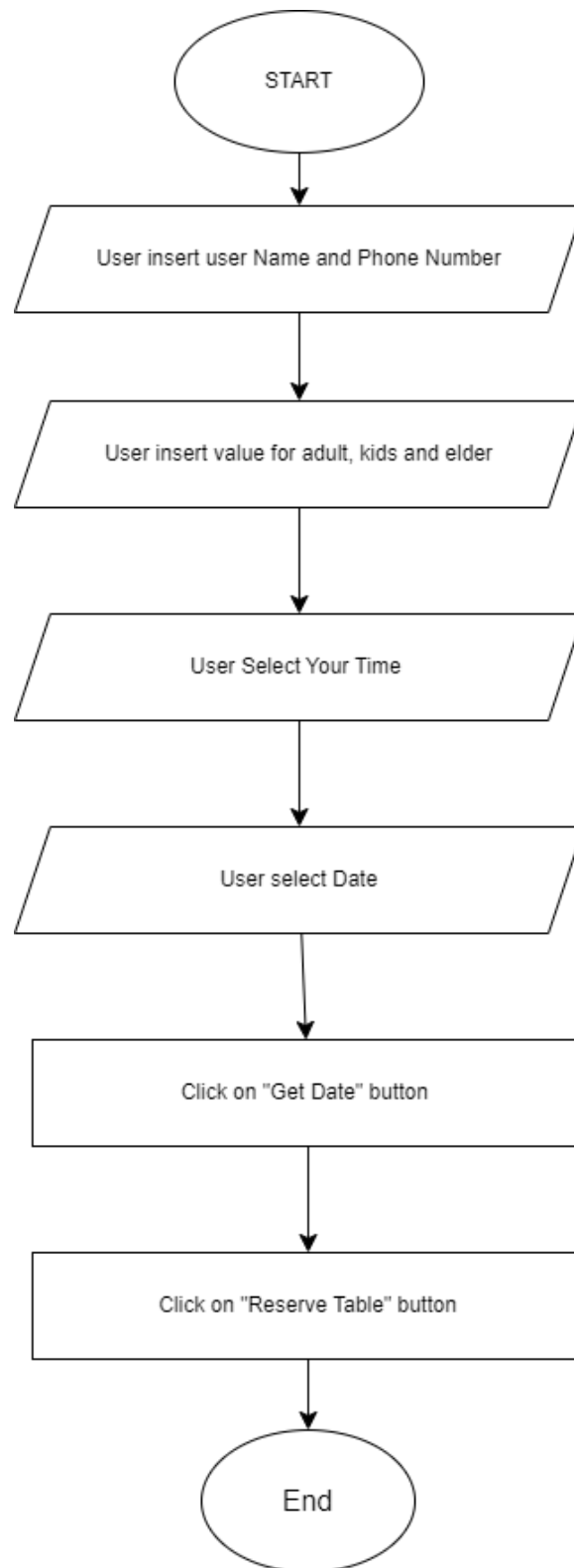
I selected to perform a system coding python for a buffet restaurant reservation system for this assignment. The system is a basic Python programme application constructed with Tkinter for the graphical user interface (GUI) and MySQL Connector to connect the Python programme to the database. required the user to enter a few details in order to make a reservation at the shop.

The user's name and phone number are entered into fields on the GUI. Users are then asked to provide a number for adult, child, and elder. To proceed with the booking process, time and date must be chosen. Finally, users must click the "reserved table" option. The computation process will begin immediately after consumers press the button. There are three price categories: adults (RM60), children (RM30), and seniors (RM40). As a result, it is critical for users to provide a number for each person while following the categories to obtain the correct price. Because the restaurant will open at 4PM and close at 10PM, the time possibilities are limited to 4PM to 9PM. After entering all of the information and clicking the "reserved table" button, the user has successfully made a reservation.

So, for database interaction, the system constructs an interactive cursor after connecting to the MySQL database. The information is then inserted into the "reserva_buffet" table, which contains the name, phone_number, num_adult, num_kids, num_elder, time, date, and total cost. The system will print "Data Inserted Successfully" once you insert the data into the reserva_buffet table in MySQL Connection. As a result, database interactions are handled by implementing suitable error handling and printing appropriate messages to the terminal.

Finally, the Buffet Reservation System provides the structure for reservation data, computing total cost, and storing data in a MySQL database. It serves as a medium for users to address individual needs and increase the overall standard of usability. Aside from that, it could help to make it easier for consumers to make reservations through the system without having to phone or walk into the restaurant.

2.0 FLOWCHART



3.0 PYTHON CODE

```
ind_asspy > ...
1 import tkinter as tk
2 from tkinter import messagebox
3 import mysql.connector
4 from tkcalendar import Calendar
5
6
7 # Constants
8 NUM_TABLES = 10
9 table_status = [False] * NUM_TABLES # False represents an available table
10 user_info = []
11
12 def reserve_table():
13     # Collect user information
14     name = name_entry.get()
15     phone_number = phone_entry.get()
16     num_adults = int(adults_entry.get())
17     num_kids = int(kids_entry.get())
18     num_elders = int(elders_entry.get())
19     time_reserve = time_var.get()
20     date_reserve = cal.get_date()
21
22     # Calculate total cost
23     total_cost = (num_adults * 60.00) + (num_kids * 30.00) + (num_elders * 40.00)
24
25     # Display reservation information
26     reservation_info = {
27         f'Name: {name}\n'
28         f'Phone Number: {phone_number}\n'
29         f'Adults: {num_adults}\n'
30         f'Kids: {num_kids}\n'
31         f'Elders/OK: {num_elders}\n'
32         f'Total Cost: RM {total_cost:.2f}'
33     }
34
35     messagebox.showinfo("Success", reservation_info)
36
37
38 # Update table status and user information
39 user_info.append({
40     'name': name,
41     'phone_number': phone_number,
42     'num_adults': num_adults,
43     'num_kids': num_kids,
44     'num_elders': num_elders,
45     'total_cost': total_cost,
46     'time_reserve': time_reserve,
47     'date_reserve': date_reserve
48 })
49
50 # To print specific value in a dictionary ( list )
51 # print(user_info[0]['name'])
52 # print(user_info[0])
53
54 #-----TESTING-----
55
56 # Connect to your MySQL database
57 mydb = mysql.connector.connect(
58     host="localhost",
59     user="root",
60     password="",
61     database="reserva_buffet"
62 )
63
64 # Create a cursor object to interact with the database
65 cursor = mydb.cursor()
66
67 # Inserting data into a table
68 sql = "INSERT INTO 'customer_details' (NAME, PHONE_NUM, NUM_ADULT, NUM_KIDS, NUM_ELDER, TIME, DATE, TOTAL_COST) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
69 val = (name_entry.get(), phone_entry.get(), adults_entry.get(), kids_entry.get(), elders_entry.get(), time_var.get(), cal.get_date(), total_cost())
70 sql = "INSERT INTO 'customer_details' (NAME, PHONE_NUM, NUM_ADULT, NUM_KIDS, NUM_ELDER, TIME, DATE, TOTAL_COST) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
71 val = (user_info[0]['name'], user_info[0]['phone_number'], user_info[0]['num_adults'], user_info[0]['num_kids'], user_info[0]['num_elders'], user_info[0]['time_re
72
73
74 ind_asspy > reserve_table
75 val = (user_info[0]['name'], user_info[0]['phone_number'], user_info[0]['num_adults'], user_info[0]['num_kids'], user_info[0]['num_elders'], user_info[0]['time_re
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

107 < label = tk.Label(root, text= "DUCKY BUFFET", font= ("Cambria", 20, "italic"), bg= "light goldenrod",
108 fg= "dark goldenrod", bd= 8, relief= "groove")
109 label.pack(ipadx= 800)
110
111 # User Information Entry Widgets
112 name_label = tk.Label(root, text= "Name:", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 245)
113 name_label.pack()
114 name_entry = tk.Entry(root)
115 name_entry.pack ( ipadx= 200)
116
117
118
119 phone_label = tk.Label(root, text= "Phone Number:", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 220 )
120 phone_label.pack()
121 phone_entry = tk.Entry(root)
122 phone_entry.pack(ipadx= 200)
123
124 adults_label = tk.Label(root, text= "Number of Adults:", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 215)
125 adults_label.pack()
126 adults_entry = tk.Entry(root)
127 adults_entry.pack(ipadx= 200)
128
129 kids_label = tk.Label(root, text= "Number of Kids:", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 220 )
130 kids_label.pack()
131 kids_entry = tk.Entry(root)
132 kids_entry.pack(ipadx= 200)
133
134 elders_label = tk.Label(root, text= "Number of Elders/OKU:", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 200)
135 elders_label.pack()
136 elders_entry = tk.Entry(root)
137 elders_entry.pack(ipadx= 200)
138
139 time_label = tk.Label(root, text= "Time: (in 24 hours)", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 215)
140 time_label.pack()
141
142 # Trip Type Dropdown

```

```

143
144 # Create the list of options
145 time_list = ["4PM", "5PM", "6PM", "7PM", "8PM", "9PM"]
146
147 time_var = tk.StringVar(root)
148 time_var.set("Select Your Time") # Default value before your selection
149 time_dropdown = tk.OptionMenu(root, time_var, *time_list)
150
151 time_dropdown.pack(pady= 10, ipadx= 50)
152
153 date_label = tk.Label(root, text= "Date: (YY-MM-DD)", fg= "salmon4", bd= 5, relief= "ridge", bg= "light goldenrod", padx= 215)
154 date_label.pack()
155
156 # Add Calendar
157 < cal = Calendar(root, selectmode= 'day',
158 | | | year = 2023, month = 1,
159 | | | day = 22)
160
161 cal.pack(pady = 5)
162
163
164 < def grad_date():
165 | date.config(text = "Selected Date is: " + cal.get_date())
166
167 # Add Button and Label
168 < tk.Button(root, text = "Get Date",
169 | | | command = grad_date).pack(pady = 3)
170
171 date = tk.Label(root, text = "")
172 date.pack(pady = 1)
173
174 # Table Listbox
175 < table_listbox = tk.Listbox(root, selectmode= tk.SINGLE, height= NUM_TABLES)
176 < table_listbox.pack(pady= 10, ipadx= 10)
177
178 # Reserve Button
179 reserve_button = tk.Button(root, text= "Reserve Table", command= reserve_table)
180 reserve_button.pack(pady= 5, ipadx= 5)
181

```

```

181
182 # Update Table listbox
183 < update_table_listbox()
184
185 # Run the main loop
186 root.mainloop()
187

```

4.0 GRAPHICAL USER INTERFACES (GUI)

The screenshot displays a graphical user interface for 'DUCKY BUFFET'. The window title is 'DUCKY BUFFET'. The interface features a yellow header bar with the restaurant's name. Below this, there are five yellow input fields for 'Name:', 'Phone Number:', 'Number of Adults:', 'Number of Kids:', and 'Number of Elders/OKU:'. A 'Time:' field is followed by a 'Select Your Time' dropdown menu. Below these is a 'Date:' field and a calendar widget for January 2023. The calendar shows the date '22' selected. At the bottom, there are two buttons: 'Get Date' and 'Reserve Table'.

DUCKY BUFFET

Name:

Phone Number:

Number of Adults:

Number of Kids:

Number of Elders/OKU:

Time:

Select Your Time

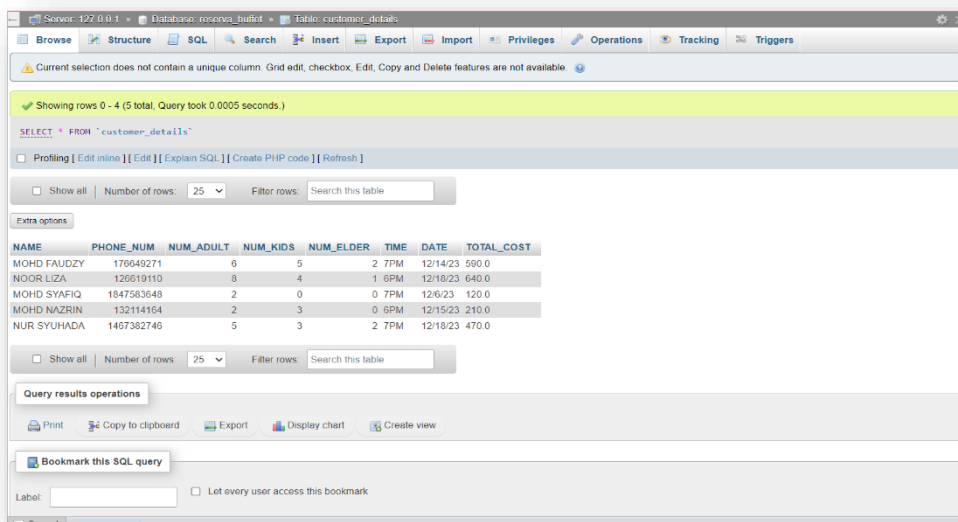
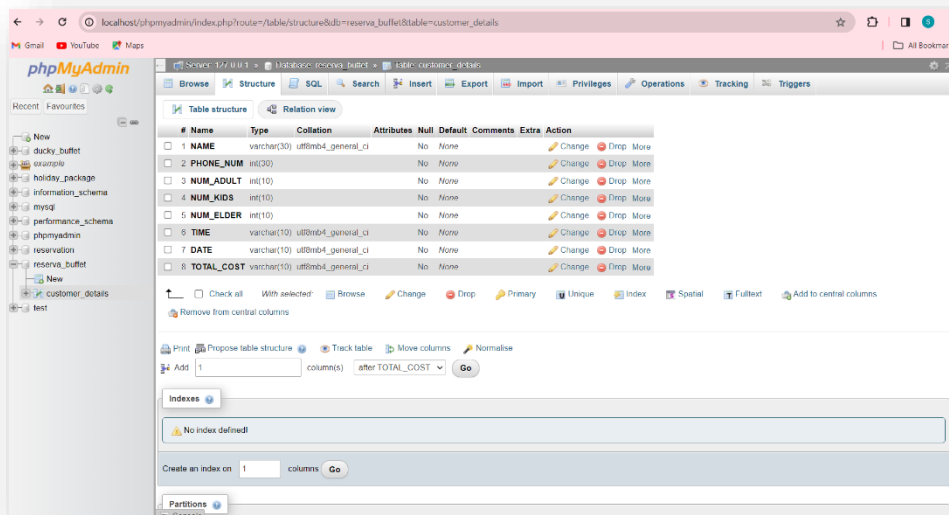
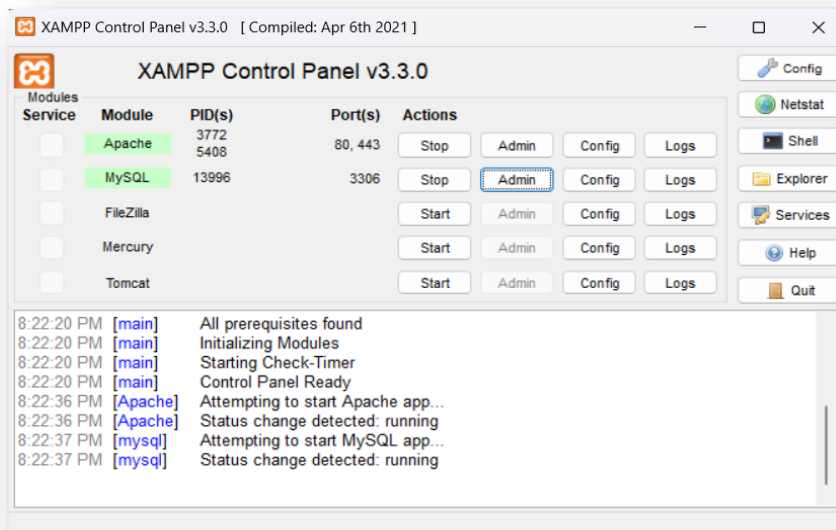
Date:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
52	26	27	28	29	30	31	1
1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22
4	23	24	25	26	27	28	29
5	30	31	1	2	3	4	5

Get Date

Reserve Table

5.0 DATABASE



6.0 CONCLUSION

Finally, the Reservation System is a well-designed Python programme with a graphical user interface (GUI) constructed using Tkinter and coupled with a MySQL database through the MySQL Connector. The system processes user inputs for user reservation data effectively, does real-time total cost calculations, and stores the data in a MySQL database table named "reserva_buffet." The system implements appropriate error handling during database interactions to ensure proper data entry into the database. Feedback techniques such as the "Data Entered Successfully" message improve system reliability. Overall, the Reservation System provides a solid foundation for managing database transactions, total calculations, and data organisation.

To summarise, I learned a lot of new things about how to code a system in Python during this assignment. I developed a new way to give commands, and there are numerous ways to construct and use them in your system. It is critical to conduct additional research and exploration in order to get more knowledge. Furthermore, this job taught me how to be more patient and persistent. Aside from that, I'd want to thank my classmates and lecturer, Sir Airul, for their assistance and guidance while I attempted to accomplish my task.