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 INFORMATIC

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

TABLE OF CONTENT

1.0 INTRODUCTION

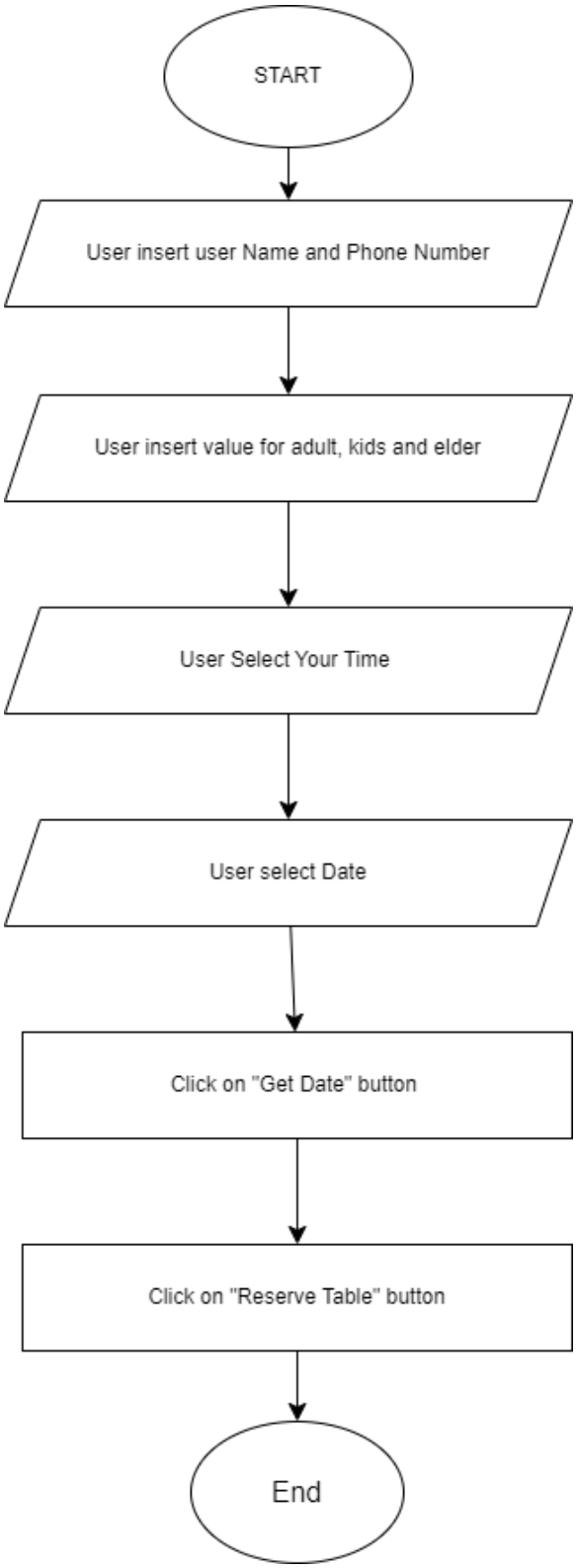
I selected to perform a system coding python for a buffet restaurant reservation system for this assignment. The system is a basic Python programme application constructed with Tkinter for the graphical user interface (GUI) and MySQL Connector to connect the Python programme to the database. required 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_ass.py > ...

```python
import tkinter as tk
from tkinter import messagebox
import mysql.connector
from tkcalendar import Calendar


# Constants
NUM_TABLES = 10
table_status = [False] * NUM_TABLES  # False represents an available table
user_info = []

def reserve_table():
    # Collect user information
        name = name_entry.get()
        phone_number = phone_entry.get()
        num_adults = int(adults_entry.get())
        num_kids = int(kids_entry.get())
        num_elders = int(elders_entry.get())
        time_reserve = time_var.get()
        date_reserve = cal.get_date()

    # Calculate total cost
        total_cost = (num_adults * 60.00) + (num_kids * 30.00) + (num_elders * 40.00)

        # Display reservation information
        reservation_info = (
            f"Name: {name}\n"
            f"Phone Number: {phone_number}\n"
            f"Adults: {num_adults}\n"
            f"Kids: {num_kids}\n"
            f"Elders/OKU: {num_elders}\n"
            f"Total Cost: RM {total_cost:.2f}"
        )

        messagebox.showinfo("Success", reservation_info)
```

ind_ass.py > ...

```python
        # Update table status and user information
        user_info.append({
            'name': name,
            'phone_number': phone_number,
            'num_adults': num_adults,
            'num_kids': num_kids,
            'num_elders': num_elders,
            'total_cost': total_cost,
            'time_reserve': time_reserve,
            'date_reserve': date_reserve
        })

        #To print specific value in a dictionary ( list )
        #print(user_info[0]['name'])
        #print(user_info[0])

        #----------------------TESTING---------------

        # Connect to your MySQL database
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="reserva_buffet"
        )

        # Create a cursor object to interact with the database
        cursor = mydb.cursor()

        # Inserting data into a table
        #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)"
        #val = (name_entry.get(), phone_entry.get(), adults_entry.get(), kids_entry.get(), elders_entry.get(), time_var.set(), cal(), total_cost())
        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)"
        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
```

ind_ass.py > reserve_table

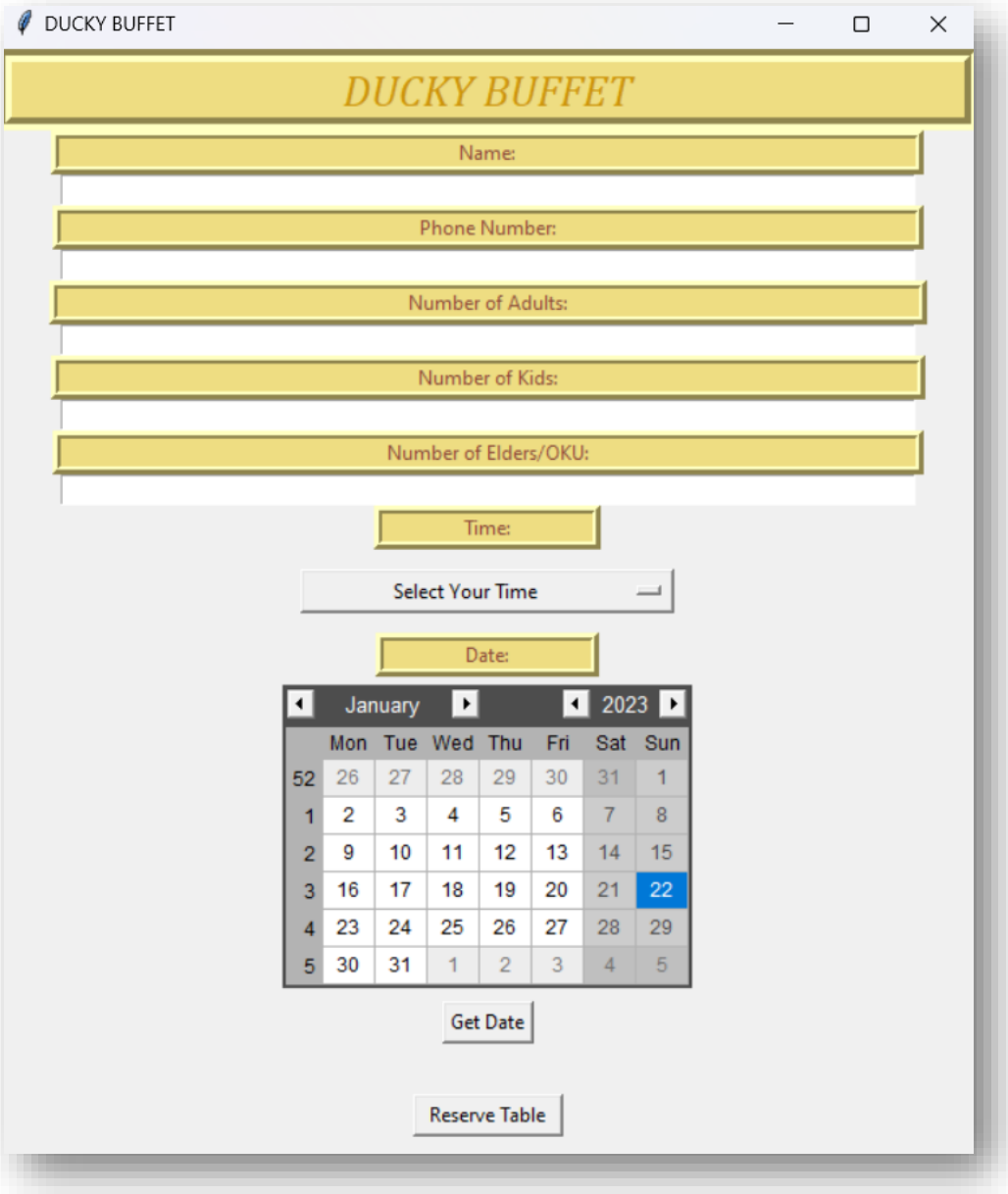
```python
        val = (user_info[0]['name'], user_info[0]['phone_number'], user_info[0]['num_adults'], user_info[0]['num_kids'], user_info[0]['num_elders'], user_info[0]['time_re
        try:
            cursor.execute(sql, val)
            mydb.commit()
            print("Data inserted successfully!")
        except mysql.connector.Error as err:
            print(f"Error: {err}")
            mydb.rollback()

        clear_user_info()

# ''' -> Comment for Multiple Line
'''
def update_table_listbox():
    table_listbox.delete(0, tk.END)
    for i, status in enumerate(table_status):
        if status:
            table_listbox.insert(tk.END, f"Table {i + 1} (Reserved)")
        else:
            table_listbox.insert(tk.END, f"Table {i + 1} (Available)")
'''

def clear_user_info():
    name_entry.delete(0, tk.END)
    phone_entry.delete(0, tk.END)
    adults_entry.delete(0, tk.END)
    kids_entry.delete(0, tk.END)
    elders_entry.delete(0, tk.END)
    time_var.set("Select Your Time")  # Default value before your selection
    date.config(text = "No Date Selected")
    user_info.clear()

# GUI setup
root = tk.Tk()
root.title("DUCKY BUFFET")
root.geometry ('600x680')

label= tk.Label(root,text= "DUCKY BUFFET",font=("Cambria", 20, "italic"), bg= "light goldenrod",
```

```
ind_ass.py > reserve_table
label= tk.Label(root,text= "DUCKY BUFFET",font=("cambria", 20, "italic"), bg= "light goldenrod",
                fg="dark goldenrod3", bd=8, relief="groove")
label.pack(ipadx=800)


# User Information Entry Widgets
name_label = tk.Label(root, text="Name:", fg="salmon4",bd=5, relief="ridge", bg="light goldenrod", padx=245)
name_label.pack()
name_entry = tk.Entry(root)
name_entry.pack ( ipadx=200)


phone_label = tk.Label(root, text="Phone Number:", fg="salmon4",bd=5, relief="ridge", bg="light goldenrod", padx=220 )
phone_label.pack()
phone_entry = tk.Entry(root)
phone_entry.pack(ipadx=200)

adults_label = tk.Label(root, text="Number of Adults:", fg="salmon4", bd=5, relief="ridge", bg="light goldenrod", padx=215)
adults_label.pack()
adults_entry = tk.Entry(root)
adults_entry.pack(ipadx=200)

kids_label = tk.Label(root, text="Number of Kids:", fg="salmon4", bd=5, relief="ridge", bg="light goldenrod", padx=220 )
kids_label.pack()
kids_entry = tk.Entry(root)
kids_entry.pack(ipadx=200)

elders_label = tk.Label(root, text="Number of Elders/OKU:", fg="salmon4",  bd=5, relief="ridge", bg="light goldenrod", padx=200)
elders_label.pack()
elders_entry = tk.Entry(root)
elders_entry.pack(ipadx=200)

time_label = tk.Label(root, text="Time: (in 24 hours)", fg="salmon4",  bd=5, relief="ridge", bg="light goldenrod", padx=215)
time_label.pack()

# Trip Type Dropdown
```

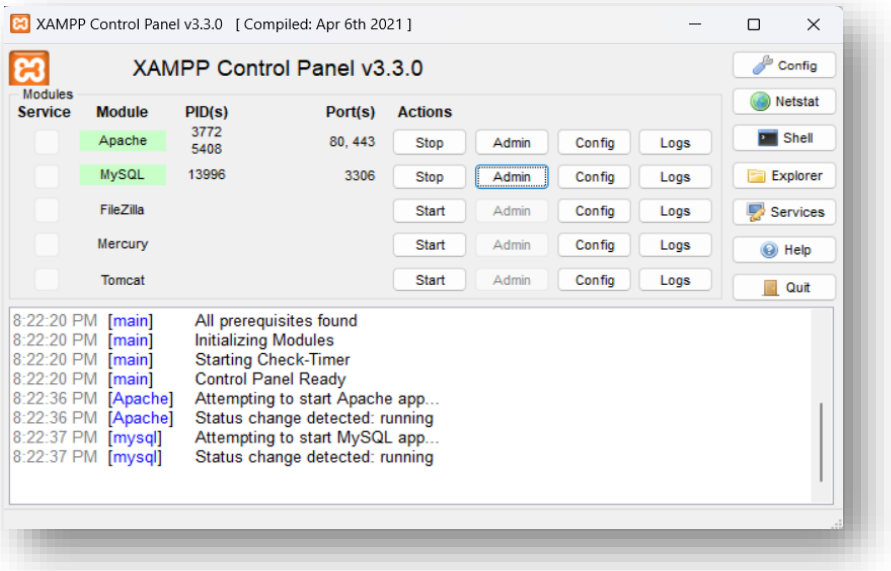
```
ind_ass.py > reserve_table

# Create the list of options
time_list = ["4PM", "5PM", "6PM", "7PM", "8PM", "9PM"]

time_var = tk.StringVar(root)
time_var.set("Select Your Time")  # Default value before your selection
time_dropdown = tk.OptionMenu(root, time_var, *time_list)

time_dropdown.pack(pady=10, ipadx=50)

date_label = tk.Label(root, text="Date:(YY-MM-DD)",fg="salmon4", bd=5, relief="ridge", bg="light goldenrod", padx=215)
date_label.pack()


# Add Calendar
cal = Calendar(root, selectmode = 'day',
               year = 2023, month = 1,
               day = 22)

cal.pack(pady = 5)

def grad_date():
    date.config(text = "Selected Date is: " + cal.get_date())

# Add Button and Label
tk.Button(root, text = "Get Date",
       command = grad_date).pack(pady = 3)

date = tk.Label(root, text = "")
date.pack(pady =1)

# Table Listbox
# table_listbox = tk.Listbox(root, selectmode=tk.SINGLE, height=NUM_TABLES)
# table_listbox.pack(pady=10, ipadx=10)

# Reserve Button
reserve_button = tk.Button(root, text="Reserve Table", command=reserve_table)
reserve_button.pack(pady=5, ipadx=5)
```

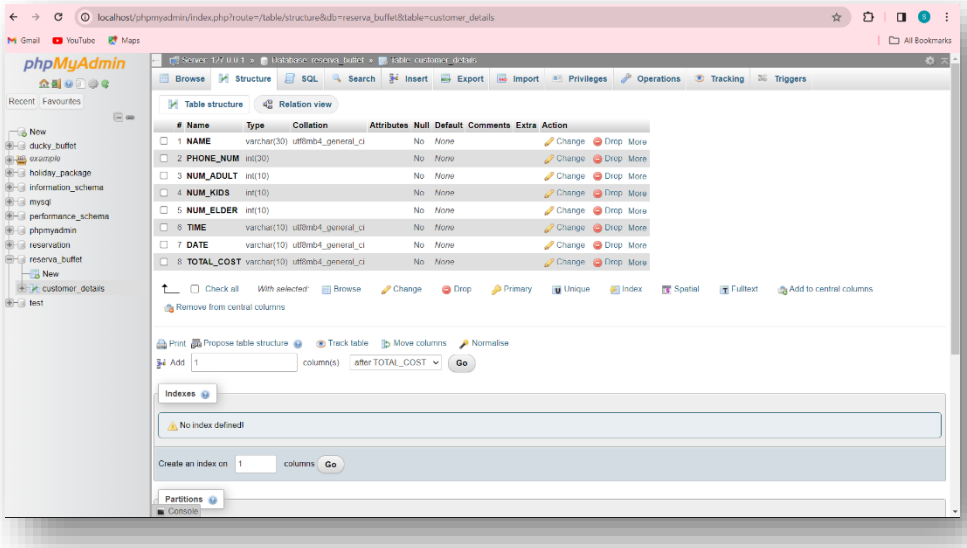
```

# Update Table Listbox
# update_table_listbox()

# Run the main loop
root.mainloop()

```

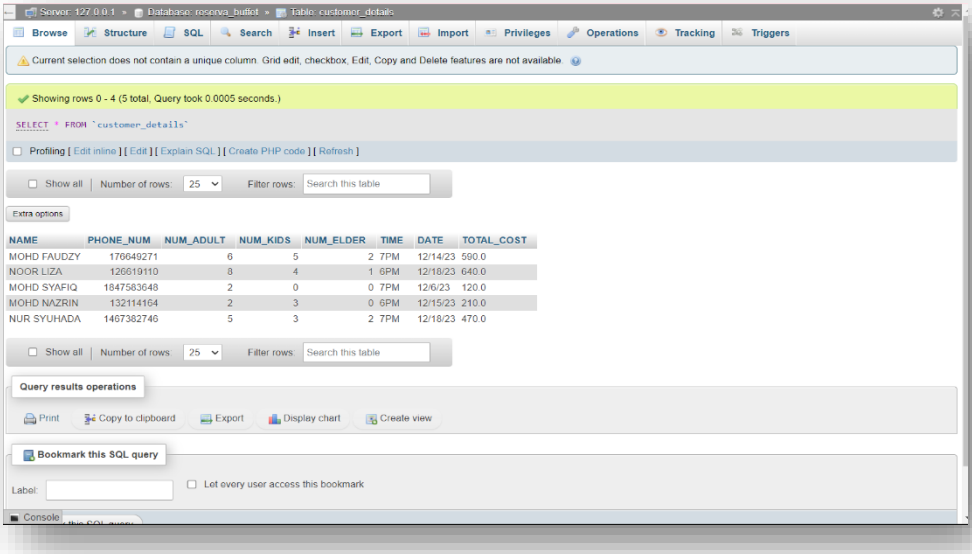
4.0 GRAPHICAL USER INTERFACES (GUI)



5.0 DATABASE


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
3772
5408
80, 443
Stop
Admin
Config
Logs
MySQL
13996
3306
Stop
Admin
Config
Logs
FileZilla
Start
Admin
Config
Logs
Mercury
Start
Admin
Config
Logs
Tomcat
Start
Admin
Config
Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
8:22:20 PM [main] All prerequisites found
8:22:20 PM [main] Initializing Modules
8:22:20 PM [main] Starting Check-Timer
8:22:20 PM [main] Control Panel Ready
8:22:36 PM [Apache] Attempting to start Apache app...
8:22:36 PM [Apache] Status change detected: running
8:22:37 PM [mysql] Attempting to start MySQL app...
8:22:37 PM [mysql] Status change detected: running


phpMyAdmin
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Table structure
Relation view
NAME
PHONE_NUM
NUM_ADULT
NUM_KIDS
NUM_ELDER
TIME
DATE
TOTAL_COST
No index defined!
Create an index on
columns
Go
Partitions


Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)
SELECT * FROM `customer_details`
NAME PHONE_NUM NUM_ADULT NUM_KIDS NUM_ELDER TIME DATE TOTAL_COST
MOHD FAUDZY 176649271 6 5 2 7PM 12/14/23 590.0
NOOR LIZA 126619110 8 4 1 6PM 12/18/23 640.0
MOHD SYAFIQ 1847583648 2 0 0 7PM 12/6/23 120.0
MOHD NAZRIN 132114164 2 3 0 6PM 12/15/23 210.0
NUR SYUHADA 1467382746 5 3 2 7PM 12/18/23 470.0
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark

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.