COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS,
UNIVERSITI TEKNOLOGI MARA,
MERBOK, KEDAH

DIPLOMA IN LIBRARY INFORMATICS (CDIM144)

PROGRAMMING FOR LIBRARIES (IML207)

“COURIER SYSTEM: PARCEL BILL”

PREPARED BY:

SITI AISYAH NATASHA BINTI SOPHIAN (2022677088)

CLASS: KCDIM1443F

PREPARED FOR:
SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE:

WEEK 12

“COURIER SYSTEM: PARCEL BILL

PREPARED BY:

SITI AISYAH NATASHA BINTI SOPHIAN (2022677088)

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS,
UNIVERSITI TEKNOLOGI MARA,
MERBOK, KEDAH

SUBMISSION DATE:

Week 12

TABLE OF CONTENT

اونيۏرسيتي تيكنولوݢي مارا
UNIVERSITI
TEKNOLOGI
MARA

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 : SITI AISYAH NATASHA BINTI SOPHIAN
Matric Number : 2022677088
Course Code : IML208
Programme Code :-
Faculty / Campus : UiTM Kampus Sungai Petani

Bahagian Pentaksiran & Penilaian Akademik
0404/2021

ACKNOWLEDGEMENTS

First and foremost, I would like to express my greatest gratitude to Mr Airul Shazwan for his guidance throughout this journey. I was given guidelines on how to do this individual assignment in subject Programming for Libraries for my third semester of Diploma in Information Management. A special gratitude to my respected lecturer for giving me a very clear and concise instructions to make it easier for me to proceed this report smoothly, also assisted on encouragement and guidance to finally make this assignment succeed without exceeding its due date.

A very huge thank you to my beloved classmates in KCDIM1443F class for giving each other cooperation and support while I was working on this assignment for this subject.

For that matter, I am able to try to put my best effort throughout the completion of this assignment. Once again, I would like to thank everyone for the time, energy and commitment.

1.0 INTRODUCTION OF COURIER SYSTEM

The Courier Parcel Bill System is a software solution designed to streamline and automate the process of calculating charges for courier services. In order for courier businesses to run their businesses successfully and keep their users satisfied in the fast-paced world of today, accurate and efficient billing systems are essential.

1.1 PROBLEM STATEMENT

Manual and prone to error parcel bill calculation methods are a major source of operational inefficiencies, billing disparities, and disgruntled customers in the current courier industry. When a courier parcel bill calculation system isn't automated, the following happens:

- **Inaccuracies and Disputes**: Determining shipping costs by hand raises the possibility of errors, which in turn causes billing disputes and erodes customer confidence and satisfaction.

- **Lack of Adaptability**: Without a centralized system, courier companies struggle to adapt to dynamic market conditions, and evolving customer demands, limiting their ability to stay competitive in the industry.

1.2 OBJECTIVE

- **Automation:** Streamline and automate the complex process of calculating courier parcel bills to reduce manual errors and enhance operational efficiency.

- **Adaptability:** Provide a flexible and customizable system that adapts to various pricing structures, and shipping options, meeting the diverse needs of courier companies.

2.0 FLOW CHART

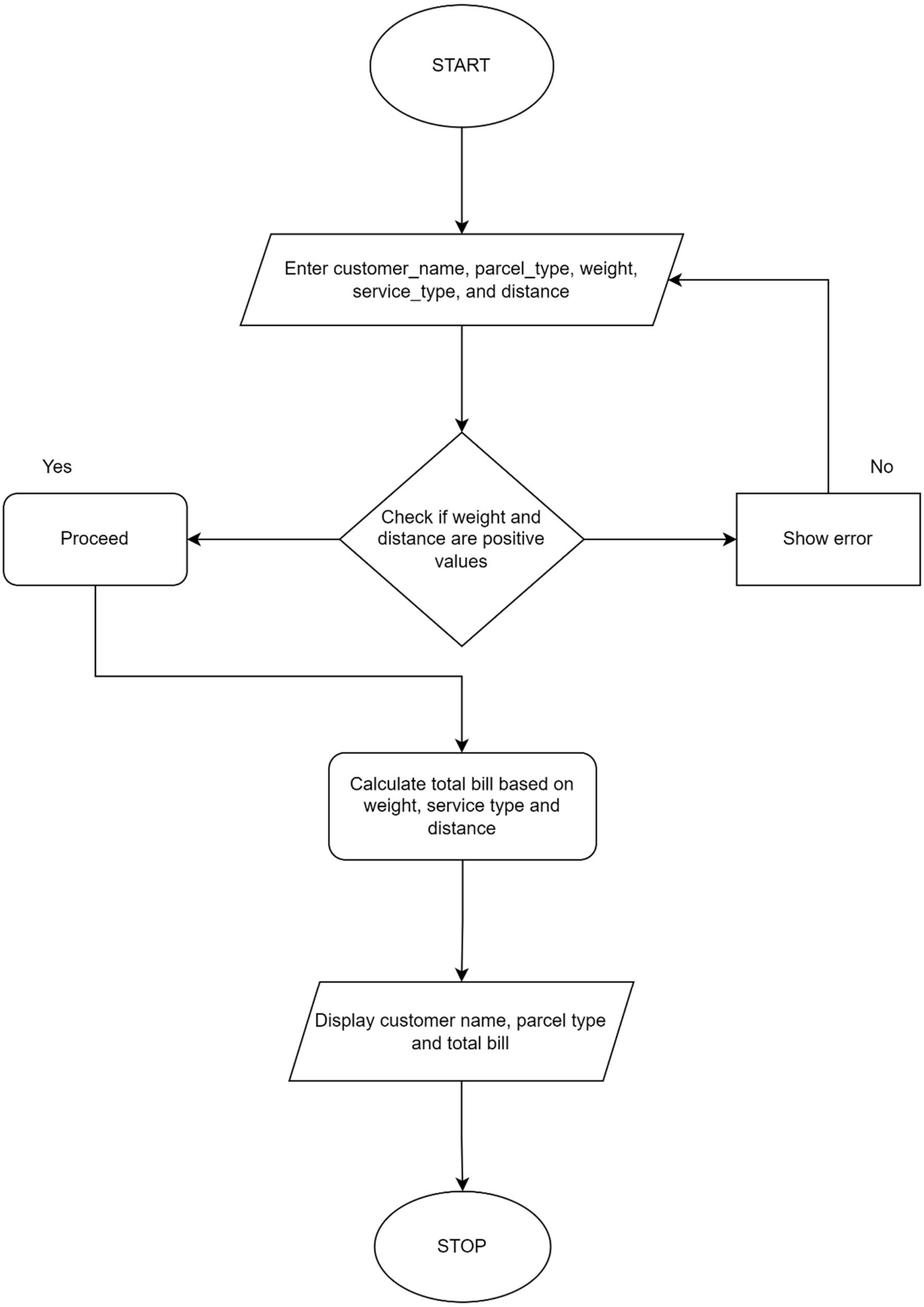
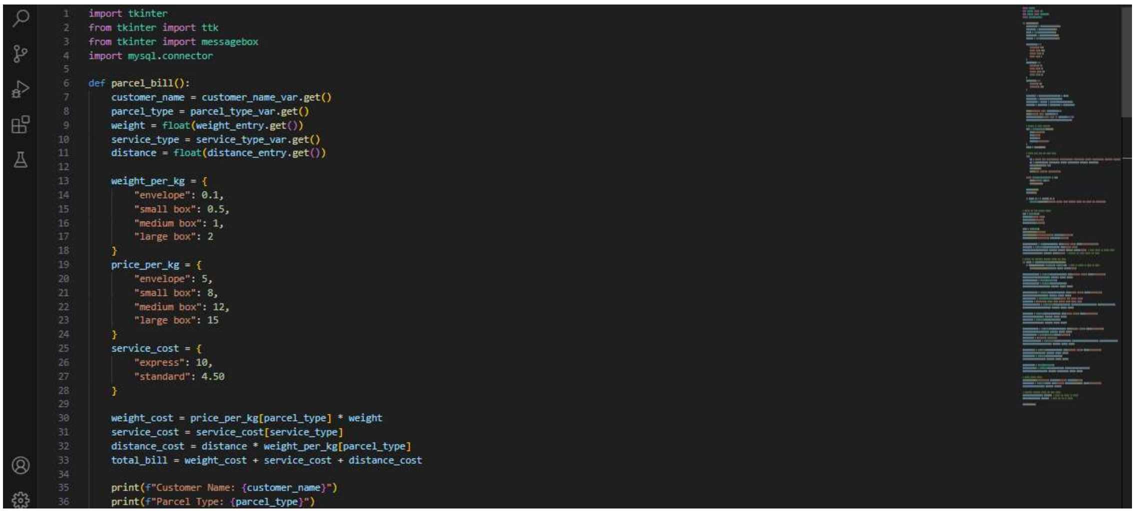
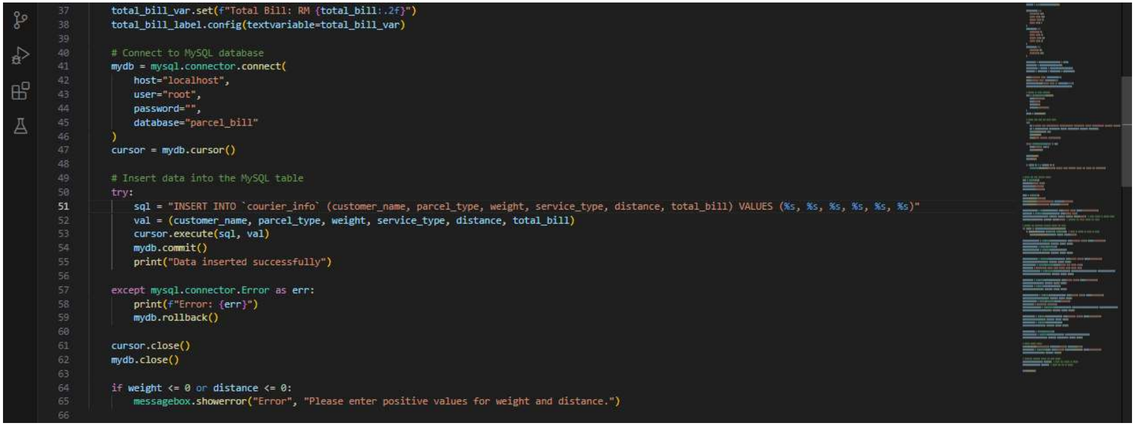

START
Enter customer_name, parcel_type, weight, service_type, and distance
Yes
No
Proceed
Check if weight and distance are positive values
Show error
Calculate total bill based on weight, service type and distance
Display customer name, parcel type and total bill
STOP

Figure 1 Parcel bill Flowchart

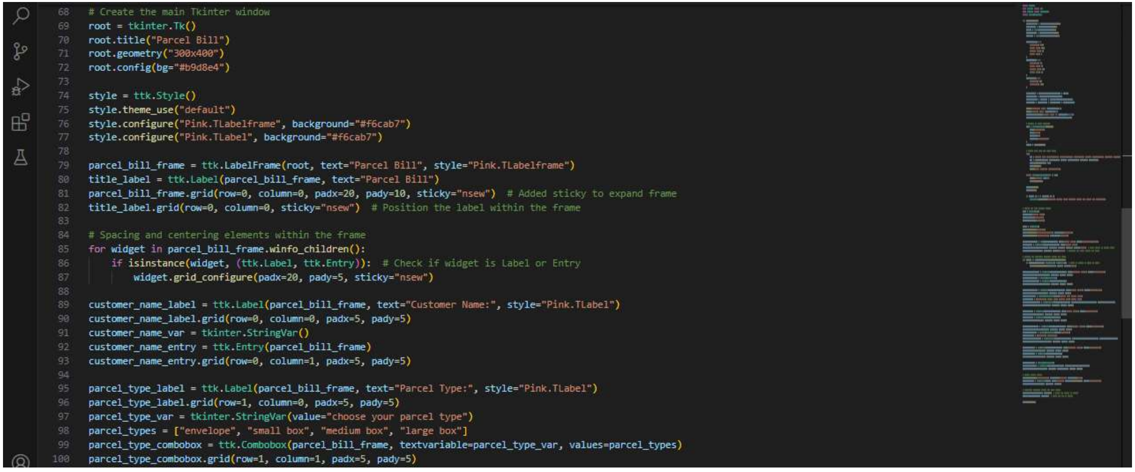
3.0 SNAPSHOT OF CODE



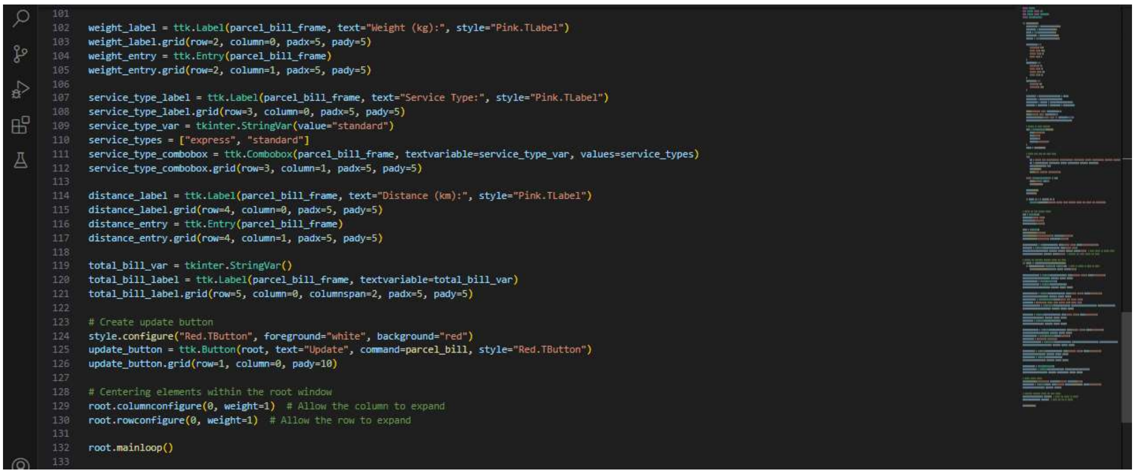
```python
import tkinter
from tkinter import ttk
from tkinter import messagebox
import mysql.connector

def parcel_bill():
    customer_name = customer_name_var.get()
    parcel_type = parcel_type_var.get()
    weight = float(weight_entry.get())
    service_type = service_type_var.get()
    distance = float(distance_entry.get())

    weight_per_kg = {
        "envelope": 0.1,
        "small box": 0.5,
        "medium box": 1,
        "large box": 2
    }
    price_per_kg = {
        "envelope": 5,
        "small box": 8,
        "medium box": 12,
        "large box": 15
    }
    service_cost = {
        "express": 10,
        "standard": 4.50
    }

    weight_cost = price_per_kg[parcel_type] * weight
    service_cost = service_cost[service_type]
    distance_cost = distance * weight_per_kg[parcel_type]
    total_bill = weight_cost + service_cost + distance_cost

    print(f"Customer Name: {customer_name}")
    print(f"Parcel Type: {parcel_type}")
```



```python
    total_bill_var.set(f"Total Bill: RM {total_bill:.2f}")
    total_bill_label.config(textvariable=total_bill_var)

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

    # Insert data into the MySQL table
    try:
        sql = "INSERT INTO `courier_info` (customer_name, parcel_type, weight, service_type, distance, total_bill) VALUES (%s, %s, %s, %s, %s, %s)"
        val = (customer_name, parcel_type, weight, service_type, distance, total_bill)
        cursor.execute(sql, val)
        mydb.commit()
        print("Data inserted successfully")

    except mysql.connector.Error as err:
        print(f"Error: {err}")
        mydb.rollback()

    cursor.close()
    mydb.close()

    if weight <= 0 or distance <= 0:
        messagebox.showerror("Error", "Please enter positive values for weight and distance.")

```



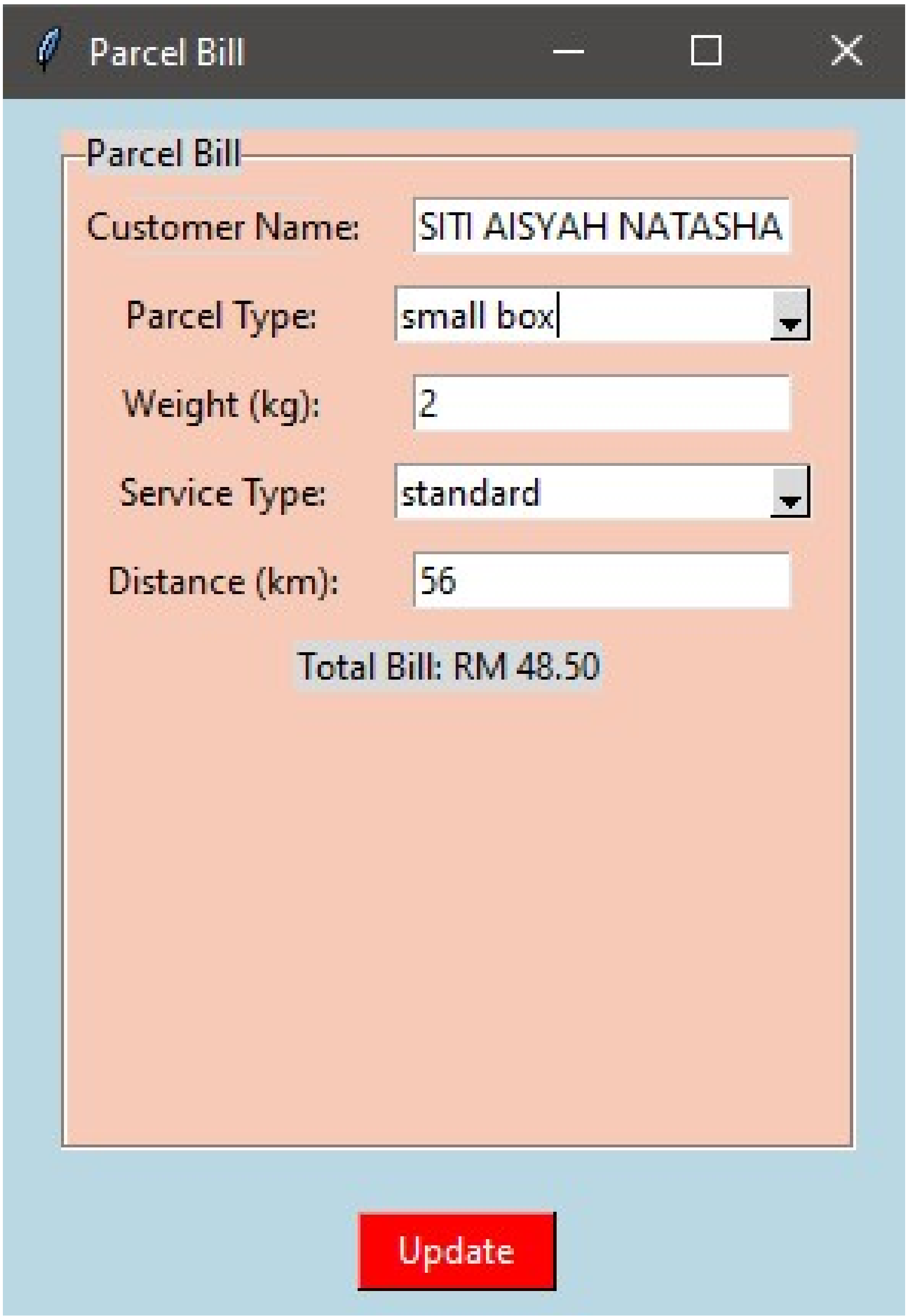
```python
# Create the main Tkinter window
root = tkinter.Tk()
root.title("Parcel Bill")
root.geometry("300x400")
root.config(bg="#b9d8e4")

style = ttk.Style()
style.theme_use("default")
style.configure("Pink.TLabelframe", background="#f6cab7")
style.configure("Pink.TLabel", background="#f6cab7")

parcel_bill_frame = ttk.LabelFrame(root, text="Parcel Bill", style="Pink.TLabelframe")
title_label = ttk.Label(parcel_bill_frame, text="Parcel Bill")
parcel_bill_frame.grid(row=0, column=0, padx=20, pady=10, sticky="nsew")  # Added sticky to expand frame
title_label.grid(row=0, column=0, sticky="nsew")  # Position the label within the frame

# Spacing and centering elements within the frame
for widget in parcel_bill_frame.winfo_children():
    if isinstance(widget, (ttk.Label, ttk.Entry)):  # Check if widget is Label or Entry
        widget.grid_configure(padx=20, pady=5, sticky="nsew")

customer_name_label = ttk.Label(parcel_bill_frame, text="Customer Name:", style="Pink.TLabel")
customer_name_label.grid(row=0, column=0, padx=5, pady=5)
customer_name_var = tkinter.StringVar()
customer_name_entry = ttk.Entry(parcel_bill_frame)
customer_name_entry.grid(row=0, column=1, padx=5, pady=5)

parcel_type_label = ttk.Label(parcel_bill_frame, text="Parcel Type:", style="Pink.TLabel")
parcel_type_label.grid(row=1, column=0, padx=5, pady=5)
parcel_type_var = tkinter.StringVar(value="choose your parcel type")
parcel_types = ["envelope", "small box", "medium box", "large box"]
parcel_type_combobox = ttk.Combobox(parcel_bill_frame, textvariable=parcel_type_var, values=parcel_types)
parcel_type_combobox.grid(row=1, column=1, padx=5, pady=5)
```



```python

weight_label = ttk.Label(parcel_bill_frame, text="Weight (kg):", style="Pink.TLabel")
weight_label.grid(row=2, column=0, padx=5, pady=5)
weight_entry = ttk.Entry(parcel_bill_frame)
weight_entry.grid(row=2, column=1, padx=5, pady=5)

service_type_label = ttk.Label(parcel_bill_frame, text="Service Type:", style="Pink.TLabel")
service_type_label.grid(row=3, column=0, padx=5, pady=5)
service_type_var = tkinter.StringVar(value="standard")
service_types = ["express", "standard"]
service_type_combobox = ttk.Combobox(parcel_bill_frame, textvariable=service_type_var, values=service_types)
service_type_combobox.grid(row=3, column=1, padx=5, pady=5)

distance_label = ttk.Label(parcel_bill_frame, text="Distance (km):", style="Pink.TLabel")
distance_label.grid(row=4, column=0, padx=5, pady=5)
distance_entry = ttk.Entry(parcel_bill_frame)
distance_entry.grid(row=4, column=1, padx=5, pady=5)

total_bill_var = tkinter.StringVar()
total_bill_label = ttk.Label(parcel_bill_frame, textvariable=total_bill_var)
total_bill_label.grid(row=5, column=0, columnspan=2, padx=5, pady=5)

# Create update button
style.configure("Red.TButton", foreground="white", background="red")
update_button = ttk.Button(root, text="Update", command=parcel_bill, style="Red.TButton")
update_button.grid(row=1, column=0, pady=10)

# Centering elements within the root window
root.columnconfigure(0, weight=1)  # Allow the column to expand
root.rowconfigure(0, weight=1)  # Allow the row to expand

root.mainloop()

```

Figure 2 snapshot of code

4.0 SNAPSHOT OF GUI



Customer Name: SITI AISYAH NATASHA

Parcel Type: small box

Weight (kg): 2

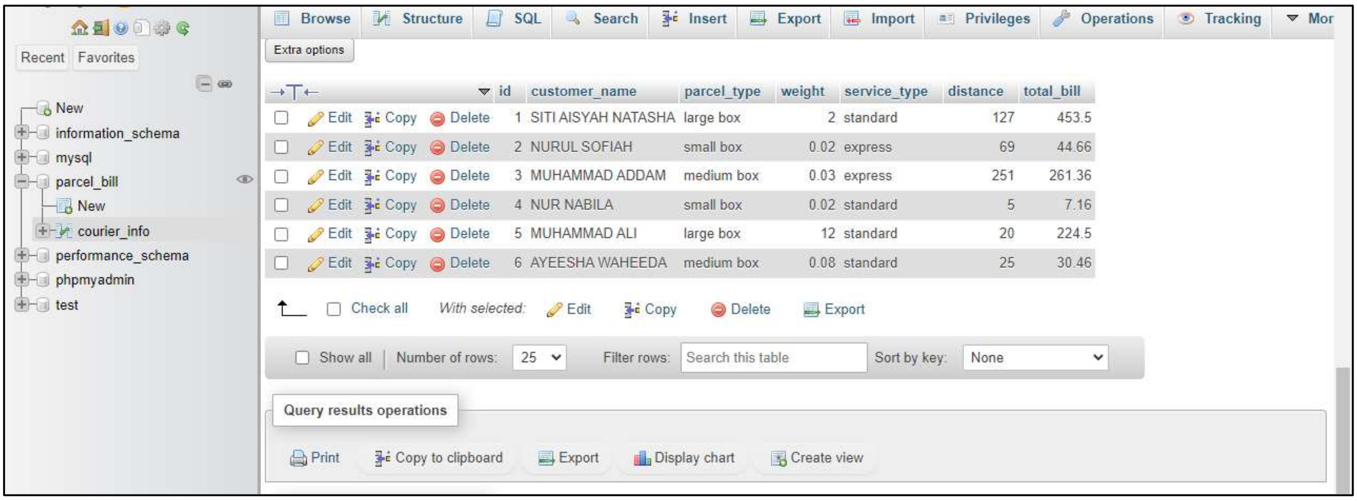
Service Type: standard

Distance (km): 56

Total Bill: RM 48.50

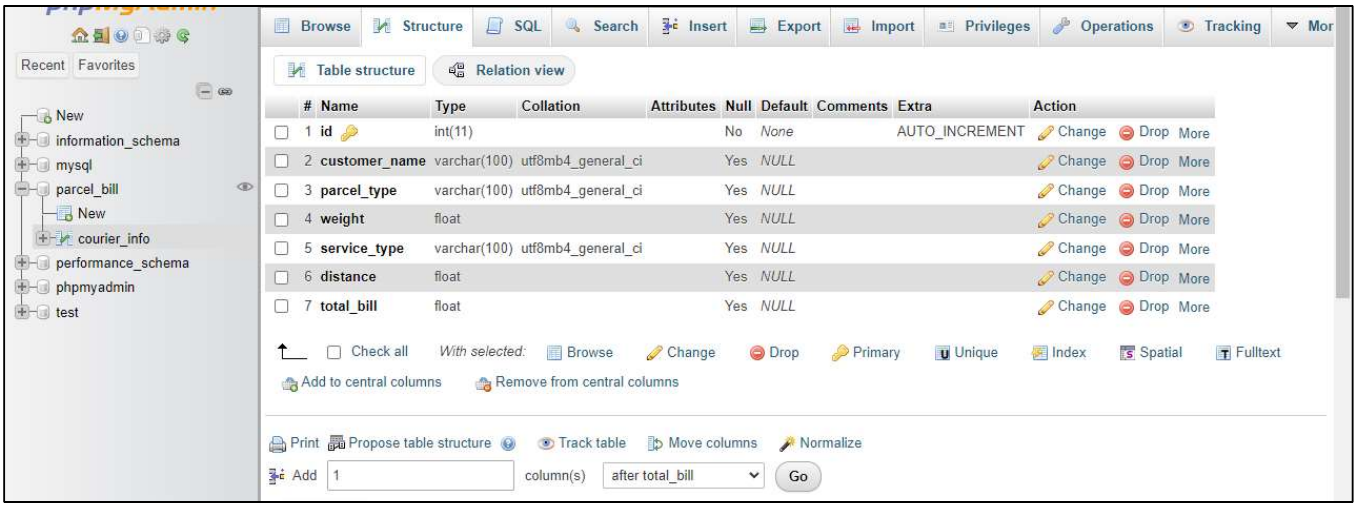
Figure 3 snapshot of GUI

5.0 SNAPSHOT OF DATABASE



				id	customer_name	parcel_type	weight	service_type	distance	total_bill
□	Edit	Copy	Delete	1	SITI AISYAH NATASHA	large box	2	standard	127	453.5
□	Edit	Copy	Delete	2	NURUL SOFIAH	small box	0.02	express	69	44.66
□	Edit	Copy	Delete	3	MUHAMMAD ADDAM	medium box	0.03	express	251	261.36
□	Edit	Copy	Delete	4	NUR NABILA	small box	0.02	standard	5	7.16
□	Edit	Copy	Delete	5	MUHAMMAD ALI	large box	12	standard	20	224.5
□	Edit	Copy	Delete	6	AYEESHA WAHEEDA	medium box	0.08	standard	25	30.46

Figure 4.0 snapshot of database



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	id	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
□	2	customer_name	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
□	3	parcel_type	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
□	4	weight	float			Yes	NULL			Change	Drop	More
□	5	service_type	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
□	6	distance	float			Yes	NULL			Change	Drop	More
□	7	total_bill	float			Yes	NULL			Change	Drop	More

Figure 4.1 snapshot of database