



اُونِيُوَرْسِيْتِي تِيْكْنُوْلُوْجِي مَارَا  
UNIVERSITI  
TEKNOLOGI  
MARA

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

DIPLOMA IN LIBRARY INFORMATIC  
(CDIM144)

PROGRAMMING FOR LIBRARIES (IML208)

GROUP PROJECT: TUITION MANAGEMENT SYSTEM

PREPARED BY:

NAME	MATRICES CARD NUMBER
ATHIRAH MAISARA BINTI MOHD NIZAM	2022449416
DAYANA BATRISYIA BINTI MOHD RABIEE	2022896288
HANIS FARZANA BINTI ZAINAL	2022481942
NURFADHILATULAKMA BINTI SERIPUL AZALI	2022463106

GROUP KCDIM1443E

PREPARED FOR: SIR AIRUL SHAZWAN BIN NORSHAHMI

SUBMISSION DATE: 16<sup>TH</sup> JANUARY 2024

**GROUP PROJECT: TUITION MANAGEMENT SYSTEM**

**BY**

<b>NAME</b>	<b>MATRICS CARD NUMBER</b>
<b>ATHIRAH MAISARA BINTI MOHD NIZAM</b>	<b>2022449416</b>
<b>DAYANA BATRISYIA BINTI MOHD RABIEE</b>	<b>2022896288</b>
<b>HANIS FARZANA BINTI ZAINAL</b>	<b>2022481942</b>
<b>NURFADHILATULAKMA BINTI SERIPUL AZALI</b>	<b>2022463106</b>

**COLLEGE OF COMPUTING, INFORMATICS AND MEDIA  
UNIVERSITI TEKNOLOGI MARA  
MERBOK, KEDAH**

**16<sup>TH</sup> JANUARY 2024**

## **ACKNOWLEDGEMENT**

ASSALAMUALAIKUM W.B.T,

First and foremost, praises and thanks to Allah SWT, for His showers of blessings throughout our group project to complete until the end. This project is really difficult and complex to do, however everything gets lot easier because of Allah's blessings. He provides us with strength and guidance, Alhamdulillah.

We would like to express our special thanks of gratitude to Sir Airul Shazwan Biin Norshahmi who gave us the golden opportunity to do this wonderful group project on the topic tuition management system, which also helped us in doing a lot of research and we came to know about so many stuffs we thankful to him.

Lastly, we would also like to thanks to our parents and our beloved friends for the support and the guidance. We are extremely grateful for them who helped us a lot in gathering different information and guide us from time to time in making this group project even though they are busy with their own stuff. Thank you, again, for everyone that are willing to cooperate to complete this project.

## TABLE OF CONTENT

CONTENT	NO PAGE
Acknowledgement	i
1.0 Introduction	1
2.0 Flowchart	2-4
3.0 Snapshot of coding	5-12
4.0 Snapshot of GUI	13-14
5.0 Snapshot of database	15-16
6.0 Conclusion	17

## 1.0 Introduction

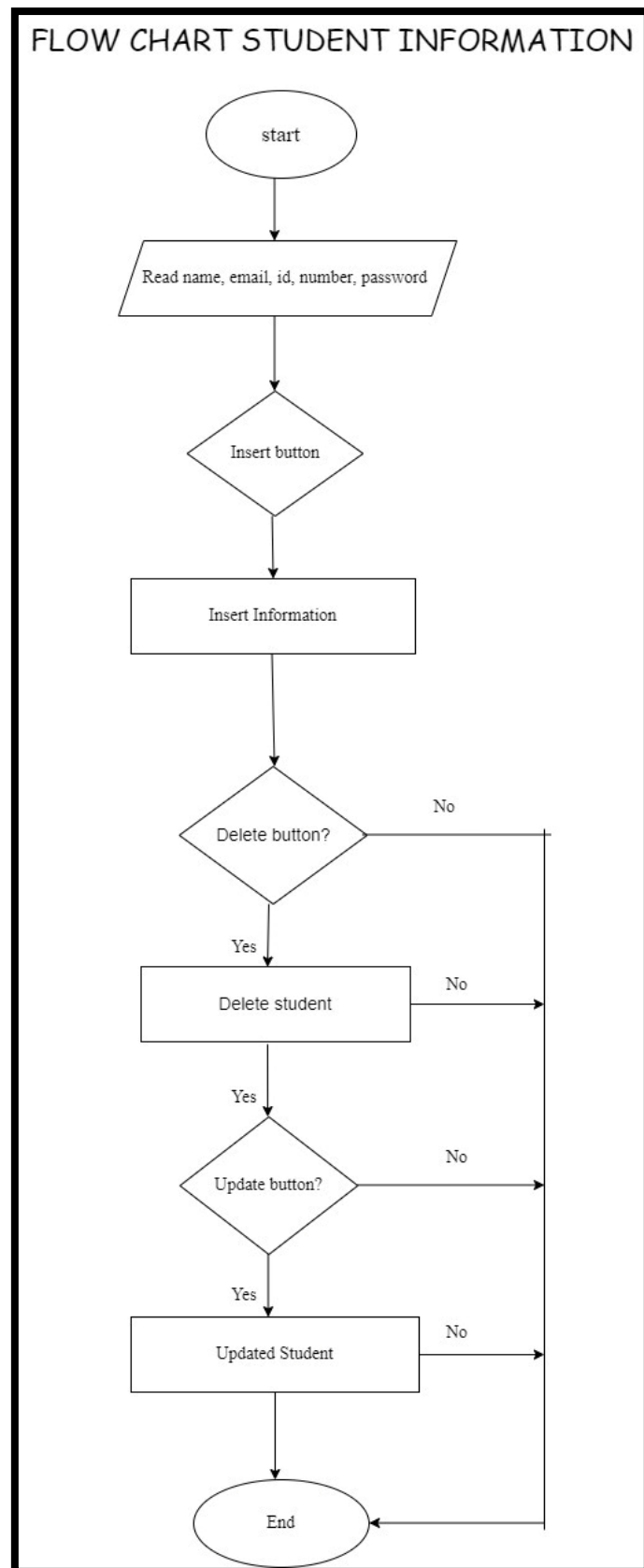
For this project, we are developing a tuition management system, that we called Brainnies Tuition Centre, which is a software solution that are designed to speed up and improve the administrative procedures that involved in managing tuition's subject packages, and fee-related activities throughout educational institutions. This system operates as an integrated framework for functions such as student enrolment, fee payment, and tuition branching. Tuition Brainnies eliminates repetitive operations, reduces administrative stress, and reduces the possibility of errors by combining capabilities, including payment process. We have 3 interfaces in total in our database, which are tuition branch, student information, and subject packages.

The first interface is student information. Students are essential to insert their information so that the tutors may see the student information without having any misunderstanding. They need to provide their name, email, phone number, student ID, and password. As a result, no one can access into the website without the student's consent. If there is an error, students have the option to update or delete the data. The second interface are tuition branch. Students have been given the option of taking lessons by online or physical class. Student that chooses online class, they are required to download Webex or Zoom applications in order to participate in the class. It's the only platforms for tutor to instruct the students. Meanwhile, the tuition centre is located in 3 different states, which are Johor, Kedah and Selangor. Each centre has a total of two branches.

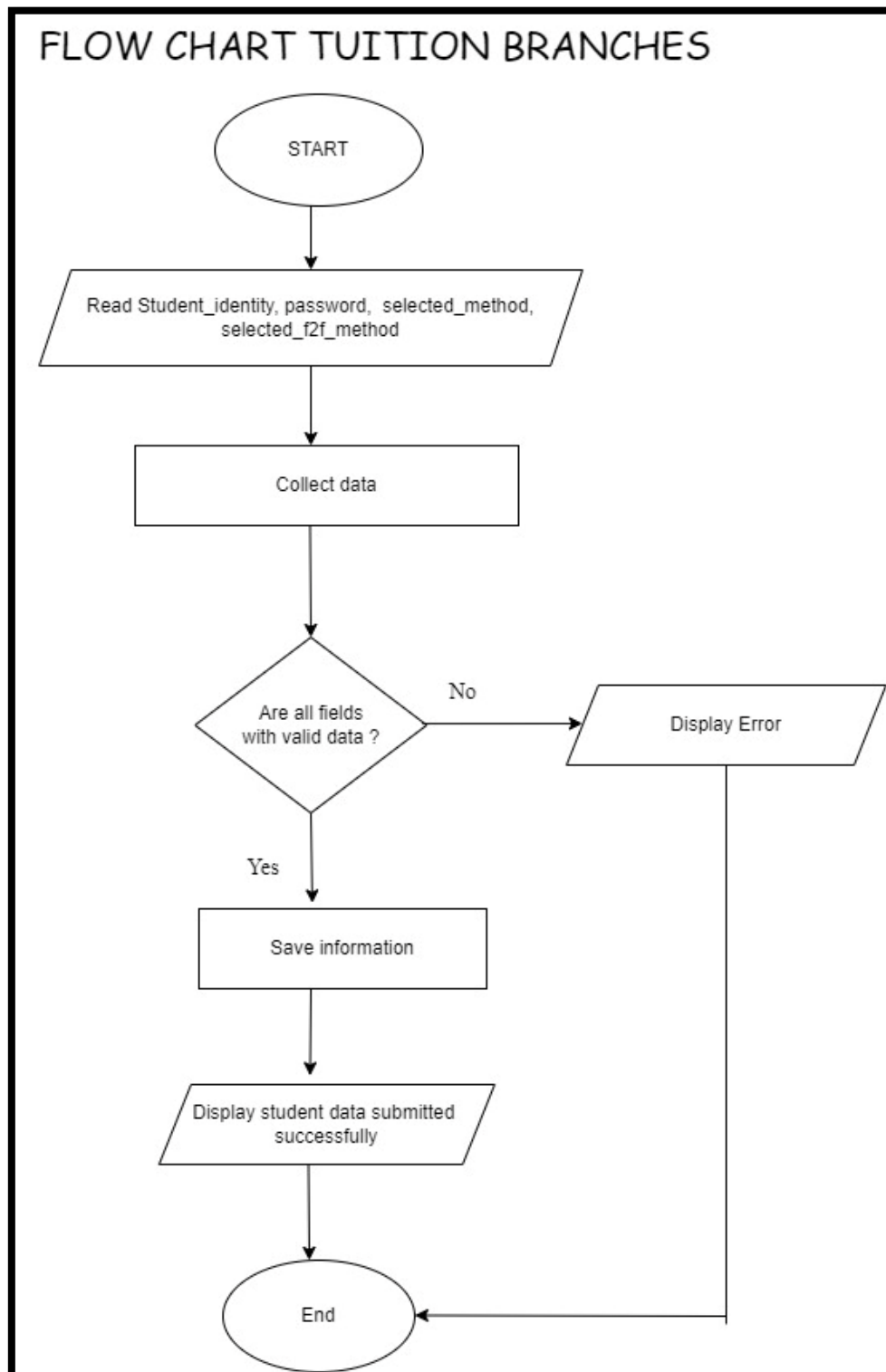
The last interface are the subject packages. There are 5 subjects in total in Brainnies Tuition Centre, which are Bahasa Melayu, English, Mathematics, Science, and History. There have been divided into 5 packages. Each students have to pay RM450 for tuition fees once they registered in Brainnies Tuition Centre. Moreover, students can get 15% off by choosing package 5 since they take all of the subject.

## 2.0 Flowchart

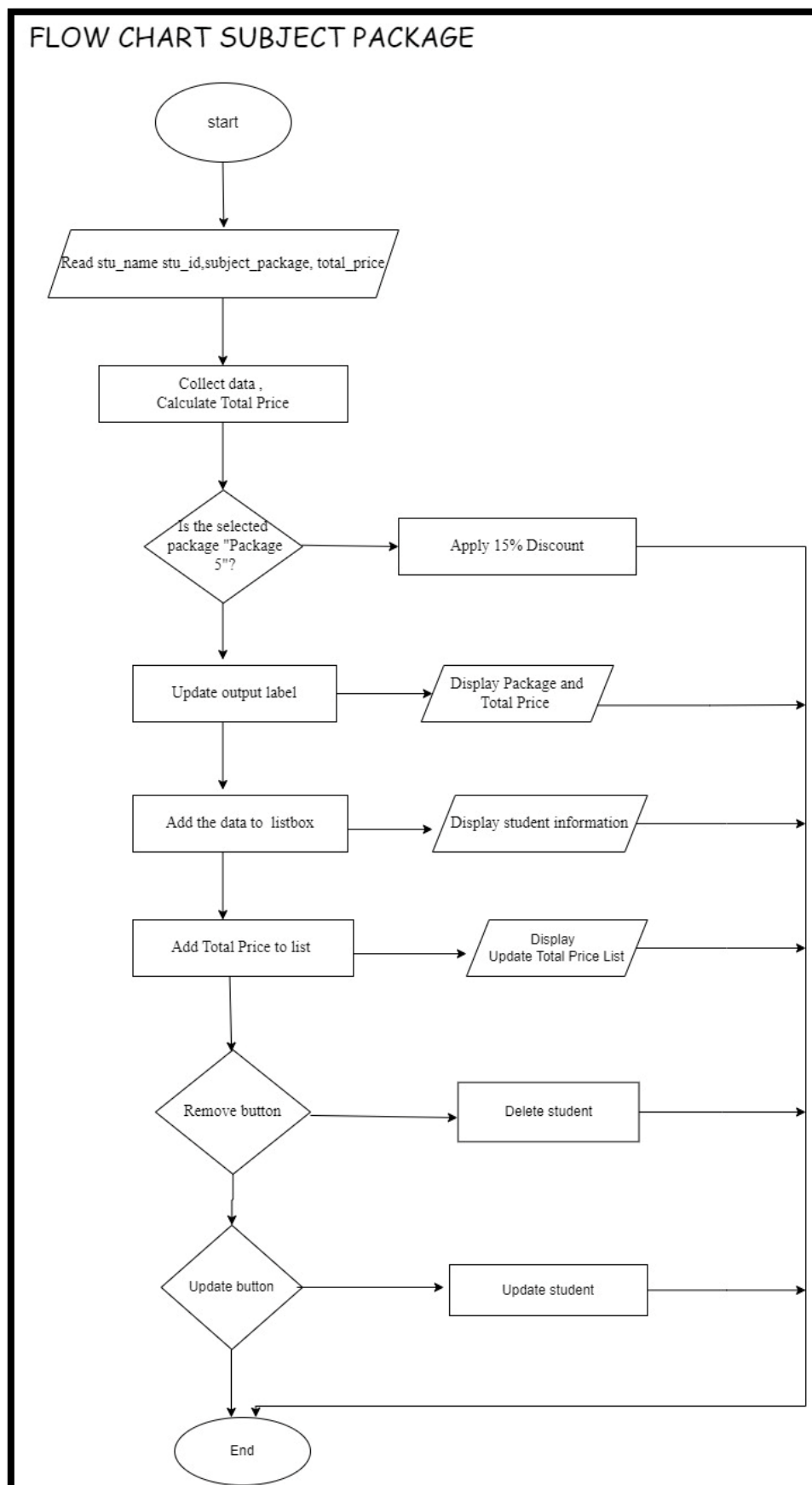
### 2.1 Student information



## 2.2 Tuition branches



## 2.3 Subject packages





### 3.0 Snapshot of coding

#### 3.1 Student information

```
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="tuition_management"
11 )
12
13 # Create a cursor object to execute SQL queries
14 mycursor = mydb.cursor()
15
16 def student_information():
17     #function of button
18     name = name_entry.get()
19     email = email_entry.get()
20     id = int(student_id_entry.get())
21     number = no_phone_entry.get()
22     password = password_entry.get()
23
24     # print the information
25     print("student information:")
26     print("name:", name)
27     print("email:", email)
28     print("Student identification:", id)
29     print("phone Number:", number)
30     print("password:", password)
31
32     # insert database
33     sql = "INSERT INTO `student_information` (student_name, student_email, student_id, phone_number, student_password) VALUES (%s, %s, %s,%s,%s)"
34     val = (name, email, id, number, password)
35     mycursor.execute(sql, val)
36     mydb.commit()
37
```

```
37
38 def delete_student():
39     try:
40         student_id = int(student_id_entry.get())
41
42         # Assuming student_id is the primary key for deletion
43         sql = "DELETE FROM student_information WHERE student_id = %s"
44         val = (student_id, )
45
46         mycursor.execute(sql, val)
47         mydb.commit()
48
49         messagebox.showinfo("Success", "Data deleted successfully.")
50     except ValueError:
51         messagebox.showerror("Error", "Invalid input.")
52     except Exception as e:
53         messagebox.showerror("Error", str(e))
54
55
56
57 def update_student():
58     try:
59         student_id = int(student_id_entry.get())
60         new_phone_number = no_phone_entry.get()
61         new_email = email_entry.get()
62
63         sql = "UPDATE student_information SET phone_number = %s, student_email = %s WHERE student_id = %s"
64         val = (new_phone_number, new_email, student_id)
65         mycursor.execute(sql, val)
66         mydb.commit()
67
68         messagebox.showinfo("Success", "Student information updated!")
69     except Exception as e:
70         messagebox.showerror("Error", str(e))
71
```

```

72 #GUI
73 root = tk.Tk()
74 root.title("Tuition Brannies")
75 root.geometry('1000x900')
76 root.configure(bg="salmon1")
77
78 frame = tk.Frame(root,bg="salmon1")
79 frame.pack()
80 frame.config()
81
82 #student information
83 stu_info_frame = tk.LabelFrame(frame, text="Student Information")
84 stu_info_frame.grid(row=3, column=0, ipadx=20, ipady=5)
85 stu_info_frame.configure(bg="salmon1", fg="saddle brown", font=("Helvetica",20, "bold"))
86
87 name_label = tk.Label(stu_info_frame, text="name")
88 name_label.grid(row=0, column=0)
89 name_label.configure(fg="brown", font=("Arial",12, "bold"))
90
91 name_entry = tk.Entry(stu_info_frame, width=50)
92 name_entry.grid(row=1, column=0, padx=60, pady=10)
93
94 email_label = tk.Label(stu_info_frame, text="email")
95 email_label.grid(row=2, column=0)
96 email_label.configure(fg="brown", font=("Arial",12, "bold"))
97
98 email_entry = tk.Entry(stu_info_frame, width=40)
99 email_entry.grid(row=3, column=0, padx=60, pady=10)
100
101 student_id_label= tk.Label(stu_info_frame, text="student identification")
102 student_id_label.grid(row=0, column=1)
103 student_id_label.configure(fg="brown", font=("Arial",12, "bold"))
104
105 student_id_entry = tk.Entry(stu_info_frame, width=50)
106 student_id_entry.grid(row=1, column=1, padx=60, pady=10)
107

```

```

108 no_phone_label= tk.Label(stu_info_frame, text="phone number")
109 no_phone_label.grid(row=2, column=1)
110 no_phone_label.configure(fg="brown", font=("Arial",12, "bold"))
111
112 no_phone_entry = tk.Entry(stu_info_frame, width=40)
113 no_phone_entry.grid(row=3, column=1, padx=60, pady=10)
114
115 password_label= tk.Label(stu_info_frame, text="password")
116 password_label.grid(row=4, column=0)
117 password_label.configure(fg="brown", font=("Arial",12, "bold"))
118
119 password_entry = tk.Entry(stu_info_frame, width=40)
120 password_entry.grid(row=5, column=0, padx=60, pady=10)
121
122 insert_button = tk.Button(stu_info_frame, text="Register", command=student_information)
123 insert_button.grid(row=5, column=1)
124 insert_button.configure(bg="chocolate3", fg="peach puff", font=("Arial", 12 , "bold"))
125
126
127 # Buttons for delete and update functions
128 delete_button = tk.Button(stu_info_frame, text="delete", command=delete_student)
129 delete_button.grid(row=5, column=0, columnspan=5, pady=10)
130 delete_button.configure(bg="chocolate3", fg="peach puff", font=("Arial", 12 , "bold"))
131
132 update_button = tk.Button(stu_info_frame, text="update", command=update_student)
133 update_button.grid(row=5, column=2, columnspan=4, pady=10)
134 update_button.configure(bg="chocolate3", fg="peach puff", font=("Arial", 12 , "bold"))
135
136 # Loop to keep the GUI running
137 while True:
138     try:
139         root.update()
140     except tk.TclError:
141         break # Exit the loop if the user closes the window
142 root.mainloop()
143

```

## 3.2 Tuition branch

```
1  import tkinter as tk
2  from tkinter import ttk
3  from tkinter import messagebox
4  import mysql.connector
5
6  # Connect to your MySQL database
7  mydb = mysql.connector.connect(
8      host="localhost",
9      user="root",
10     password="",
11     database="tuition_management"
12 )
13
14 # Create a cursor object to execute SQL queries
15 mycursor = mydb.cursor()
16
17 def collect_data():
18     Student_identity = int(stu_entry.get())
19     password = int(password_entry.get())
20     selected_method = method_online.get()
21     selected_f2f_method = f2f_method.get()
22
23     print("Student:", Student_identity)
24     print("password student:", password)
25     print("method", selected_method)
26     print("f2f method", selected_f2f_method)
27
28
29 # Inserting data into the database
30 try:
31     sql = "INSERT INTO tuition_branches (student_id, password, method, tui_area) VALUES (%s, %s, %s, %s)"
32     val = (Student_identity, password, selected_method, selected_f2f_method)
33     mycursor.execute(sql, val)
34     mydb.commit()
35
```

```
36
37 # To show the data entry is successful or not.
38 if not Student_identity or not password or not selected_method or not selected_f2f_method:
39     messagebox.showerror("Error", "Please fill in all fields and enter a valid data.")
40 else:
41     messagebox.showinfo('Success', "Student data submitted successfully")
42
43 except ValueError:
44     messagebox.showinfo("Error", "Invalid input. Please enter valid numeric value for id number and password")
45
46 root= tk.Tk()
47 root.title("brainnies tuition")
48 root.geometry('600x600')
49 root.configure(bg="salmon1")
50
51 frame = tk.Frame(root,bg="salmon1")
52 frame.pack()
53 frame.config()
54
55 #page title
56 label=tk.Label(root, text ='WELCOME TO BRAINNIES TUITION', font=("Times New Roman", 20, "bold", ))
57 label.pack(padx=10, pady=10)
58
59 frame= tk.Frame(root, height=40, width=50)
60 frame.pack(padx=20, pady=20)
61
62 #student id
63 stu_label= tk.Label(frame, text= "student identity\n", font=("Times New Roman", 10,))
64 stu_label.grid(row=1, column=0, padx=4, pady=4)
65 stu_entry = tk.Entry(frame)
66 stu_entry.grid(row=2, column=0)
67
68 password=tk.Label(frame, text= "password", font=("Times New Roman",10))
69 password.grid(row=3, column=0, padx=5, pady=5, ipadx=6, ipady=6)
70 password_entry=tk.Entry(frame)
71 password_entry.grid(row=4, column=0, padx=5, pady=5, ipadx=3, ipady=3)
72
```

```

71
72 #method
73 insert_info_frame = tk.LabelFrame(frame, text="\nCHOOSE YOUR METHOD\n")
74 insert_info_frame.grid( row=6, column=0, ipadx=50, ipady=50)
75
76 # Default value before your selection
77 method_online = ttk.Combobox(insert_info_frame, values= ["face-to-face","online"])
78 method_online.grid(row=9, column=0, padx=5, pady=5, ipadx=7, ipady=7)
79 method_online.set("Select Your Method")
80
81 f2f_method=ttk.Combobox(insert_info_frame, values=["Kedah : Sungai Petani", "Kedah : Kulim",
82                                                  "Johor : Muar ", "Johor : Pontian ",
83                                                  "Selangor : Shah Alam", "Selangor : Sepang", "ZOOM", "WEBEX"])
84 f2f_method.set("Pick the Place")
85 f2f_method.grid(row=13, column=0, padx=5, pady=5, ipadx=7, ipady=7)
86
87
88 # Save Button
89 save_button = tk.Button(frame, text="Submit", command=collect_data,)
90
91 command=(collect_data)
92 save_button.grid(row=24, column=0, padx=5, pady=5, ipadx=5, ipady=5)
93
94
95
96 root.mainloop()

```

### 3.3 Subject packages

```
C: > Users > thira > Downloads > Telegram Desktop > mine_3.py > collect_data
1  import tkinter as tk
2  from tkinter import messagebox
3  import mysql.connector
4
5  mydb = mysql.connector.connect(
6      host="localhost",
7      user="root",
8      password="",
9      database="tuition_management"
10 )
11
12 mycursor = mydb.cursor()
13
14 # Global list to store total prices
15 total_prices_list = []
16
17 def collect_data():
18     stu_name = stu_name_entry.get()
19     stu_id = stu_id_entry.get()
20     subject_package = package_var.get()
21
22     # Prices for different packages
23     prices = {
24         "Package 1": 58,
25         "Package 2": 58,
26         "Package 3": 68,
27         "Package 4": 78,
28         "Package 5": 98,
29     }
30
31     total_price = prices[subject_package] + 450
32
33     # Check if the selected package is Package 5
34     if subject_package == "Package 5":
35         discount_amount = 0.15 * total_price
36         disc_price = total_price - discount_amount
37         output_label.config(text=f"Total Price: RM{disc_price:.2f}")
```

```

38     else:
39         disc_price = total_price
40         output_label.config(text=f"Total Price: RM{disc_price:.2f}")
41
42     # Insert data into the database
43     sql = "INSERT INTO subject_information (stu_name, stu_id, sub_package, total_price) VALUES (%s, %s, %s, %s)"
44     val = (stu_name, stu_id, subject_package, disc_price)
45     mycursor.execute(sql, val)
46     mydb.commit()
47
48     # Update the output label
49     output_label.config(text=f"Package: {subject_package}, Total Price: RM{disc_price:.2f}")
50
51 def remove_student():
52     try:
53         stu_id = int(stu_id_entry.get())
54
55         # Delete data from the database
56         sql = "DELETE FROM subject_information WHERE stu_id=%s"
57         val = (stu_id,)
58         mycursor.execute(sql, val)
59         mydb.commit()
60
61         messagebox.showinfo("Success", "Student data deleted successfully")
62     except ValueError:
63         messagebox.showerror("Error", "Invalid input for student ID.")
64     except Exception as e:
65         print(e)
66         messagebox.showerror("Error", "An error occurred while deleting the student data.")
67
68 def student_update():
69     try:
70         stu_id = int(stu_id_entry.get())
71         new_subject_package = package_var.get()
72
73         if not new_subject_package:
74             messagebox.showerror("Error", "Please select a subject package.")

```

```

75         return
76
77         sql = "UPDATE subject_information SET sub_package=%s WHERE stu_id=%s"
78         val = (new_subject_package, stu_id)
79
80         mycursor.execute(sql, val)
81         mydb.commit()
82
83         messagebox.showinfo("Success", "Student information updated!")
84     except ValueError:
85         messagebox.showerror("Error", "Invalid input for student ID.")
86     except Exception as e:
87         messagebox.showerror("Error", str(e))
88
89
90 # Create Window
91 root = tk.Tk()
92 root.title("Tuition Management System")
93 root.geometry("800x800")
94 root.configure(bg="salmon1")
95
96
97 # Title
98 label = tk.Label(root, text="Package Price", font=("", 10, "bold"), bg= "salmon1")
99 label.grid(ipadx=10, ipady=10)
100
101 # Prices text
102 prices_text = tk.Text(root, height=17, width=85, font=("Helvetica", 10))
103 prices_text.grid(row=2, column=0)
104
105 prices_text.insert(tk.END, "Package & Fees:\n\n")
106 prices_text.insert(tk.END, "Package 1: BM & BI \nPrice: RM58\n\n")
107 prices_text.insert(tk.END, "Package 2: MATHS & SCIENCE \nPrice: RM58\n\n")
108 prices_text.insert(tk.END, "Package 3: BM & HISTORY \nPrice: RM68\n\n")
109 prices_text.insert(tk.END, "Package 4: MATHS, SCIENCE, HISTORY \nPrice: RM78\n\n")
110 prices_text.insert(tk.END, "Package 5: BM, BI, MATHS, SCIENCE, HISTORY \nPrice: RM98\n\n")
111 prices_text.insert(tk.END, "Fees: RM450 per person\n\n")

```

```

112 prices_text.insert(tk.END, "Discount: 10% off for Package 5 only \n")
113 prices_text.configure(state='disabled')
114
115 # Frame
116 frame = tk.Frame(root)
117 frame.grid()
118
119 # Student Information Frame
120 stu_info_frame = tk.LabelFrame(frame, text="")
121 stu_info_frame.grid(row=0, column=0, padx=20, pady=10)
122
123 stu_name_label = tk.Label(stu_info_frame, text="Student Name:", width=50)
124 stu_name_label.grid(row=0, column=0)
125
126 stu_name_entry = tk.Entry(stu_info_frame, width=50)
127 stu_name_entry.grid(row=1, column=0)
128
129 stu_id_label = tk.Label(stu_info_frame, text="Student Identification Number:", width=50)
130 stu_id_label.grid(row=0, column=1)
131
132 stu_id_entry = tk.Entry(stu_info_frame, width=50)
133 stu_id_entry.grid(row=1, column=1)
134
135 package_label = tk.Label(stu_info_frame, text="Subject Package:", width=50)
136 package_label.grid(row=3, column=0)
137
138 package_var = tk.StringVar(stu_info_frame)
139 package_var.set("Select Subject Package")
140 package_dropdown = tk.OptionMenu(stu_info_frame, package_var, "Package 1", "Package 2", "Package 3", "Package 4", "Package 5")
141 package_dropdown.grid(row=4, column=0)
142
143 # Button
144 add_button = tk.Button(root, text="Add Student", command=collect_data)
145 add_button.grid(pady=10)
146
147 # Delete Button
148 button_frame = tk.LabelFrame(root, text="")

```

```

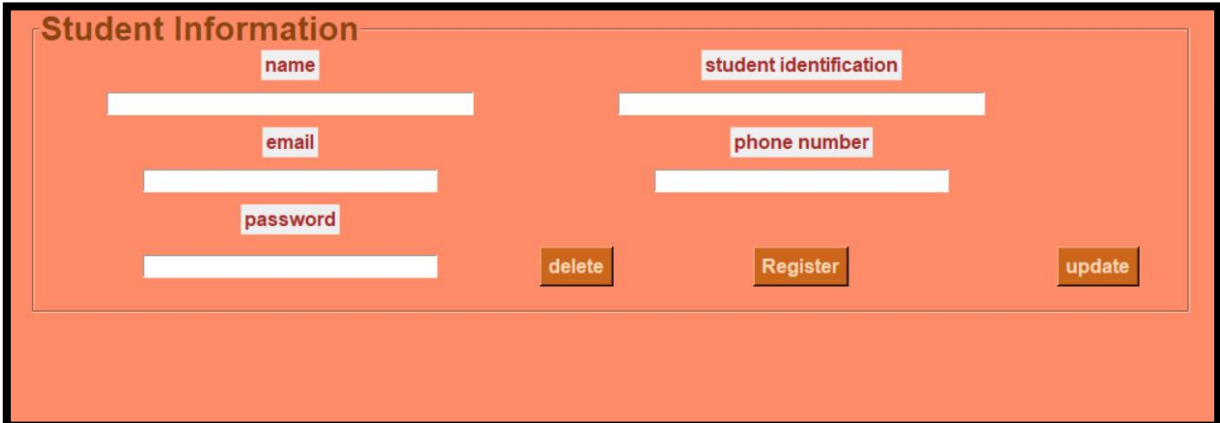
149 button_frame.grid(row=10, column=0, padx=10, pady=10)
150 delete_button = tk.Button(button_frame, text="Remove", command=remove_student)
151 delete_button.grid(row=10, column=1, pady=15)
152
153 # Update Button
154 update_button = tk.Button(button_frame, text="Update", command= student_update)
155 update_button.grid(row=10, column=0, pady=10)
156
157 # Total Price Label
158 label = tk.Label(root, text='Total Price', font=("Times New Romans", 12), bg="salmon1")
159 label.grid(ipadx=10, ipady=10)
160 output_label = tk.Label(root, text="")
161 output_label.grid()
162
163
164 root.mainloop()

```



## 4.0 Snapshot of GUI

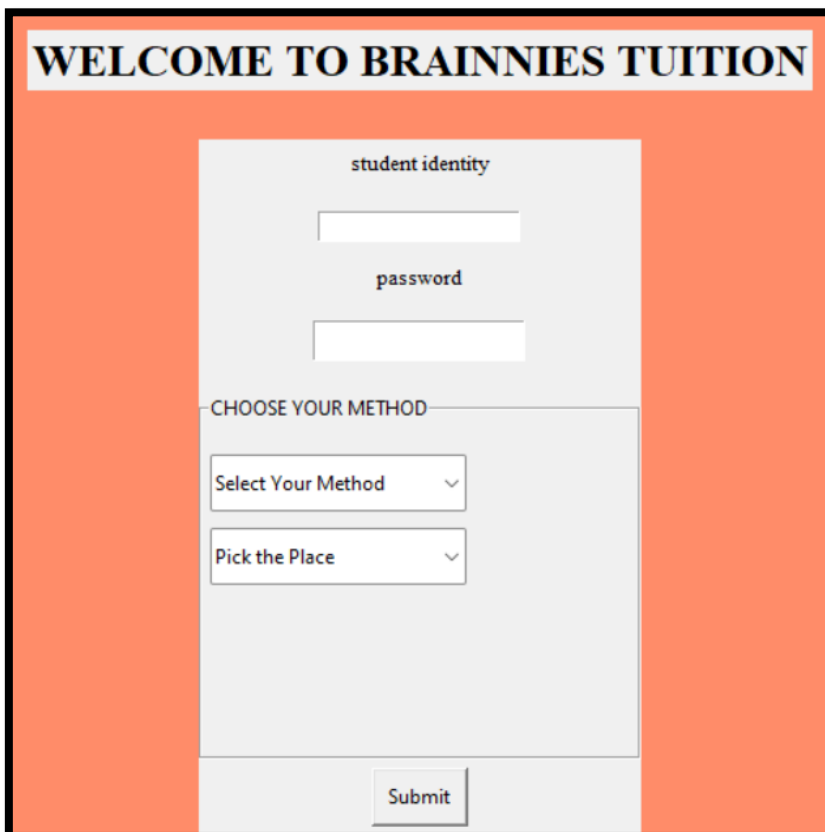
### 4.1 Student Information



A screenshot of a web form titled "Student Information" with an orange background. The form contains several input fields and three buttons. The fields are arranged in two columns: "name" and "student identification" on the top row, "email" and "phone number" on the second row, and "password" on the third row. At the bottom right, there are three buttons: "delete", "Register", and "update".

Student Information	
name	student identification
email	phone number
password	
	delete Register update

### 4.2 Tuition branches



A screenshot of a web form titled "WELCOME TO BRAINNIES TUITION" with an orange background. The form is centered in a light gray box. It contains input fields for "student identity" and "password". Below these is a section titled "CHOOSE YOUR METHOD" with two dropdown menus: "Select Your Method" and "Pick the Place". A "Submit" button is at the bottom.

**WELCOME TO BRAINNIES TUITION**

student identity

password

CHOOSE YOUR METHOD

Select Your Method

Pick the Place

Submit

### 4.3 Subject packages

Package Price								
<p>Package &amp; Fees:</p> <p>Package 1: BM &amp; BI Price: RM58</p> <p>Package 2: MATHS &amp; SCIENCE Price: RM58</p> <p>Package 3: BM &amp; HISTORY Price: RM68</p> <p>Package 4: MATHS, SCIENCE, HISTORY Price: RM78</p> <p>Package 5: BM, BI, MATHS, SCIENCE, HISTORY Price: RM98</p>								
<table><tr><td>Student Name:</td><td>Student Identification Number:</td></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td>Subject Package:</td><td></td></tr><tr><td><input type="button" value="Select Subject Package"/></td><td></td></tr></table>	Student Name:	Student Identification Number:	<input type="text"/>	<input type="text"/>	Subject Package:		<input type="button" value="Select Subject Package"/>	
Student Name:	Student Identification Number:							
<input type="text"/>	<input type="text"/>							
Subject Package:								
<input type="button" value="Select Subject Package"/>								
<div><div>Add Student</div><div><input type="text"/></div><div><div>Update</div><div>Remove</div></div><div>Total Price</div></div>								

## 5.0 Snapshot of database

### 5.1 Student information

The screenshot shows the phpMyAdmin interface for the 'tuition\_management' database. The left sidebar lists various databases and tables, with 'tuition\_management' expanded to show 'student\_information', 'subject\_information', and 'tuition\_branches'. The main panel displays the 'student\_information' table structure and data. A warning message states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green bar indicates 'Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)'. The SQL query shown is 'SELECT \* FROM `student\_information`'. Below the query, there are options for 'Show all', 'Number of rows' (set to 25), and 'Filter rows' (Search this table). An 'Extra options' button is also present. The table data is as follows:

student_name	student_id	phone_number	student_email	student_password
raziq	2022431	122096690	raziq20@gmail.com	167827
sara	2022786	106632756	sara.m@gmail.com	56724
zaza	2022917	162522685	zaza167@gmail.com	24267
kamaruddin	2022691	195724189	din675@gmail.com	13279

### 5.2 Tuition branches

The screenshot shows the phpMyAdmin interface for the 'tuition\_management' database, specifically the 'tuition\_branches' table. The left sidebar shows the database structure with 'tuition\_branches' selected. The main panel displays the table structure and data. A warning message states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green bar indicates 'Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)'. The SQL query shown is 'SELECT \* FROM `tuition\_branches`'. Below the query, there are options for 'Show all', 'Number of rows' (set to 25), and 'Filter rows' (Search this table). An 'Extra options' button is also present. The table data is as follows:

student_id	password	method	tui_area
16782	1427	Select You	Kedah : Kulim
17195	234567	online	ZOOM
345678	2349	face-to-fa	Selangor : Shah Alam
5678	1027	face-to-fa	Johor : Muar
222890	8167	face-to-fa	Selangor : Sepang
23456789	4567098	online	WEBEX

## 5.3 Subject packages

Server: 127.0.0.1 » Database: tuition\_management » Table: subject\_information

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

Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

```
SELECT * FROM `subject_information`
```

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

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

Extra options

stu_name	stu_id	sub_package	total_price
yaya	202234567	Package 1	508.0
azim	202298765	Package 1	508.0
abdul razak	202245679	Package 5	528.0
fatihah	20220171	Package 2	508.0
athirah	2021884732	Package 5	465.8
najuwa	2021904212	Package 3	518

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

Query results operations

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

[Bookmark this SQL query](#)

Console

## 6.0 Conclusion

In conclusion, we create an interactive GUI using Tkinter for students to enter their information and also for owner to calculate the total price and average price. In subject information and student information, we added buttons like add, remove and update as a function for users and owners to add data, remove it or update information. The GUI interface have connected to database where the information is stored, this is a very simple and straightforward project making it easy for owners to record student information as well as their subject information. There is also calculation to calculate the total price of the chosen subject pack.

However, there are several flaws in this system that should be addressed. The existing interface could cause difficulties for students, perhaps leading to dissatisfaction and delays. Addressing these difficulties by incorporating students' information, organizing navigation, and improving the overall design can lead to a more efficient and satisfying registration experience. For instance, having an improved instructed and well-organized interface to ensure that students and owner are able to comprehend and utilize the interface more easily. In the evolving world of academic registration systems, it is critical to consider the students' perspective and implement modifications that not only fulfil present expectations but also anticipate future requirements.