



UNIVERSITI TEKNOLOGI MARA

KEDAH BRANCH

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATICS LIBRARY (IM144)

IML208 : PROGRAMMING FOR LIBRARIES

INDIVIDUAL PROJECT: DATA ENTRY FOR MYSQL DATABASE

CAT DAYCARE CENTRE

Prepared by:

NUR MAISARAH BINTI ABDUL MANAH

(2022892948)

GROUP KCDIM1443F

Prepared for:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

Submission date:

04.01.2024

INDIVIDUAL PROJECT: DATA ENTRY FOR MYSQL DATABASE

CAT DAYCARE CENTRE

PREPARED BY:

NUR MAISARAH BINTI ABDUL MANAH

(2022892948)

GROUP KCDIM1443F

IM144 – DIPLOMA IN INFORMATICS LIBRARY

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVESITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

ACKNOWLEDGMENT

I want to take this opportunity to express my gratitude to those who have supported and inspired me throughout the completion of this assignment.

First and foremost, I would like to thank Sir Airul Shazwan Bin Norshahimi

for his guidance, expertise, and continuing support, your help has been instrumental in shaping this work.

I am also indebted to my classmates and friends who offered encouragement and engaged in valuable discussions on the subject matter. Your input has greatly contributed to the depth and quality of this assignment.

Furthermore, I want to express my heartfelt thanks to my family for their unwavering encouragement and understanding during the countless hours spent on this project-/Finally, I want to extend my appreciation to Allah SAW for giving me the necessary motivation and strength to complete this assignment.

This assignment would not have been completed without the collective support and encouragement from the aforementioned individuals. Thank you for being a part of this journey.

TABLE OF CONTENTS

ACKNOWLEDGMENT.....	i
1.0 INTRODUCTION	1
2.0 FLOWCHART	2
3.0 PYTHON CODE.....	3
4.0 MYSQL DATABASE	6

1.0 INTRODUCTION

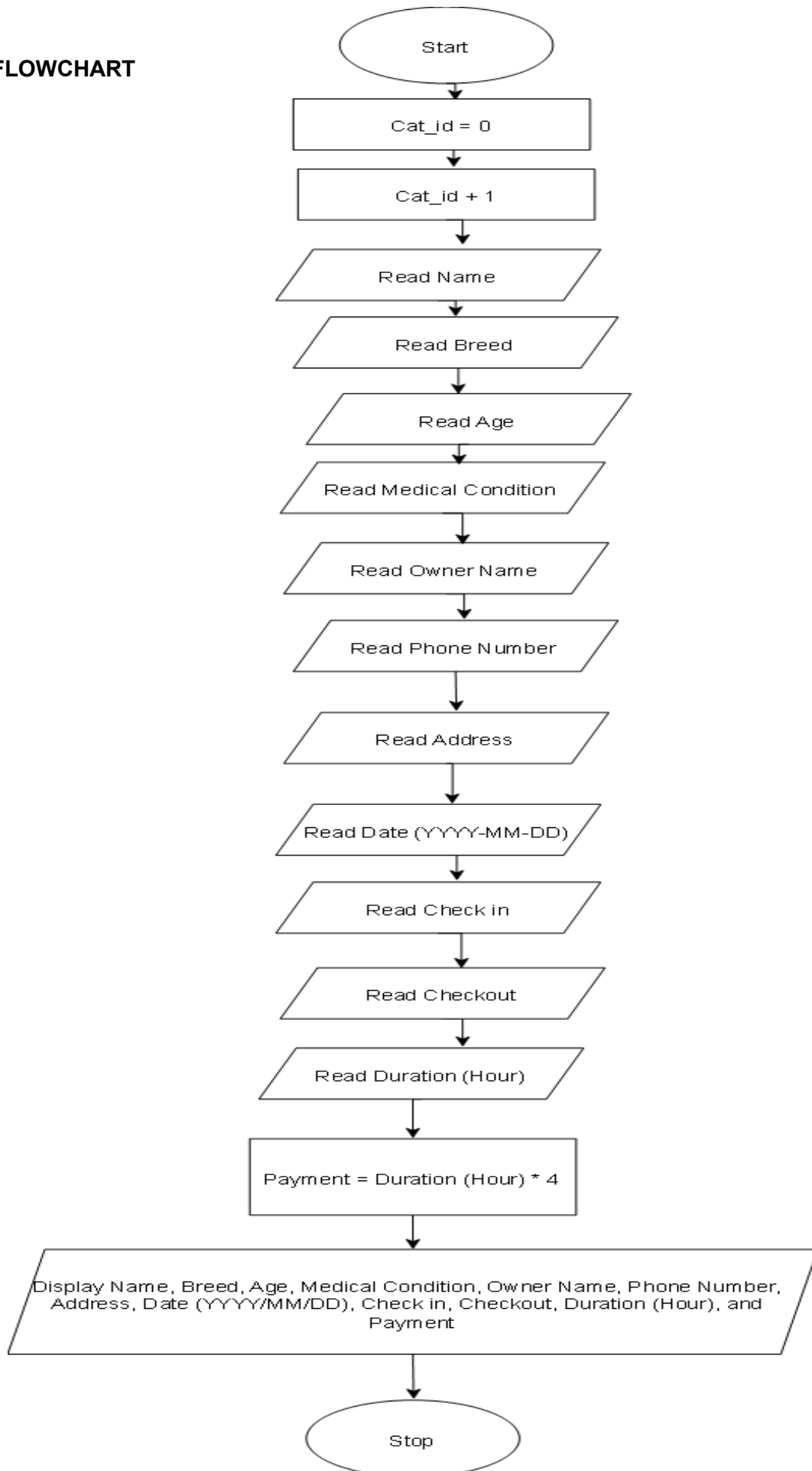
Welcome to the purrfect world of our Cat Daycare Database, where feline enthusiasts and their delightful companions find a seamless haven for their daycare needs. This comprehensive database serves as the digital heartbeat of our feline-centric haven, meticulously storing essential information about each cat, their devoted owners, and the crucial reservation details.

In the bustling realm of modern pet care, the demand for organized and efficient management has become increasingly evident. Our Cat Daycare Database addresses this need with precision and care, streamlining the process of keeping track of the diverse personalities and requirements of our furry guests. Through the integration of advanced technology, this database ensures a smooth and enjoyable experience for both the cats and their loving owners.

The necessity for such a database arises from the complexities of managing a dynamic cat daycare environment. From name and health records, our database ensures that every detail is securely recorded and easily accessible. By storing information about the owners, we facilitate seamless communication, creating a bond of trust between the daycare staff and the caregivers. Moreover, the reservation time details are diligently documented to ensure a structured and organized daily routine, providing cats with the consistency they thrive on.

In essence, our Cat Daycare Database isn't just a repository of data; it's the heartbeat of our commitment to exceptional feline care. As we navigate the intricate world of cat daycare, this database is a testament to our dedication to creating a comfortable, safe, and loving environment for our four-legged friends and their human companions. Welcome to a world where every meow matters and every detail is lovingly preserved in our feline haven's digital embrace.

2.0 FLOWCHART



3.0 PYTHON CODE

```
1 import tkinter
2 from tkinter import ttk
3 import mysql.connector
4
5 class CatGenerator:
6     def __init__(self):
7         self.cat_id = 0
8
9     def generate_running_number(self):
10         self.cat_id += 1
11         return self.cat_id
12
13     def insert_running_number(self, cat_name, cat_breed, cat_age, cat_medcon, date, check_in, check_out, duration_hour, owner_name, phone_num,
14 address, payment):
15         cat_id = self.generate_running_number()
16
17         sql = 'INSERT INTO cat(CAT_ID, CAT_NAME, CAT_BREED, CAT_AGE, CAT_MEDCON, DATE, CHECK_IN, CHECK_OUT, DURATION_HOUR, OWNER_NAME,
18 PHONE_NUM, ADDRESS, PAYMENT) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)'
19
20         val = (cat_id, cat_name, cat_breed, cat_age, cat_medcon, date, check_in, check_out, duration_hour, owner_name, phone_num, address,
21 payment)
22
23         try:
24             cursor.execute(sql, val)
25             mydb.commit()
26             print('Data saved successfully.')
27         except Exception as e:
28             print(f'Error saving data: {e}')
```

Figure 3.1 Python code for generating running number and connecting python code to mysql database

```
26
27 # Connect to your MySQL database
28 mydb = mysql.connector.connect(
29     host='localhost',
30     user='root',
31     password='',
32     database='cat_daycare_centre'
33 )
34
35 # Create a cursor object to execute SQL queries
36 cursor = mydb.cursor()
37
```

Figure 3.2 Connecting python code to mysql database

```

65 def perform_calculation():
66     try:
67         # Get the value from the Entry widget and convert it to a float
68         hour = float(duration_hour_entry.get())
69
70         # Perform a calculation
71         payment = hour * 4
72
73         # Update the result_label with the calculated result
74         payment_label.config(text=f'Payment: RM {payment}')
75
76     except ValueError:
77         #Handle the case where the input is not a valid float
78         payment_label.config(text='Invalid input. Please enter a number.')
79

```

Figure 3.3 Python code for performing calculation(payment)

```

89 #GUI code
90
91 #cat's name
92 cat_name_label = tkinter.Label(cat_info_frame, text = 'Name: ')
93 cat_name_label.grid(row = 0, column = 0, padx=10, pady=10)
94 cat_name_entry = tkinter.Entry(cat_info_frame)
95 cat_name_entry.grid(row = 0, column = 1, padx=10, pady=10)
96
97 #cat's breed
98 cat_breed_label = tkinter.Label(cat_info_frame, text = 'Breed: ')
99 cat_breed_label.grid(row = 0, column = 2, padx=10, pady=10)
100 cat_breed_entry = tkinter.Entry(cat_info_frame)
101 cat_breed_combobox = ttk.Combobox(cat_info_frame, values=['','Siamese', 'British Shorthair', 'Maine Coon', 'Persian', 'Ragdoll', 'Sphynx',
102 'American Shorthair', 'Abyssinian', 'Exotic Shorthair', 'Scottish Fold', 'Burmese', 'Birman', 'Bombay', 'Devon Rex', 'Balinese', 'Javanese',
103 'Oriental Shorthair', 'Munchkin'])
104 cat_breed_combobox.grid(row = 0, column = 3, padx=10, pady=10)
105
106 #cat's age
107 cat_age_label = tkinter.Label(cat_info_frame, text = 'Age: ')
108 cat_age_label.grid(row = 0, column = 4, padx=10, pady=10)
109 cat_age_spinbox = tkinter.Spinbox(cat_info_frame, from_ = 1, to = 20)
110 cat_age_spinbox.grid(row = 0, column = 5, padx=10, pady=10)
111
112 #cat's medical condition
113 cat_medical_condition_label = tkinter.Label(cat_info_frame, text = 'Medical condition: ')
114 cat_medical_condition_label.grid(row = 1, column = 0, padx=10, pady=10)
115 cat_medical_condition_combobox = ttk.Combobox(cat_info_frame, values = ['', 'None'])
116 cat_medical_condition_combobox.grid(row = 1, column = 1, padx=10, pady=10)
117

```

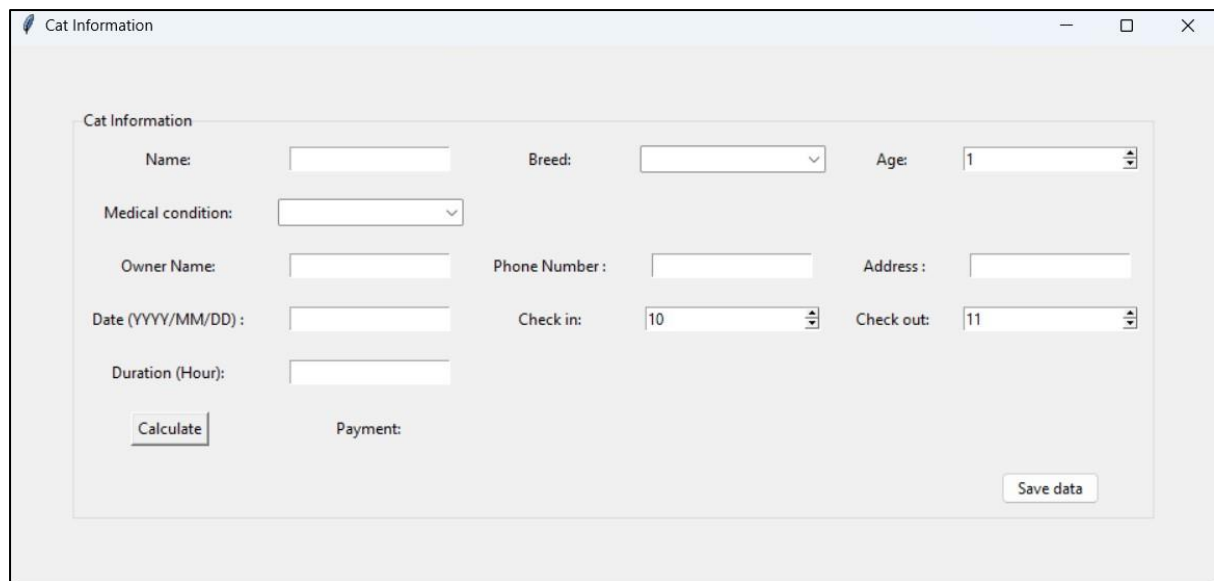
Figure 3.4 Examples of python code for attributes

```

175
176 button = ttk.Button(cat_info_frame, text='Save data', command=save_data)
177 button.grid(row=6, column=5, padx=10, pady=10)
178
179
180 root.mainloop()
181

```

Figure 3.5 Save button



The image shows a graphical user interface (GUI) window titled "Cat Information". The window contains a form with the following fields and controls:

- Name:** A text input field.
- Breed:** A dropdown menu.
- Age:** A spinner box with the value "1" displayed.
- Medical condition:** A dropdown menu.
- Owner Name:** A text input field.
- Phone Number :** A text input field.
- Address :** A text input field.
- Date (YYYY/MM/DD) :** A text input field.
- Check in:** A spinner box with the value "10" displayed.
- Check out:** A spinner box with the value "11" displayed.
- Duration (Hour):** A text input field.
- Calculate:** A button.
- Payment:** A text input field.
- Save data:** A button.

Figure 3.6 GUI interface for data entry

4.0 MYSQL DATABASE

The database is called 'cat_daycare_centre'. It contains 1 table named 'cat' with 13 key attributes pertaining to the data related to cat daycare service.

The attributes in the table are CAT_ID, CAT_NAME, CAT_BREED, CAT_AGE, CAT_MEDCON, DATE, CHECK IN, CHECK OUT, DURATION_HOUR, OWNER_NAME, PHONE_NUM, ADDRESS, and PAYMENT.

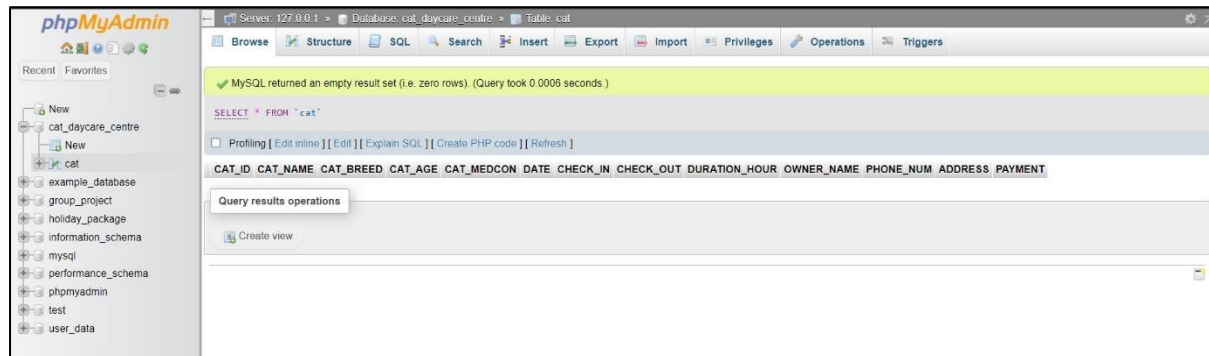


Figure 4.1 Table in cat_daycare_centre database

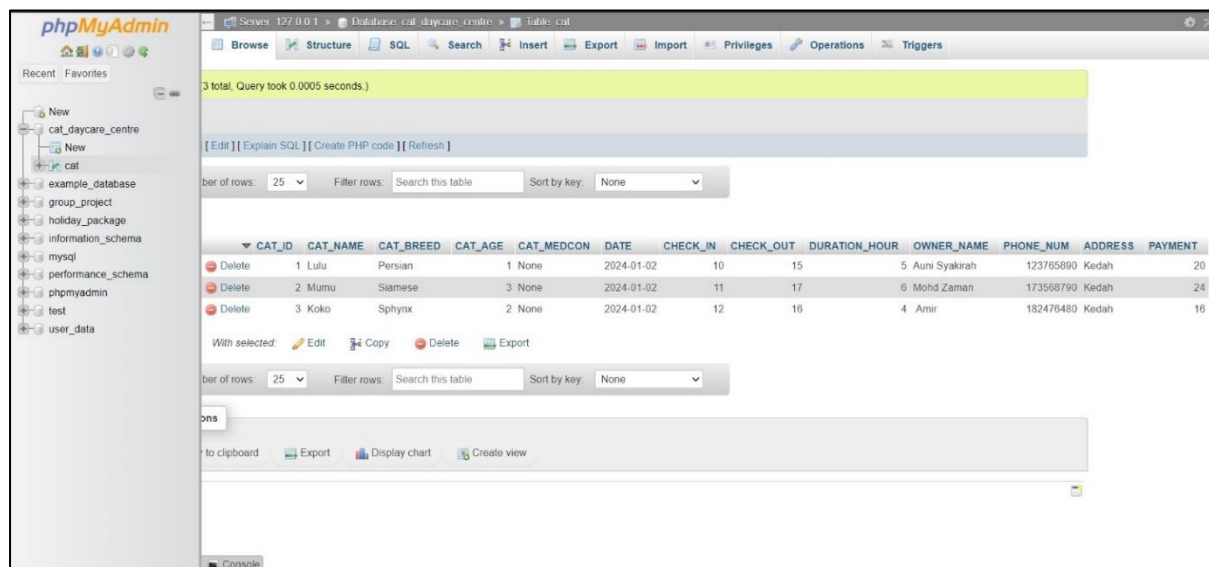


Figure 4.2 Examples of stored data in cat_daycare_centre database

phpMyAdmin

Server: 127.0.0.1 » Database: cat_daycare_centre » Table: cat

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 CAT_ID	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 CAT_NAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 CAT_BREED	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 CAT_AGE	int(10)			No	None			Change Drop More
<input type="checkbox"/>	5 CAT_MEDCON	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 DATE	date			No	None			Change Drop More
<input type="checkbox"/>	7 CHECK_IN	float			No	None			Change Drop More
<input type="checkbox"/>	8 CHECK_OUT	float			No	None			Change Drop More
<input type="checkbox"/>	9 DURATION_HOUR	float			No	None			Change Drop More
<input type="checkbox"/>	10 OWNER_NAME	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	11 PHONE_NUM	int(20)			No	None			Change Drop More
<input type="checkbox"/>	12 ADDRESS	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	13 PAYMENT	float			No	None			Change Drop More

☐ Check all
 With selected:
 ☐ Browse
 ☐ Change
 ☐ Drop
 ☐ Primary
 ☐ Unique
 ☐ Index
 ☐ Spatial
 ☐ Fulltext

Print
 Propose table structure
 Move columns
 Normalize

Add 1 column(s) after PAYMENT Go

Figure 4.3 Structure