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

DIPLOMA IN LIBRARY INFORMATIC

(IM 144)

PROGRAMMING FOR LIBRARIES

(IML208)

INDIVIDUAL ASSIGNMENT:

" BADMINTON RESERVATION SYSTEM"

PREPARED BY:

NURRUL HAIRIENA BINTI ARZA

(2022829308)

CLASS: KIM 1443E

PREPARED FOR:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE:

4 JANUARI 2024

INDIVIDUAL ASSIGNMENT:

" BADMINTON RESERVATION SYSTEM"

PREPARED BY:

NURRUL HAIRIENA BINTI ARZA

(2022829308)

CLASS: KIM 1443E

PREPARED FOR:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

SUBMISSION DATE:

4 JANUARI 2024

ACKNOWLEDGEMENT

Assalamualaikum warahamtullahi wabarakatuh

TABLE OF CONTENT

1.0 INTRODUCTION

The intended design, known as the Badminton Reservation System, aims to transform the way badminton players reserve courts. This intuitive programme seamlessly combines the power of MySQL for efficient data administration with the interactive graphical user interface provided by Python's tkinter module. The system provides players with an easier way to make court reservations, whether they are for themselves or for big tournaments. With convenience, players may choose their favourite court, input their details, and schedule the play time. The programme guarantees a reasonable and clear pricing structure by continuously calculating the overall cost and including reductions for extended play periods. This strategy seeks to reduce the lack of interaction between the participants and the system by facilitating a smoother reservation process. the Badminton Reservation System, a planned design with the goal of changing the way badminton players reserve courts. The capabilities of MySQL for effective data management and Python's tkinter module for an interactive graphical user interface are effortlessly combined in this user-friendly application. In order to make court reservations easier for players or any other customer alike, the system gives players an easier method. Players can select their preferred court, enter their information, and set the play time with ease. All the user data will be saved in the database for badminton reservations. Other than that, a fair and open price structure is ensured by the programme, which continually determines the total cost and includes decreases for longer play sessions. By supporting a smoother reservation procedure, this approach seeks to close the communication gap between the system and players.

2.0 FLOWCHART
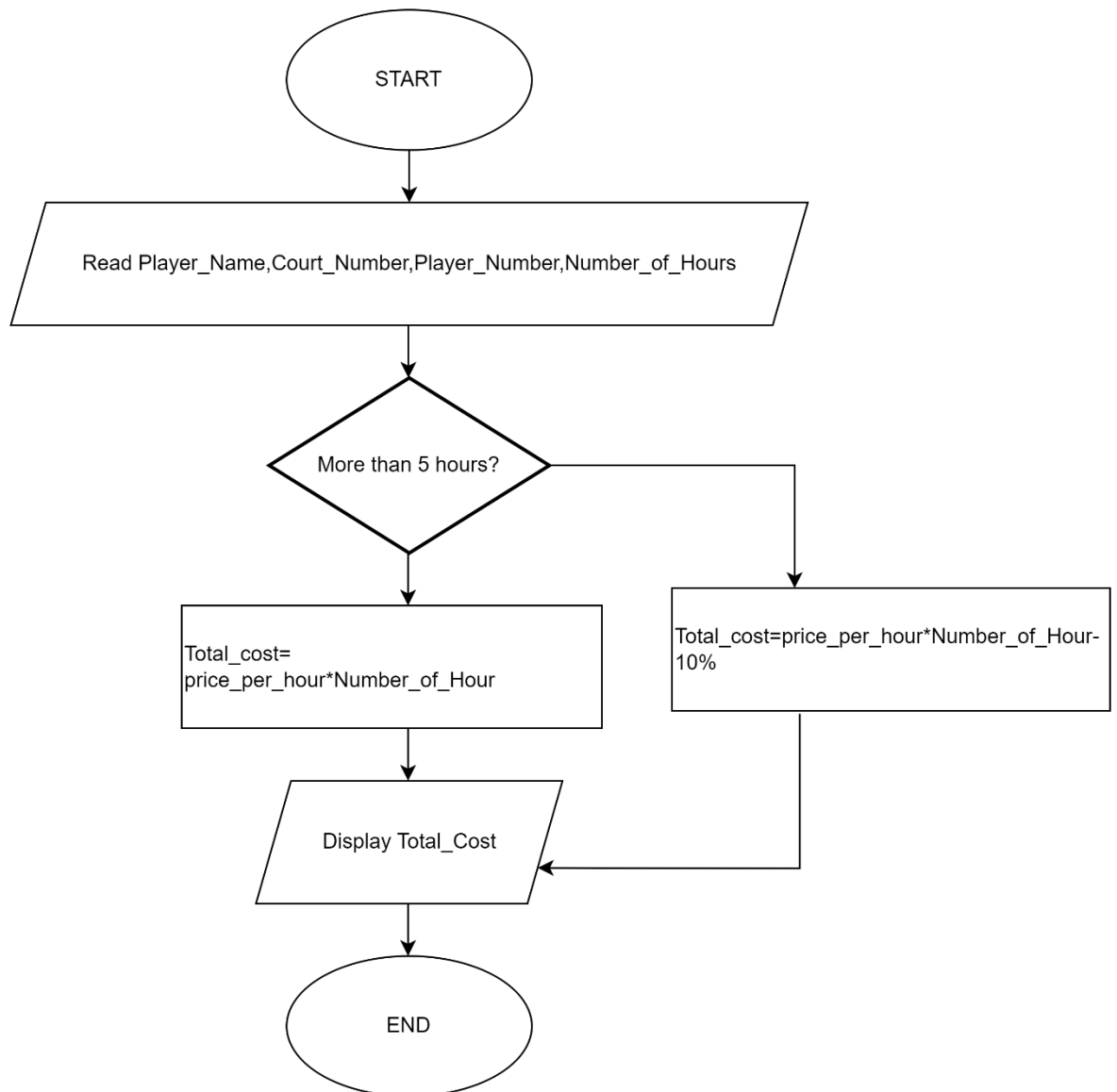
START

Read Player_Name,Court_Number,Player_Number,Number_of_Hours

More than 5 hours?

Total_cost=
price_per_hour*Number_of_Hour

Total_cost=price_per_hour*Number_of_Hour-10%

Display Total_Cost

END

3.0 SNAPSHOT OF PHYTON CODE

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

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

def calculate_cost():
    number_of_hours = int(number_of_hours_spinbox.get())
    number_of_players = int(number_of_players_spinbox.get())

    # price 15.00 per hour
    price_per_hour = 15.00

    # Calculate the total cost
    total_cost =price_per_hour * number_of_hours

    # Apply a 10% discount if the player plays more than 5 hours
    if number_of_hours > 5:
        discount_percentage = 10
        discount_amount = (total_cost * discount_percentage) / 100
        total_cost -= discount_amount

    cost_label.config(text=f"Total Price: RM{total_cost:.2f}", font=("Cooper Black", 12, "bold"))
    return total_cost
```

```python
def enter_data():
    player_name = player_name_entry.get()
    court_number = court_number_combobox.get()
    number_of_players = int(number_of_players_spinbox.get())
    number_of_hours = int(number_of_hours_spinbox.get())
    total_cost = calculate_cost()

    print("Player Name:", player_name)
    print("Court Number:", court_number)
    print("Number_of_players:", number_of_players)
    print("Number of Hours:", number_of_hours)
    print("Total Cost:", total_cost)
    print("----------------------------------")

    # Inserting data into the "court booking" table
    sql = "INSERT INTO court_booking (player_name, court_number, number_of_players, number_of_hours, total_cost) VALUES (%s, %s, %s, %s, %s)"
    values = (player_name, court_number, number_of_players, number_of_hours, total_cost)

    try:
        cursor.execute(sql, values)
        mydb.commit()
        print("Data inserted successfully!")
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        mydb.rollback()

#my window title
window = tk.Tk()
window.title("Badminton Reservation System",)
```

```python
67     frame = tk.Frame(window)
68     frame.pack()
69
70     # Player Info Frame
71     player_info_frame = tk.LabelFrame(frame, text="Court Booking", font=("Cooper Black",12),bg="olivedrab3")
72     player_info_frame.grid(row=0, column=0, padx=20, pady=10)
73
74     # Price Information
75     prices_text = tk.Text(player_info_frame, height=6, width=73)
76     prices_text.grid(row=0, column=0, columnspan=7, pady=10)
77
78     prices_text.insert(tk.END, "Prices per hour:\n")
79     prices_text.insert(tk.END, "1 hour: RM15.00\n")
80     prices_text.insert(tk.END, "Above than 5 hour discount 10%\n")
81     prices_text.configure(state='disabled')
82
83     # Player Information
84     tk.Label(player_info_frame, text="Player Name", font=("Cooper Black", 10)).grid(row=1, column=0)
85     player_name_entry = tk.Entry(player_info_frame, font=("Jumbo", 10, "bold"))
86     player_name_entry.grid(row=1, column=1, pady=4)
87
88     tk.Label(player_info_frame, text="Court Number", font=("Cooper Black", 10)).grid(row=1, column=2)
89     court_number_combobox = ttk.Combobox(player_info_frame, values=["C1", "C2", "C3", "C4"], font=("Jumbo", 10, "bold"))
90     court_number_combobox.grid(row=1, column=3, columnspan=2, pady=4)
91
92     tk.Label(player_info_frame, text="Number of hours", font=("Cooper Black", 10)).grid(row=3, column=2, pady=7)
93     number_of_hours_spinbox = ttk.Spinbox(player_info_frame, from_=1, to=10, font=("Jumbo", 10, "bold"))
94     number_of_hours_spinbox.grid(row=3, column=3,columnspan=2,pady=4)
95
96     tk.Label(player_info_frame, text="Player Number", font=("Cooper Black", 10)).grid(row=3, column=0, pady=8)
97     number_of_players_spinbox = ttk.Spinbox(player_info_frame, from_=1, to=20, font=("Jumbo", 10, "bold"))
98     number_of_players_spinbox.grid(row=3, column=1, pady=4)
```
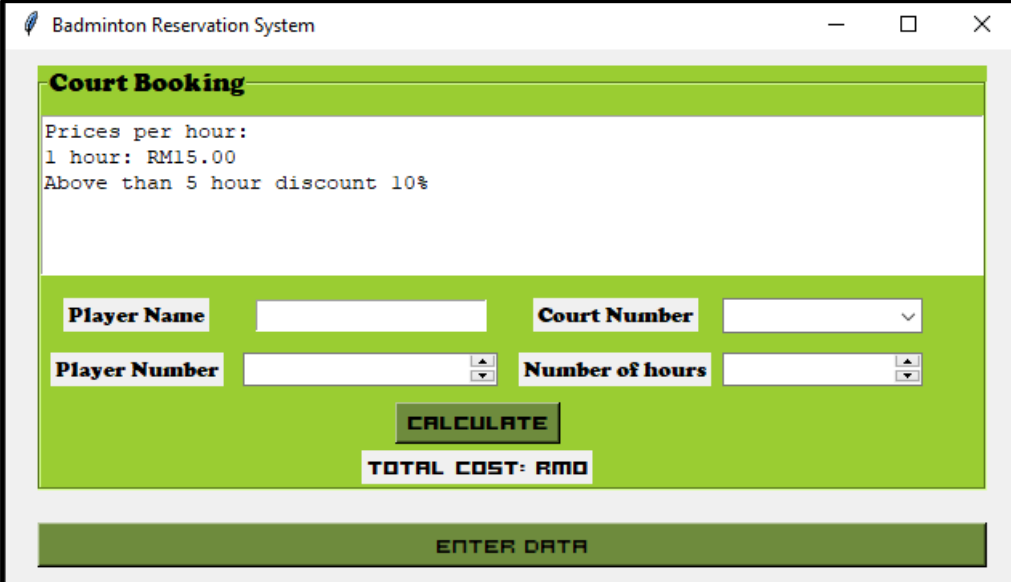
```python
97     number_of_players_spinbox = ttk.Spinbox(player_info_frame, from_=1, to=20, font=("Jumbo", 10, "bold"))
98     number_of_players_spinbox.grid(row=3, column=1, pady=4)
99
100    # Cost Calculation
101    calculate_button = tk.Button(player_info_frame, text="Calculate", command=calculate_cost, font=("hooge 05_53", 9, "bold",),bg="darkolivegreen4")
102    calculate_button.grid(row=6, column=1, columnspan=2, pady=2)
103
104    cost_label = tk.Label(player_info_frame, text="Total Cost: RM0", font=("hooge 05_53", 9, "bold"))
105    cost_label.grid(row=7, column=1, columnspan=2, pady=2)
106
107    # Button inserts data into my SQL
108    button = tk.Button(frame, text="Enter data", command=enter_data,font=("hooge 05_53",10), bg="darkolivegreen4")
109    button.grid(row=3, column=0, sticky="news", padx=20, pady=10)
110
111    window.mainloop()
112
```

4.0 SNAPSHOT OF GUI

5.0 SNAPSHOT OF DATABASE