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

DIPLOMA IN INFORMATICS LIBRARY
(IM144)

PROGRAMMING FOR LIBRARIES
(IML208)

ASSIGNMENT 1: INDIVIDUAL PROJECT

Prepared by:

NUR HIDAYATI BINTI JAKARIA
(2022843754)

GROUP: KCDIM1443E

Prepared for:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

Submission date:

4th JANUARY 2024

ASSIGNMENT 1: INDIVIDUAL PROJECT

PREPARED BY:

NUR HIDAYATI BINTI JAKARIA

(2022843754)

GROUP: KCDIM1443E

CDIM144 – DIPLOMA IN INFORMATICS LIBRARY

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVESITI TEKNOLOGI MARA (UITM)

MERBOK, KEDAH

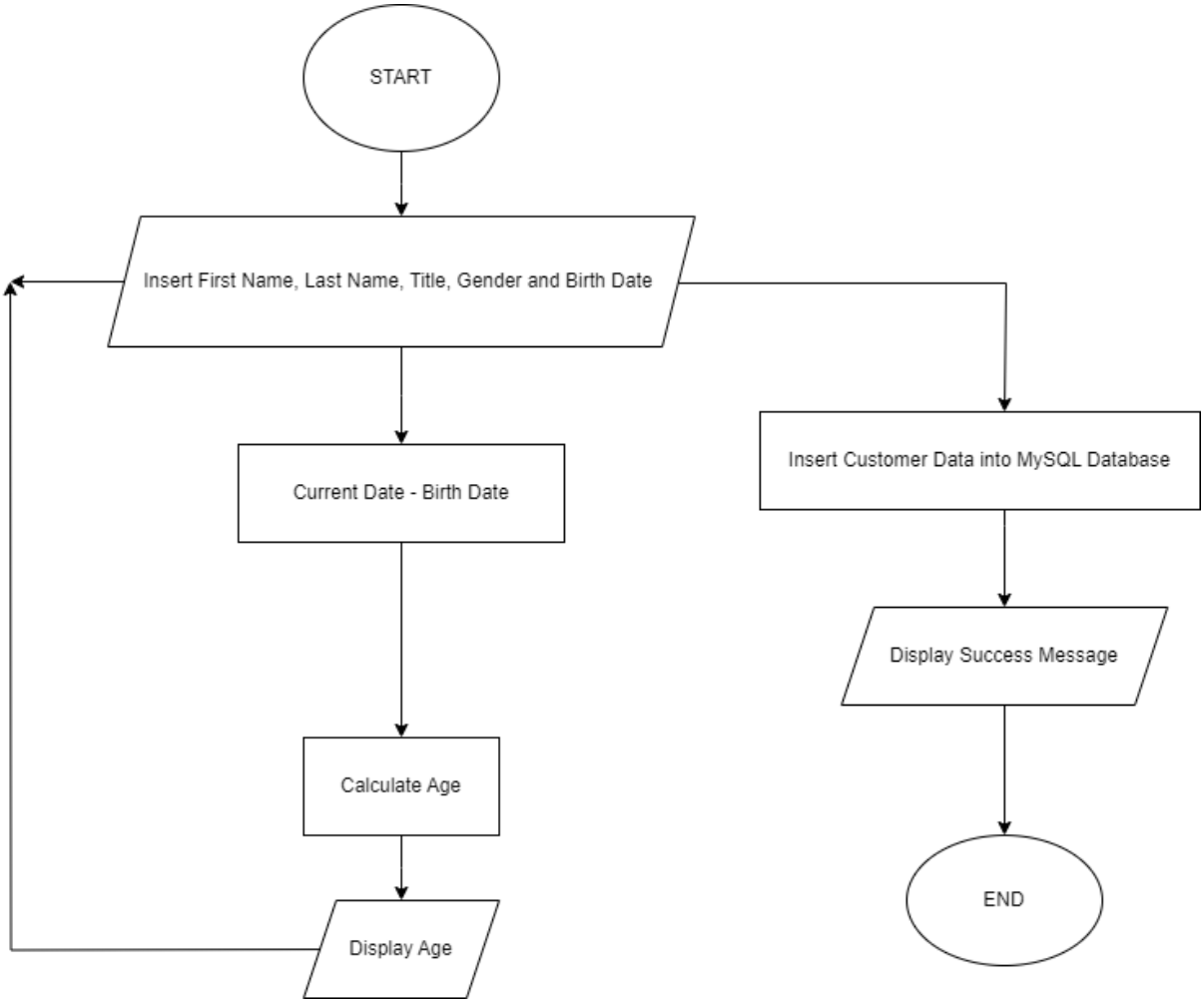
TABLE OF CONTENT

INTRODUCTION

The title of my individual project is Stationery Membership Registration. This project is about how customers can register their information to get a membership at a stationery shop and get benefits from it. There are several attributes that the customers have to insert. For instance, like their first name, last name, title (Mr., Ms., or Mrs.), gender, their date of birth, and also their age. In this project, I insert a calculation to get the customers' age by their date of birth.

After they have already entered all the qualifications given, their personal information will be transferred to the database. To insert their information into the database, they have to click enter data, and the information will appear in the database. If the data is successfully inserted into the database, there will be a message box that says "Data Inserted Successfully." With this message box, customers can know that their information has already been inserted into the system.

FLOWCHART


START
Insert First Name, Last Name, Title, Gender and Birth Date
Current Date - Birth Date
Calculate Age
Display Age
Insert Customer Data into MySQL Database
Display Success Message
END

SNAPSHOOT OF CODE

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

def insert_data():
    cus_first_name = first_name_entry.get()
    cus_last_name = last_name_entry.get()
    cus_title = title_combobox.get()
    cus_gender = gender_combobox.get()
    cus_birth_day = dayField.get()
    cus_birth_month = monthField.get()
    cus_birth_year = yearField.get()
    cus_age = age_year_entry.get()

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

    # Create a cursor object to execute SQL queries
    mycursor = mydb.cursor()

    # SQL query to insert data into the table
    insert_query = "INSERT INTO customer (Cus_First_Name, Cus_Last_Name,
Cus_Title, Cus_Gender, Cus_Birth_Day, Cus_Birth_Month, Cus_Birth_Year,
Cus_Age) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"

    # Execute the query with the data
    mycursor.execute(insert_query, (cus_first_name, cus_last_name, cus_title,
cus_gender, cus_birth_day, cus_birth_month, cus_birth_year, cus_age))

    # Commit the changes to the database
    mydb.commit()

    mycursor.close()
    mydb.close()

    messagebox.showinfo("Success", "Data inserted successfully!")

# function for checking error
def checkError() :

    # if any of the entry field is empty
```

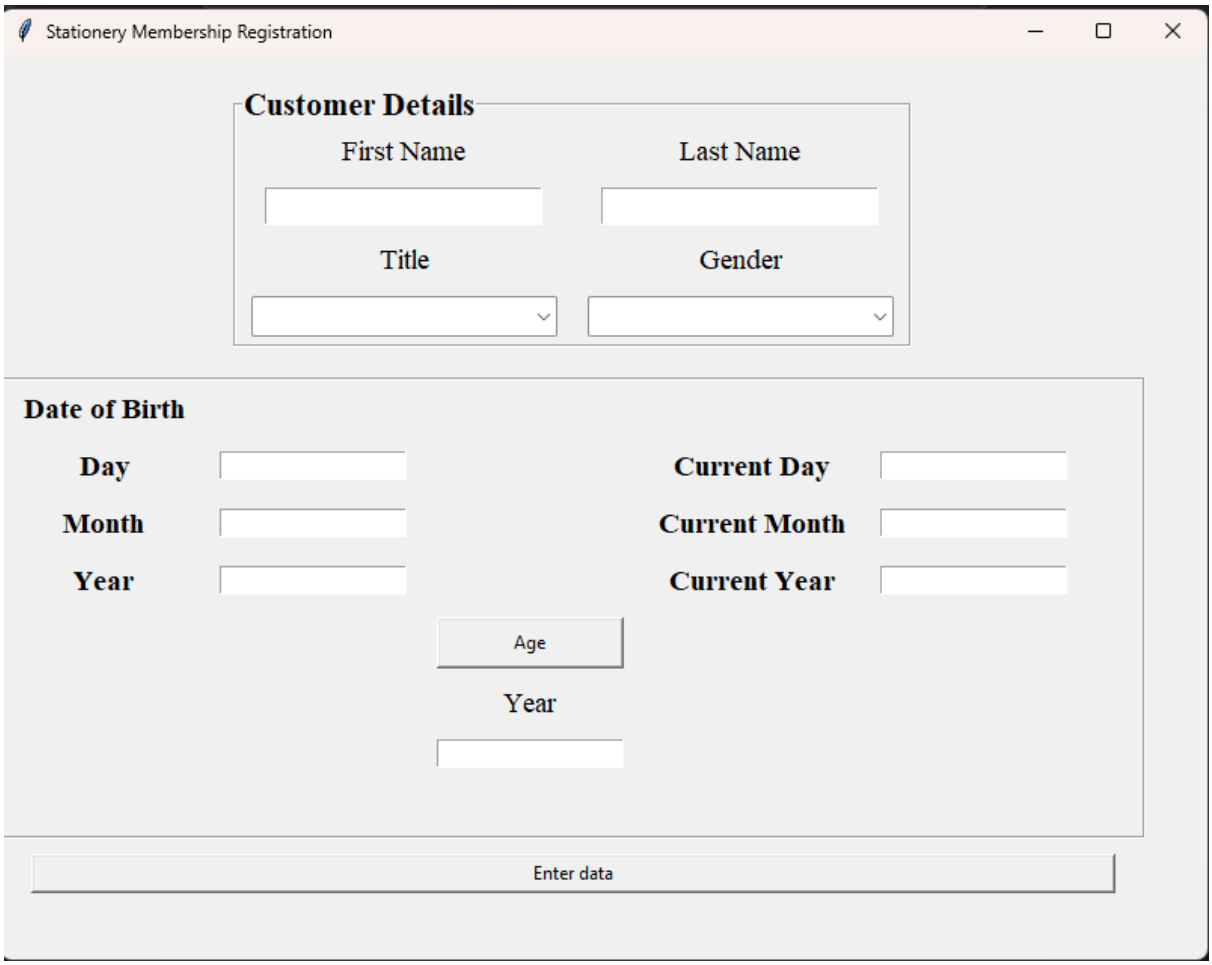
```
    # then show an error message and clear
    # all the entries
    if (dayField.get() == "" or monthField.get() == ""
        or yearField.get() == "") :

        # show the error message
        messagebox.showerror("Input Error")

        return -1


# function to calculate Age
def calculate_age() :

    # check for error
    value = checkError()

    # if error is occur then return
    if value == -1 :
        return

    else :

        # take a value from the respective entry boxes get method returns
current text as string
        birth_day = int(dayField.get())
        birth_month = int(monthField.get())
        birth_year = int(yearField.get())

        given_day = int(current_day.get())
        given_month = int(current_month.get())
        given_year = int(current_year.get())

        # if birth date is greater then given birth_month then do not count
this month and add 30 to the date so as to subtract the date and get the
remaining days
        month =[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

        if (birth_day > given_day):
            given_month = given_month - 1
            given_day = given_day + month[birth_month-1]


        # if birth month exceeds given month, then do not count this year and
add 12 to the month so that we can subtract and find out the difference
        if (birth_month > given_month):
            given_year = given_year - 1
            given_month = given_month + 12
```

```
        # calculate day, month, year
        calculated_day = given_day - birth_day;
        calculated_month = given_month - birth_month;
        calculated_year = given_year - birth_year;

        # calculated day, month, year write back to the respective entry boxes

        # insert method inserting the value in the text entry box.

        age_year_entry.insert(10, str(calculated_year))


        # if birth date is greater then given birth_month then donot count
this month and add 30 to the date so as to subtract the date and get the
remaining days
        month =[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

        if (birth_day > given_day):
            given_month = given_month - 1
            given_day = given_day + month[birth_month-1]


        # if birth month exceeds given month, then do not count this year and
add 12 to the month so that we can subtract and find out the difference
        if (birth_month > given_month):
            given_year = given_year - 1
            given_month = given_month + 12

        # calculate day, month, year
        calculated_day = given_day - birth_day;
        calculated_month = given_month - birth_month;
        calculated_year = given_year - birth_year;

        # calculated day, month, year write back to the respective entry boxes

        # insert method inserting the value in the text entry box.

        age_year_entry.insert(10, str(calculated_year))

root = tk.Tk()
root.title("Stationery Membership Registration")
root.geometry("800x600")

# Customer Details
customer_frame = tk.LabelFrame(root, text= "Customer Details", font= ('Times
New Roman',16, 'bold'))
customer_frame.grid(row=0, column=0, padx=20, pady=20)
```

```
first_name_label = tk.Label(customer_frame, text="First Name", font=('Times
New Roman',14))
first_name_label.grid(row=0, column=0)
first_name_entry = tk.Entry(customer_frame, font=('Times New Roman',14))
first_name_entry.grid(row=1, column=0)

last_name_label = tk.Label(customer_frame, text="Last Name", font=('Times New
Roman',14))
last_name_label.grid(row=0, column=1)
last_name_entry = tk.Entry(customer_frame, font=('Times New Roman',14))
last_name_entry.grid(row=1, column=1)

title_label = tk.Label(customer_frame, text="Title", font=('Times New
Roman',14))
title_label.grid(row=2, column=0)
title_combobox = ttk.Combobox(customer_frame, values=["Mr.", "Ms.", "Mrs."],
font=('Times New Roman',14))
title_combobox.grid(row=3, column=0)

gender_label = tk.Label(customer_frame, text="Gender", font=('Times New
Roman',14))
gender_label.grid(row=2, column=1)
gender_combobox = ttk.Combobox(customer_frame, values=["Female", "Male"],
font=('Times New Roman',14))
gender_combobox.grid(row=3, column=1)

for widget in customer_frame.winfo_children():
    widget.grid_configure(padx=10, pady=5)

# Date of Birth details
birth_date_frame = tk.LabelFrame(root)
birth_date_frame.grid(row=1, column=0, ipadx=20, ipady=20)

birth_date_label = tk.Label(birth_date_frame, text= "Date of Birth",
font=('Times New Roman',14, 'bold'))
birth_date_label.grid(row=1, column=0)

# Date of birth
birth_date = tk.Label(birth_date_frame, text="Day", font=('Times New
Roman',14, 'bold'))
birth_date.grid(row = 2, column = 0)
birth_month = tk.Label(birth_date_frame, text="Month", font=('Times New
Roman',14, 'bold'))
birth_month.grid(row = 3, column = 0)
birth_year = tk.Label(birth_date_frame, text="Year", font=('Times New
Roman',14, 'bold'))
birth_year.grid(row = 4, column = 0)
```

```
# Create a text entry box for filling or typing the information(dob).
dayField = tk.Entry(birth_date_frame)
dayField.grid(row = 2, column = 1)
monthField = tk.Entry(birth_date_frame)
monthField.grid(row = 3, column = 1)
yearField = tk.Entry(birth_date_frame)
yearField.grid(row = 4, column = 1)

# Current Year
curr_day = tk.Label(birth_date_frame, text= "Current Day", font=('Times New
Roman',14, 'bold'))
curr_day.grid(row=2, column=3)
curr_month = tk.Label(birth_date_frame, text= "Current Month", font=('Times
New Roman',14, 'bold'))
curr_month.grid(row=3, column=3)
curr_year = tk.Label(birth_date_frame, text= "Current Year", font=('Times New
Roman',14, 'bold'))
curr_year.grid(row=4, column=3)

# Create a text entry box for filling or typing the information(current year).
current_day = tk.Entry(birth_date_frame)
current_day.grid(row = 2, column = 4)
current_month = tk.Entry(birth_date_frame)
current_month.grid(row = 3, column = 4)
current_year = tk.Entry(birth_date_frame)
current_year.grid(row = 4, column = 4)

# Age results
resultantAge = tk.Button(birth_date_frame, text = "Age", command =
calculate_age, padx=25, pady=5)
resultantAge.grid(row=5, column=2, sticky= "news")

age_year = tk.Label(birth_date_frame, text= "Year", font=('Times New
Roman',14))
age_year.grid(row=6, column=2, sticky= "news")
age_year_entry = tk.Entry(birth_date_frame)
age_year_entry.grid(row=7, column=2, sticky= "news")

for widget in birth_date_frame.winfo_children():
    widget.grid_configure(padx=10, pady=5)

# Button
button = tk.Button(root, text= "Enter data", command= insert_data)
button.grid(row=3, column=0, sticky= "news", padx=20, pady=10)

root.mainloop()
```

SNAPSHOOT OF GUI



SNAPSHOOT OF DATABASE

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 31, 2023 at 02:46 PM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `stationery membership registration`
--

-- --------------------------------------------------------

--
-- Table structure for table `customer`
--

CREATE TABLE `customer` (
  `Cus_First_Name` varchar(30) NOT NULL,
  `Cus_Last_Name` varchar(30) NOT NULL,
  `Cus_Title` varchar(3) NOT NULL,
  `Cus_Gender` varchar(6) NOT NULL,
  `Cus_Birth_Day` int(2) NOT NULL,
  `Cus_Birth_Month` int(2) NOT NULL,
  `Cus_Birth_Year` int(4) NOT NULL,
  `Cus_Age` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `customer`
--

INSERT INTO `customer` (`Cus_First_Name`, `Cus_Last_Name`, `Cus_Title`,
`Cus_Gender`, `Cus_Birth_Day`, `Cus_Birth_Month`, `Cus_Birth_Year`, `Cus_Age`)
VALUES
```

```
('Hidayati', 'Jakaria', 'Ms.', 'Female', 2, 8, 2004, 19),
('Hidayati', 'Jakaria', 'Ms.', 'Female', 2, 8, 2004, 19),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('cus_first_name', 'cus_last_name', 'cus', 'cus_ge', 0, 0, 0, 0),
('Cus_First_Name', 'Cus_Last_Name', 'Cus', 'Cus_Ge', 0, 0, 0, 0),
('DANIAL', 'JAKARIA', 'Mr.', 'Male', 12, 4, 2008, 15),
('YUSNAH', 'YUSOF', 'Mrs', 'Female', 19, 10, 1974, 49);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```