UNIVERSITI TEKNOLOGI MARA (UiTM) CAWANGAN KEDAH

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

(CDIM144)

PROGRAMMING FOR LIBRARIES [IML 208]

INDIVIDUAL ASSIGNMENT

PREPARED BY:

IQMA ALEEYA ZAHARA BINTI ABDUL HALIM (2023602626)

KCDIM 144 3E

PREPARED FOR:

MOHD FIRDAUS BIN MOHD HELMI

INDIVIDUAL ASSIGNMENT: LIBRARY MEMBERSHIP REGISTRATION PROGRAMME

PREPARED BY:

IQMA ALEEYA ZAHARA BINTI ABDUL HALIM

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UiTM) CAWANGAN KEDAH

DIPLOMA IN LIBRARY INFORMATICS

ACKNOWLEDGEMENT

In the name of Allah, the Most Merciful, I want to show my gratitude to The Creators for easing everything and help me in all ways towards completing this assignment, without his grace I will not be completing this ease fully.

I would like to show my appreciation to my lecturer in this Programming for Libraries [Iml 208], Sir Mohd Firdaus Bin Mohd Helmi for guiding me throughout this assignment. He helps me a lot by explaining earnestly, being flexible and passionately lecturing us that makes us easily understand this subject. It all thanks to him that I finally completed this assignment.

I am also grateful to my classmates and course mates for helping me figure things out together, sharing as much information possible, they also gave me mental support and I am so grateful for that.

Lastly, I want to thank my lecturer in advance for remarking on my assignment and I am sorry if I made any mistake here. I will make sure to work hard on improving myself and correct my mistake in the future.

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 behaviour 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 : IQMA ALEEYA ZAHARA BINTI ABDUL HALIM
Matric Number : 2023602626
Course Code : IML208
Programme Code :-
Faculty / Campus : UiTM Kampus Sungai Petani

TABLE OF CONTENT

1.0 INTRODUCTION

The "Library Membership Registration" project is designed to facilitate efficient management of library members through a comprehensive system that encompasses essential CRUD (Create, Read, Update, Delete) functionalities.

This program aims to streamline the membership registration process by enabling librarians to easily add new members, view existing records, update information, and remove members as needed.

In addition to managing member data, the system also incorporates financial capabilities, allowing for the calculation of total and average membership fees, which can inform budgeting and financial reporting. Emphasizing data integrity and scalability, the project ensures that member IDs are unique during creation, and it is structured to accommodate future enhancements, such as additional member attributes or functionalities.

Overall, this system serves not only as a practical tool for library management but also as an educational resource, fostering an understanding of CRUD operations and data management principles in software development.

Project Name: Library membership registration

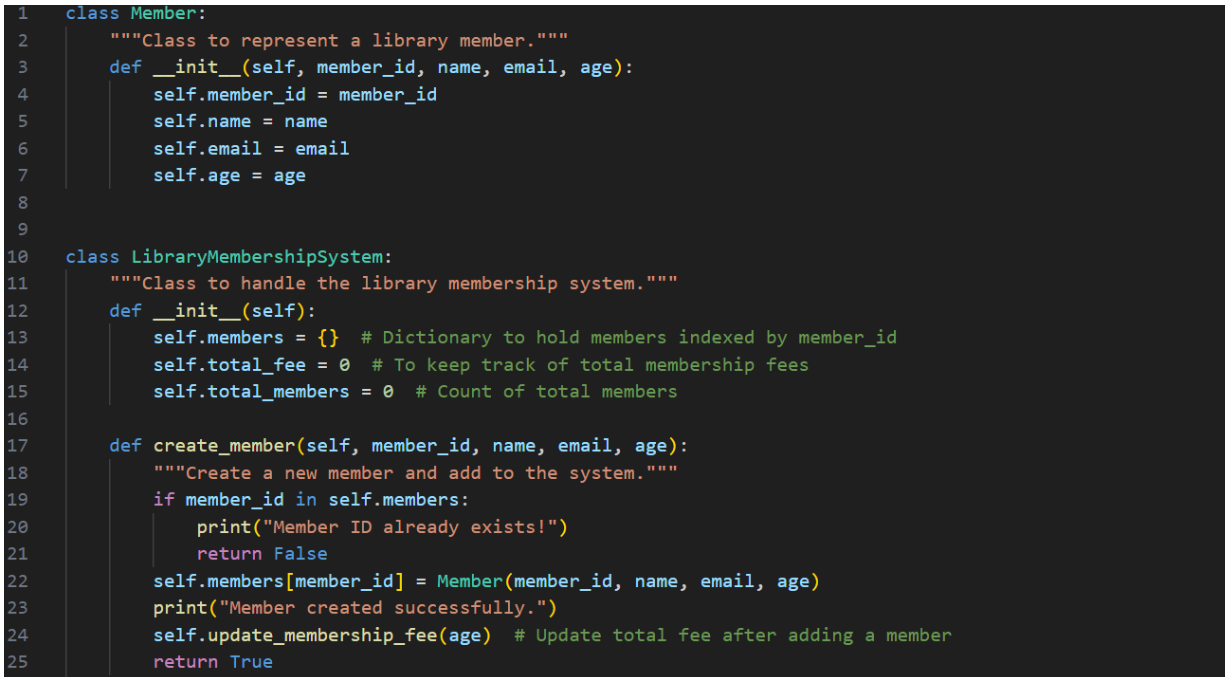
File name: library_membership.py

2.0 PROMPT DATA

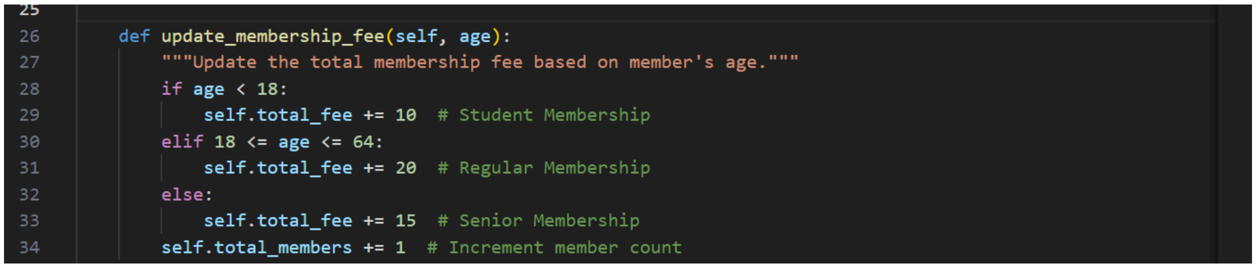
- Member ID
- Member Name
- Member Email
- Member Age
- Membership Fee

3.0 FUNCTION

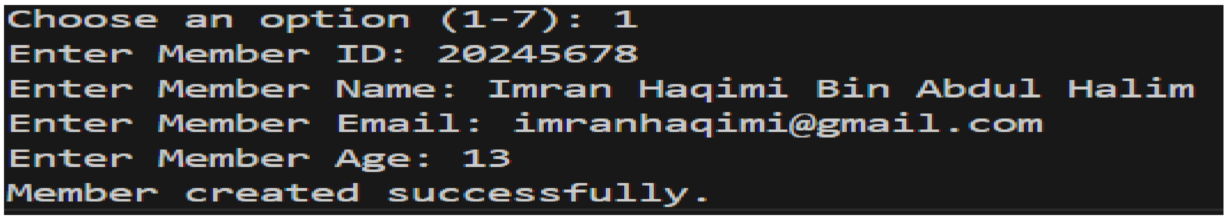
i. Create data: Adds a new member after checking if the `member_id` already exists.



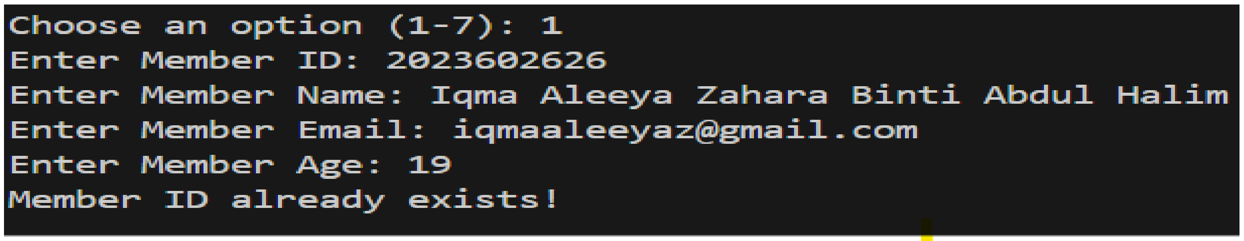
```python
class Member:
    """Class to represent a library member."""
    def __init__(self, member_id, name, email, age):
        self.member_id = member_id
        self.name = name
        self.email = email
        self.age = age


class LibraryMembershipSystem:
    """Class to handle the library membership system."""
    def __init__(self):
        self.members = {}  # Dictionary to hold members indexed by member_id
        self.total_fee = 0  # To keep track of total membership fees
        self.total_members = 0  # Count of total members

    def create_member(self, member_id, name, email, age):
        """Create a new member and add to the system."""
        if member_id in self.members:
            print("Member ID already exists!")
            return False
        self.members[member_id] = Member(member_id, name, email, age)
        print("Member created successfully.")
        self.update_membership_fee(age)  # Update total fee after adding a member
        return True
```



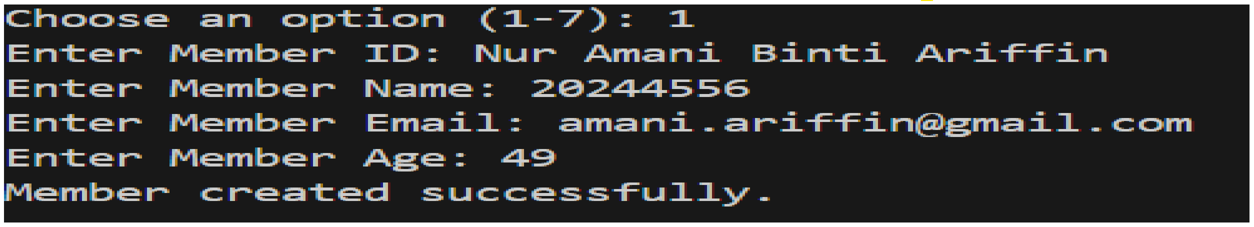
```python
    def update_membership_fee(self, age):
        """Update the total membership fee based on member's age."""
        if age < 18:
            self.total_fee += 10  # Student Membership
        elif 18 <= age <= 64:
            self.total_fee += 20  # Regular Membership
        else:
            self.total_fee += 15  # Senior Membership
        self.total_members += 1  # Increment member count
```



```
Choose an option (1-7): 1
Enter Member ID: 20245678
Enter Member Name: Imran Haqimi Bin Abdul Halim
Enter Member Email: imranhaqimi@gmail.com
Enter Member Age: 13
Member created successfully.
```

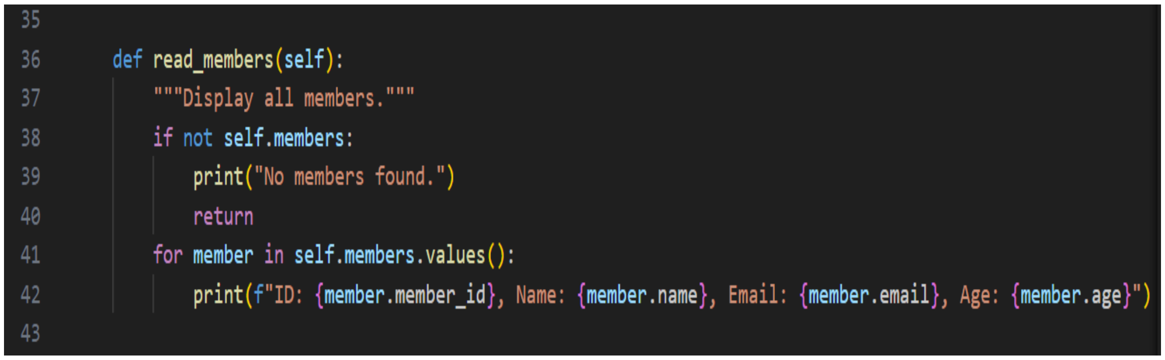


```
Choose an option (1-7): 1
Enter Member ID: 2023602626
Enter Member Name: Iqma Aleeya Zahara Binti Abdul Halim
Enter Member Email: iqmaaleeyaz@gmail.com
Enter Member Age: 19
Member ID already exists!
```

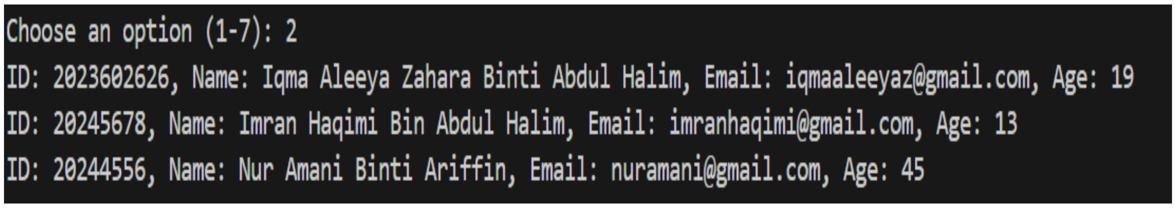


```
Choose an option (1-7): 1
Enter Member ID: Nur Amani Binti Ariffin
Enter Member Name: 20244556
Enter Member Email: amani.ariffin@gmail.com
Enter Member Age: 49
Member created successfully.
```

ii. Read data: - `read_members`: Displays all members in the system.

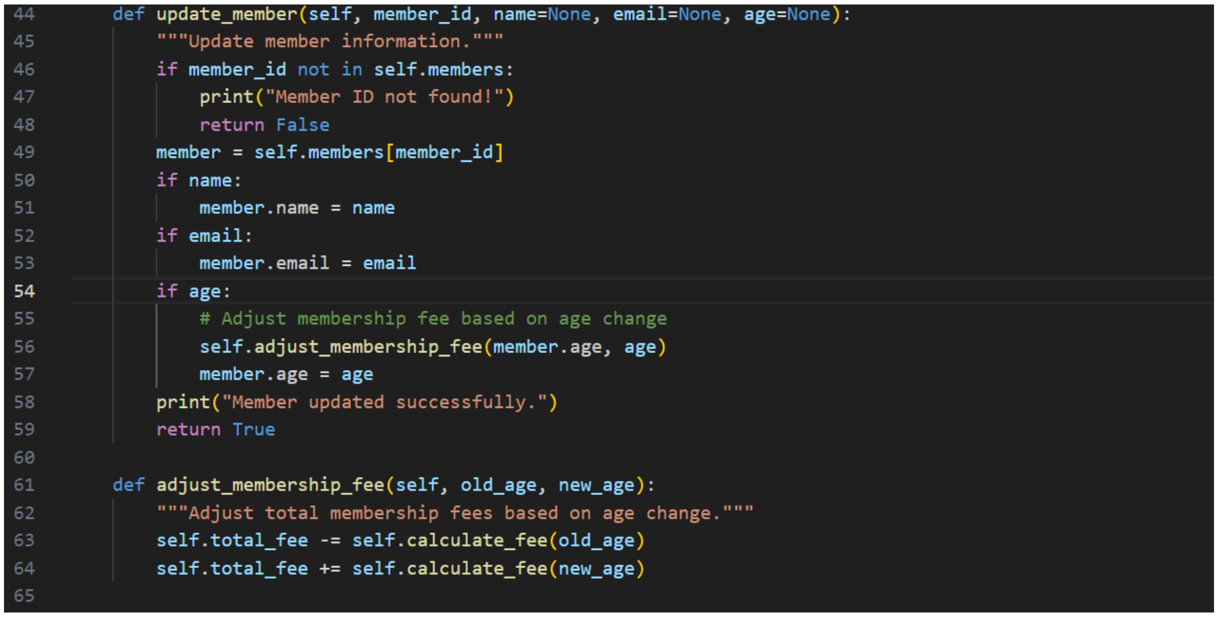


```python

def read_members(self):
    """Display all members."""
    if not self.members:
        print("No members found.")
        return
    for member in self.members.values():
        print(f"ID: {member.member_id}, Name: {member.name}, Email: {member.email}, Age: {member.age}")

```

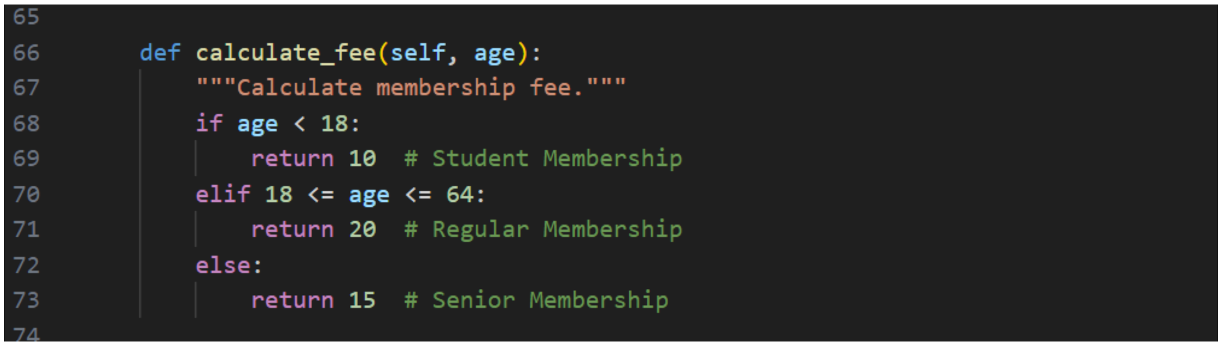


```
Choose an option (1-7): 2
ID: 2023602626, Name: Iqma Aleeya Zahara Binti Abdul Halim, Email: iqmaaleeyaz@gmail.com, Age: 19
ID: 20245678, Name: Imran Haqimi Bin Abdul Halim, Email: imranhaqimi@gmail.com, Age: 13
ID: 20244556, Name: Nur Amani Binti Ariffin, Email: nuramani@gmail.com, Age: 45
```

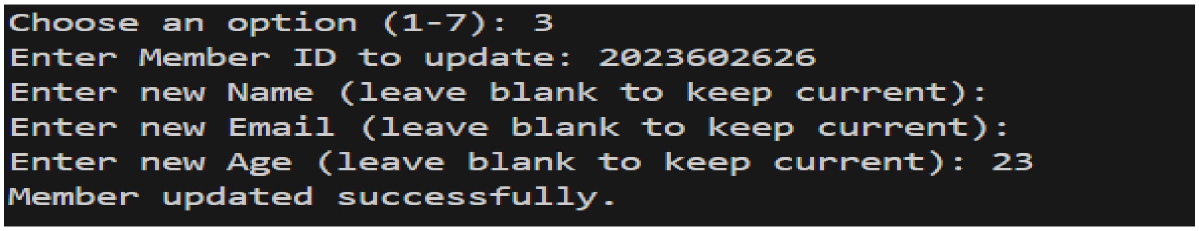
iii. **Update data:** Modify the member's information



```python
    def update_member(self, member_id, name=None, email=None, age=None):
        """Update member information."""
        if member_id not in self.members:
            print("Member ID not found!")
            return False
        member = self.members[member_id]
        if name:
            member.name = name
        if email:
            member.email = email
        if age:
            # Adjust membership fee based on age change
            self.adjust_membership_fee(member.age, age)
            member.age = age
        print("Member updated successfully.")
        return True

    def adjust_membership_fee(self, old_age, new_age):
        """Adjust total membership fees based on age change."""
        self.total_fee -= self.calculate_fee(old_age)
        self.total_fee += self.calculate_fee(new_age)

```

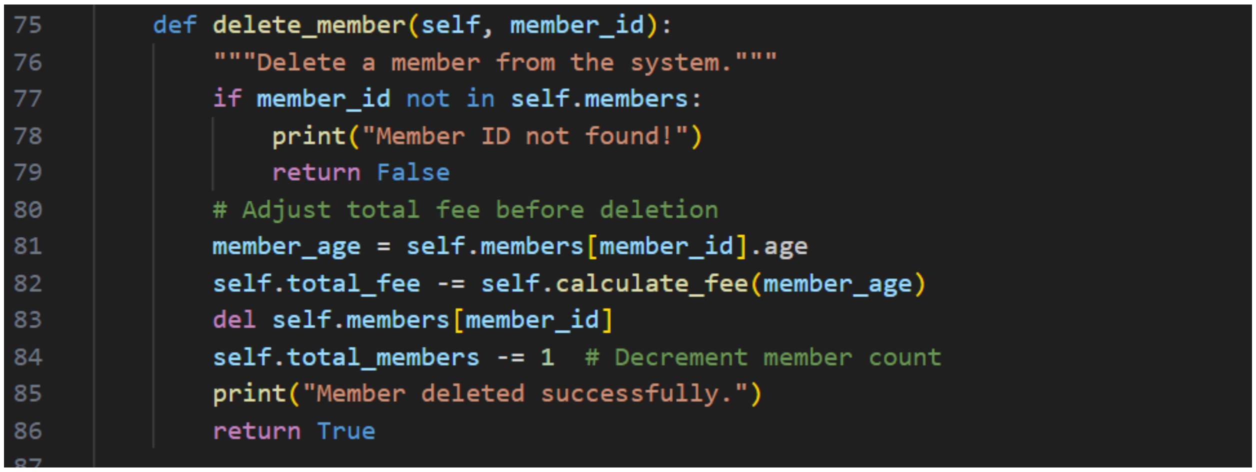


```python

    def calculate_fee(self, age):
        """Calculate membership fee."""
        if age < 18:
            return 10  # Student Membership
        elif 18 <= age <= 64:
            return 20  # Regular Membership
        else:
            return 15  # Senior Membership

```

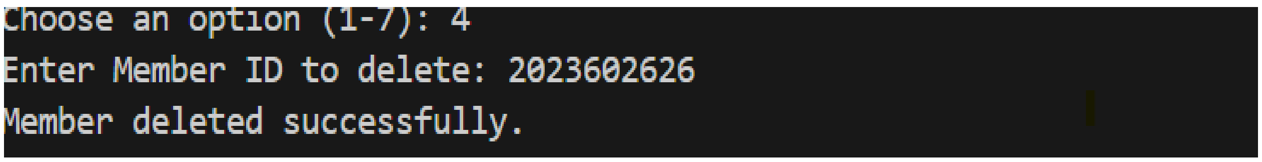


```
Choose an option (1-7): 3
Enter Member ID to update: 2023602626
Enter new Name (leave blank to keep current):
Enter new Email (leave blank to keep current):
Enter new Age (leave blank to keep current): 23
Member updated successfully.
```

iv. **Delete existing data:** Removes a member from the system.

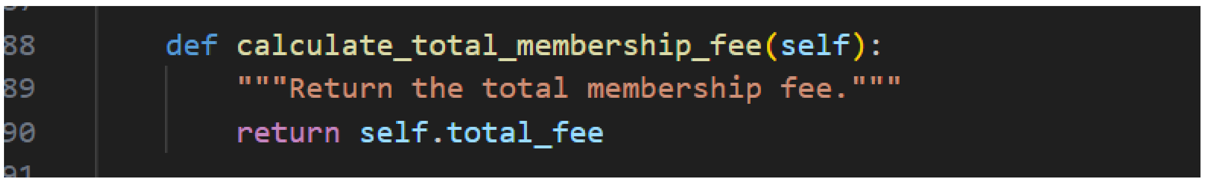


```python
    def delete_member(self, member_id):
        """Delete a member from the system."""
        if member_id not in self.members:
            print("Member ID not found!")
            return False
        # Adjust total fee before deletion
        member_age = self.members[member_id].age
        self.total_fee -= self.calculate_fee(member_age)
        del self.members[member_id]
        self.total_members -= 1  # Decrement member count
        print("Member deleted successfully.")
        return True
```

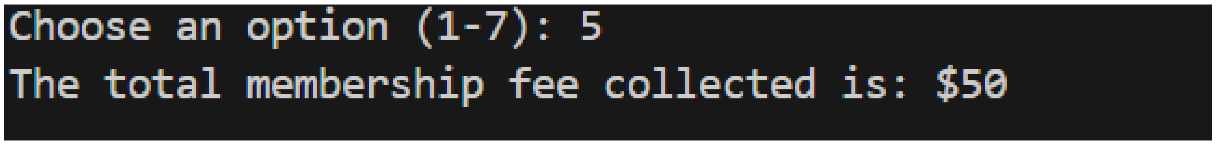


```
Choose an option (1-7): 4
Enter Member ID to delete: 2023602626
Member deleted successfully.
```

v. **Calculate Total Membership Fees:** Sum of all membership

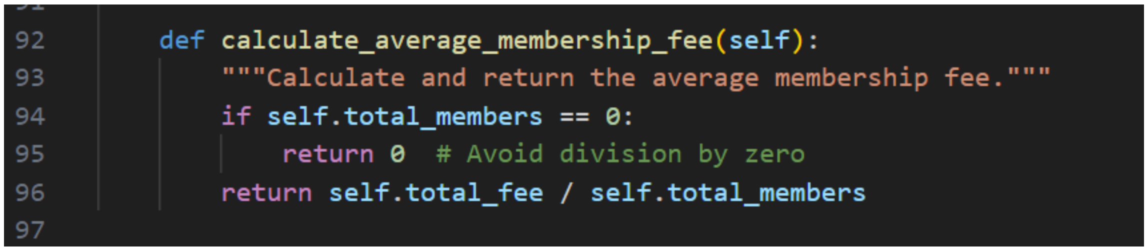


```python
    def calculate_total_membership_fee(self):
        """Return the total membership fee."""
        return self.total_fee
```

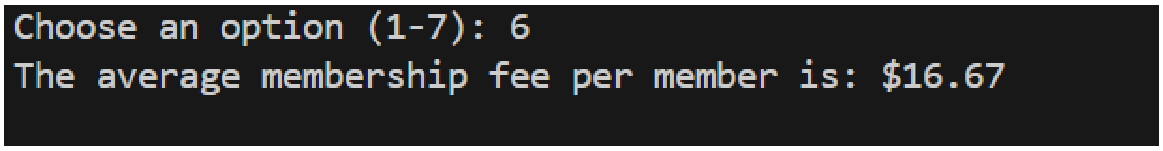


```
Choose an option (1-7): 5
The total membership fee collected is: $50
```

vi. **Calculate Average Membership Fee:** Total fees divided by the number of members.

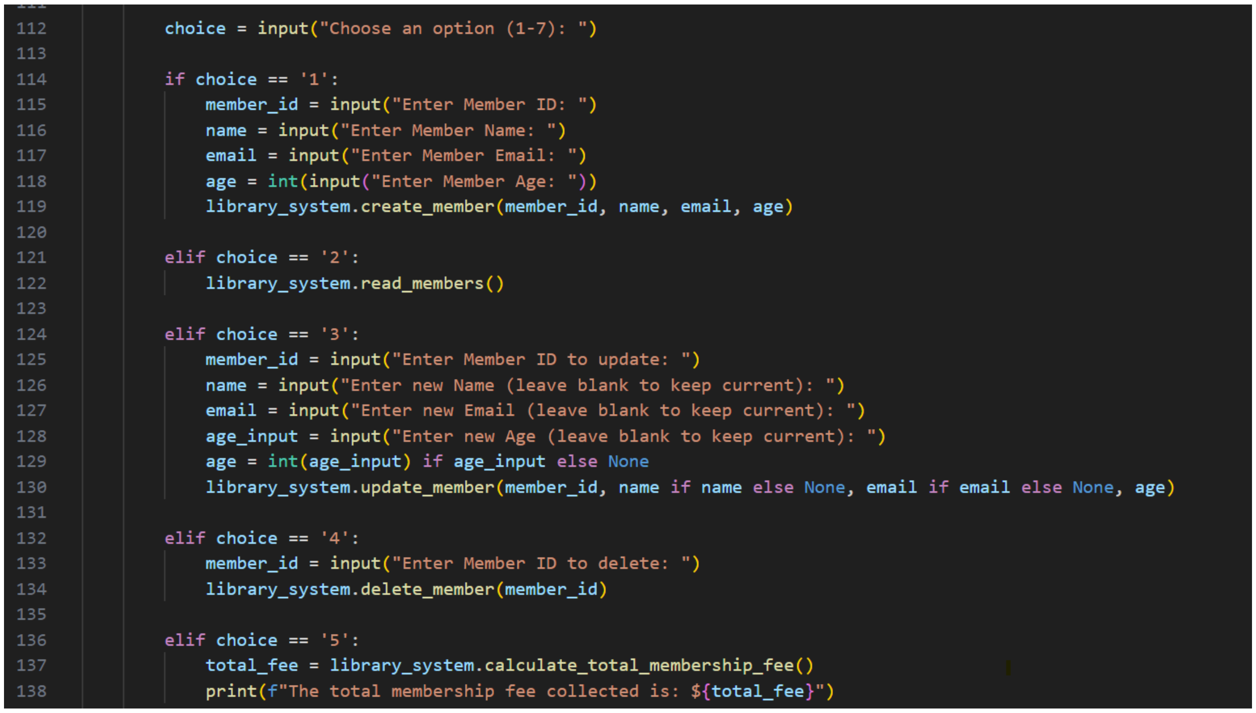


```python
    def calculate_average_membership_fee(self):
        """Calculate and return the average membership fee."""
        if self.total_members == 0:
            return 0  # Avoid division by zero
        return self.total_fee / self.total_members
```

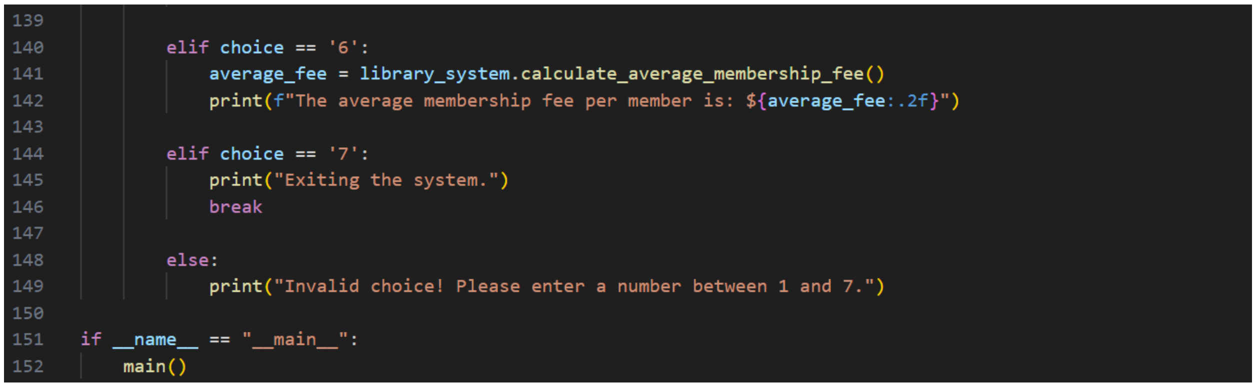


```
Choose an option (1-7): 6
The average membership fee per member is: $16.67
```

vii. **Choice**: choose an option



```
        choice = input("Choose an option (1-7): ")

        if choice == '1':
            member_id = input("Enter Member ID: ")
            name = input("Enter Member Name: ")
            email = input("Enter Member Email: ")
            age = int(input("Enter Member Age: "))
            library_system.create_member(member_id, name, email, age)

        elif choice == '2':
            library_system.read_members()

        elif choice == '3':
            member_id = input("Enter Member ID to update: ")
            name = input("Enter new Name (leave blank to keep current): ")
            email = input("Enter new Email (leave blank to keep current): ")
            age_input = input("Enter new Age (leave blank to keep current): ")
            age = int(age_input) if age_input else None
            library_system.update_member(member_id, name if name else None, email if email else None, age)

        elif choice == '4':
            member_id = input("Enter Member ID to delete: ")
            library_system.delete_member(member_id)

        elif choice == '5':
            total_fee = library_system.calculate_total_membership_fee()
            print(f"The total membership fee collected is: ${total_fee}")
```



```

        elif choice == '6':
            average_fee = library_system.calculate_average_membership_fee()
            print(f"The average membership fee per member is: ${average_fee:.2f}")

        elif choice == '7':
            print("Exiting the system.")
            break

        else:
            print("Invalid choice! Please enter a number between 1 and 7.")

if __name__ == "__main__":
    main()
```

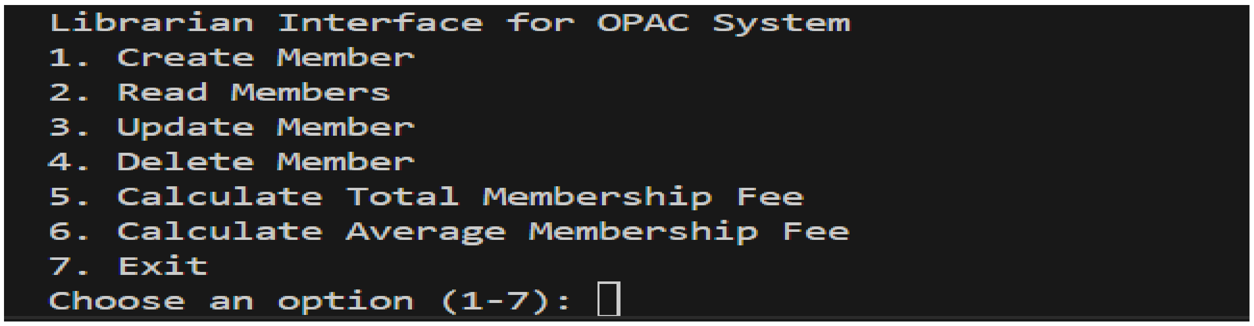
4.0 CONDITIONAL STATEMENT

i- If
ii- Else
iii- Elif
iv- For
v- Return
vi- While
vii- Def
viii- class

GUI :

No

5.0 RESULT



```
Librarian Interface for OPAC System
1. Create Member
2. Read Members
3. Update Member
4. Delete Member
5. Calculate Total Membership Fee
6. Calculate Average Membership Fee
7. Exit
Choose an option (1-7):
```

6.0 STRENGTH

1. Comprehensive CRUD Functionality:

- The system provides full Create, Read, Update, and Delete (CRUD) capabilities for managing library members, allowing for flexible and comprehensive member management.

2. Data Validation:

- The functionality includes checks to ensure that member IDs are unique upon member creation, which helps to maintain data integrity.

3. Financial Calculations:

- The system features capabilities to calculate total and average membership fees, which can provide valuable insights for management and financial reporting.

4. Scalability:

- The structured approach allows for future expansions, such as adding more member attributes or functionalities, adapting to changing requirements.

By implementing these strengths, the project can effectively address the typical needs of a library membership management system while also providing opportunities for learning and improvement in software development skills.

7.0 KAIZEN (ROOM FOR IMPROVEMENT)

1. User Interface Enhancements:

   - Providing a clear, intuitive design will improve user experience. Consider adding buttons, forms, and validations to enhance usability instead of relying solely on command-line interactions.

2. User Authentication:

   - Consider adding user authentication to restrict access to the system, ensuring only authorized librarians can manage membership data.

3. Membership Types and Benefits:

   - Introduce different membership types (e.g., student, senior, family) with varying fees and benefits. This feature would cater to a broader audience and enhance the membership model.

4. Advanced User Features:

   - Explore feature additions such as email notifications for members when their registration is confirmed, or membership fees are due.

12. Feedback Mechanism:

   - Implement a user feedback mechanism where librarians can report bugs or request new features, promoting continuous improvement based on user experience.

Implementing these improvements will help make the project not only more functional and user-friendly but also increase its maintainability and scalability for future development.